WebAlly: A Friendsourcing Approach to Solve CAPTCHAs for People with Visual Impairments

ANONYMOUS AUTHOR(S)

CAPTCHAs were rated as the second most frustrating task in daily web use. It is even more challenging for people with visual impairments (PVIs), especially with task-based visual CAPTCHAs like Google reCAPTCHA, where descriptive guidance from crowdsourcing assistive services hardly works. Other existing solutions, such as audio CAPTCHAs, have shown to be unusable. Compared to traditional crowdsourcing, friendsourcing operates in a similar way that could solicit help from friends rather than strangers and might mitigate privacy and security risks. To investigate the pros and cons of transferring inaccessible task-based visual CAPTCHAs to friends and the possible impact on their social relationships, we designed and tested a proof-of-concept system called WebAlly, using Google reCAPTCHA as the task in a study with 10 pairs of a PVI adult and a friend or family member (18 in total). Post-study interviews showed that WebAlly did well in task performance and participants felt the system protects PVIs' privacy and security. PVI participants also preferred to use WebAlly over solving reCAPTCHAs themsevles or using the crowdsourcing, which they considered as a backup alternative. Two thirds of participants also felt WebAlly brings their relationship with their friends closer.

CCS Concepts: • **Human-centered computing → Accessibility systems and tools**.

Additional Key Words and Phrases: Accessibility, Friendsourcing, Crowdsourcing, Task-based CAPTCHA

ACM Reference Format:

Anonymous Author(s). 2018. WebAlly: A Friendsourcing Approach to Solve CAPTCHAs for People with Visual Impairments. In *CSCW '21: ACM Conference on Computer-Supported Cooperative Work and Social Computing, November 03–07, 2021.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Large swathes of the web remain inaccessible for the 285 million people with visual impairments [37]. For instance, CAPTCHAs (Completely Automated Public Turing tests to tell Computers and Humans Apart) are commonly used to authenticate users in numerous day-to-day web surfing tasks [28] like registering new accounts, leaving comments on social media and completing financial transactions, yet are rated in a global study by WebAIM as the second most frustrating task encountered in day-to-day web use by PVIs [51].

CAPTCHAs are inaccessible for PVIs because they require users to engage in complex visual-processing tasks. For example, today, task-based visual CAPTCHAs, like GeeTest [43] and Google reCAPTCHA, are widely used across the web and require users perform high precision operations like dragging a slider to solve a puzzle, clicking certain icons from a mixed set, or selecting a few images from the gallery. These CAPTCHAs are challenging if not impossible to solve for PVIs who use assistive technologies to navigate the web.

To address this concern, prior work has explored a number of solutions. Most commonly, PVIs may use audio CAPTCHAs instead of visual CAPTCHAs. While an accessible alternative, in theory, prior work has found that audio

CAPTCHAs are disproportionately hard for PVIs relative to visual CAPTCHAs for people without visual impairments — they are significantly slower and require more attention and memory-capacity [12]. While researchers have proposed improved audio CAPTCHAs that are less difficult for PVIs to successfully complete [21] and help maintain their independence, these alternatives have not yet been widely implemented and cannot help PVIs in the short-term. Other solutions to help PVIs in the short-term include automated CAPTCHA solving services (e.g., WebVisum [47]), but these solutions work only for simple visual CAPTCHAs in which people are asked to identify distorted letters and numbers. There are also task-based CAPTCHA solvers (e.g., Anti-CAPTCHA [39]), but these pose security risks — they require users to install software with dangerous system-level permissions.

Outside of the context of CAPTCHAs, crowdsourcing and friendsourcing methods have shown great promise in helping PVIs transfer difficult web tasks to crowd workers or friends. Although they require interdependence, crowdsourcing and friendsourcing can help PVIs on these accessibility challenges they might encounter in today's web. Prior art, like BeMyEyes [40] and VizWiz [13], have explored connecting PVIs in need of help with remote online assistance from sighted helpers to, for example, answer questions about nearby surroundings and objects or request for vocal-guidance on how to use an inaccessible interface. However, these applications are limited to text- or voice-based descriptive guidance, which limit their utility in the context of helping PVIs outsource the solving of task-based CAPTCHAs. Indeed, while synchronous guidance can be helpful, the high-precision nature of modern task-based CAPTCHAs makes it difficult to provide instructions on how to solve the CAPTCHAs for people with visual impairments. One workaround to this limitation might be allowing helpers to temporarily take control of PVIs system to solve the CAPTCHA, but there are significant privacy and security risks in allowing sighted strangers control over PVIs' devices.

Friendsourcing can help PVIs overcome many of the same accessibility hurdles as does crowdsourcing, but could have a number of additional benefits. First, friendsourcing could pose fewer privacy and security risks for remote control situations in which a helper takes over the PVIs system to solve high-precision CAPTCHAs. Second, friendsourcing aligns with existing workflows for PVIs: PVIs often muster assistance from their friends and family members to overcome accessibility challenges in the physical world [1]. Third, friendsourcing can reduce or remove the financial costs associated with using trained crowd workers. However, friendsourcing is also typically slower and less reliable than crowdsourcing [8], and PVIs might want to avoid burdening their social connections with frequent requests for help [48]. Moreover, prior work suggests a complicated relationship between friendsourcing and payment — while PVIs may feel indebted to and want to compensate friends who assist them remotely, neither party may feel comfortable with monetary payments [59].

In short, PVIs commonly encounter task-based visual CAPTCHAs that frustrate and encumber their of the web, yet existing solutions fall short of their needs. We hypothesize that a remote assistance, friendsourcing tool that allows friends to directly solve these CAPTCHAs on PVIs behalf may help, but it remains unclear how a friendsourcing approach would compare to a crowdsourcing approach in terms of: Specifically, we are interested in the research question of how does our novel friendsourcing CAPTCHA solution work in terms of:

- **RQ1:** Task performance metrics (like accuracy and speed)
- **RQ2:** Perceived privacy and security risks
- **RQ3:** User preference (for friendsourcing or crowdsourcing)
- **RQ4:** Willingness to compensate helpers and preferred methods of compensation
- **RQ5:** Perceived impact on the social relationship between PVIs and their helpers

To answer these research questions, we implemented WebAlly, a prototype tool to help PVIs friendsource Google reCAPTCHA tasks. PVIs can trigger a request for remote assistance with keyboard shortcuts. Helpers receive a link to a secure webpage that they can open on any device and solve the Google reCAPTCHA task for PVI users synchronously via an interactive screenshot. We conducted a within-subjects, two-by-two (crowdsourcing vs. friendsourcing, paid vs. free) lab study with 18 participants (10 PVIs and 8 sighted friends) to evaluate WebAlly and answer our research questions. We recruited participants in pairs (one PVI and one friend who served as a remote helper), and had PVIs use WebAlly to request that their helper help them solve a Google reCAPTCHA task. Across the different experimental conditions, we primed PVIs to think of their remote helper as a friend or stranger, and to think of whether or not they would be paying their remote helper.

We found that while WebAlly with friendsourcing requires interdependence, PVIs prefer it over using other time-consuming or low success-rate solutions, and expressed wanting to have WebAlly with crowdsourcing as a backup alternative. As for compensation, participants generally tend to compensate the helpers but have their own preferences in how to compensate. Participants also found that WebAlly did well in task performance and protecting their privacy and security. In addition, we found that WebAlly had a positive impact on their social relationship with the helpers. Based on our results, we derived several practical design implications for future work.

In summary, the contributions of this work are:

- A framework to source direct help from friends to solve a top-rated challenge for web accessibility—task-based visual CAPTCHA—for PVIs.
- A prototype and a user study that revealed the challenges, social considerations, and practical implications of designing friendsourcing tools to solve task-based visual CAPTCHAs.

2 RELATED WORK

2.1 PVI with CAPTCHAs

CAPTCHAs are designed to distinguish human users from robots. Traditional CAPTCHAs usually pose a visual challenge, like recognizing images, words, or numbers out of a single or set of specific images. These interactive tasks are meant to be simple for human users with their perception and common sense. However, those tasks' nature prevents certain groups of users from solving them and further accessing web resources. Researchers found that CAPTCHA brings accessibility issues for users with visual, physical, cognitive, or auditory disabilities [11, 22, 33, 49]. For PVIs specifically, they could not access the visual challenges and rely on other alternatives and help from the outside world.

As an alternative, audio CAPTCHA is more accessible for PVIs. Prior work identified two types of audio CAPTCHAs: content-based and rule-based [14]. Content-based audio CAPTCHAs ask users to convert the speech to text, usually a few words. They are the most commonly used audio CAPTCHAs. For instance, Google reCAPTCHA is the most widely used CAPTCHA service and has been adopted by many popular websites [46]. It also provides a built-in audio version for people who cannot pass the visual challenges. Rule-based CAPTCHAs ask users to interpret information and perform some in-mind actions like counting number of times they hear something, which are less common. However, audio CAPTCHAs are not always available on many websites, and current audio CAPTCHA designs have proven to be difficult and time-consuming for PVIs throughout several research studies [12, 27, 29, 30]. To prevent hackings from Natural Language Processing (NLP)-based bots [14], the audio file provided to users are usually speakers saying words at randomly spaced intervals with background noise. These interferences challenges both automated agents and human users [27, 52]. Many existing research studies have also tried to increase the accessibility of audio CAPTCHAs. Fanelle et al. designed four novel audio CAPTCHAs to increase accuracy and speed [21]. Jain et al. proposed *reCAPGen*, a system that uses automatic speech recognition for generating more usable and secure audio CAPTCHAs [27]. They all explored how users (especially PVIs) can independently solve audio CAPTCHAs.

Prior work also provided many examples of directly breaking CAPTCHAs. Some early research leveraged image and pattern recognition techniques to break visual CAPTCHAs [18, 34, 57]. More recent research also provided various types of hackings towards task-based CAPTCHAs like Google reCAPTCHA [7, 32, 50, 58]. In industry, there are also many paid CAPTCHA solving services like Anti-CAPTCHA [39] and Buster [41]. Although these techniques could be easily adopted in browser extensions or system-level applications to hack inaccessible CAPTCHAs directly for PVIs, the original purposes of these research are still aimed for improving the CAPTCHA's security by revealing how they can be hacked. Thus, we are exploring how friendsourcing can serve as a solution for solving tasks like CAPTCHA without compromising its security features. Additionally, using hacking services like Anti-CAPTCHA would introduce privacy and security issues for users. Users will need to download browser extension folders directly from their website rather than installing from official web stores, and users are required to edit their personal computer's registry to make the tool work.

2.2 Privacy and Security Concerns of PVIs

As online resources have been more and more available and accessible for users with visual impairments, there is a trend towards empowering PVIs to protect private information and their online security. Gurari et al. introduced the first visual privacy dataset originated from PVIs, revealing a challenge of understanding and protecting their privacy needs [23]. The dataset also includes information that can be easily captured on PVIs' computer screens. As crowdsourcing remote assistance services like BeMyEyes and Eyecoming [42] have been widely used by PVIs and make their lives easier, researchers have also investigated how these services would raise privacy and security risks [3, 5, 6, 56]. Akter et al. conducted a study to understand privacy concerns when PVIs use camera-based assistive technologies [6]. Ahmed et al. took another angle and studied the information sharing preferences of sighted bystanders of assistive devices [3]. Existing research has also shown that PVIs have strong security and privacy concerns in using CAPTCHA [2, 4, 19, 24, 26]. Holman et al. identified their top 10 security challenges and CAPTCHA has been listed as the top one challenge [26], which poses a challenge of how to help PVIs solve these small tasks like CAPTCHA without compromising their privacy and security.

2.3 Friendsourcing for PVIs

Socio-technical researchers have conducted many studies on collaborative systems. Traditional crowdsourcing has proved a convenient way to get answers quickly from the online crowd. The VizWiz smartphone application allows visually-impaired users to send visual questions to sighted crowd workers and get answers soon [13]. However, such services can be limited due to the cost of the paid crowd workers, which might add extra and unexpected burden to PVIs [16]. As a free alternative, friendsourcing could also help users solicit answers and assists from friends via online social network services, and the answers are often from more trustworthy and tailored to their interests than using a search engine [36]. Traditional online social networks sites include Facebook and Twitter [10, 35, 36, 48]. Over 50% of social network users report that they used these sites to ask their friends questions [35]. To further fulfill social network's potential in sourcing help, researchers also built systems to help combat email harassment [31], personalize user experiences [10] or help PVIs get answers of questions about their surroundings via cameras [17]. For PVIs specifically, AbdraboTarek et al. proposed an assistive tool for blind users to deal with their daily activities via smartphone and

Twitter [1]. Brady et al. studied PVIs' perceptions of social microvolunteering via a Facebook application that answers visual questions on behalf of blind users [16].

Although friendsourcing was initially a free option to solicit answers and help, different compensations might lead to other changes such as different response rates or an impact on the social relationship. Zhu et al. studied the effects of extrinsic rewards and monetary payments to further investigate how friendsourcing would impact PVIs' social relationship with their friends [59]. Other research also revealed how these rewards might undermine the original motivation that drives friendsourcing activity and change the perceived relationship between people [25, 38, 53, 54]. In addition, independence is often considered as a goal in assistive technologies [9]. Even sometimes the goal is not explicitly stated, the researchers agree that "all accessible computing approaches share a common goal of improving independence, access, and quality of life for people with disabilities" [55]. However, as Bennett et al. pointed out, interdependence is also valuable because the interactions between people with disabilities and their allies are often two-way and mutually beneficial [9]. In our work, we designed a collaborative system that allows allies to help solve CAPTCHAs for PVIs. It is important to note that our friendsourcing approach is complementary to existing mechanisms.

3 DESIGN FRAMEWORK

We began by identifying the challenges PVIs experience with task-based CAPTCHAs — we enumerated existing solutions to help PVIs overcome these CAPTCHAs, investigated how existing solutions fall short of PVIs' needs, and uncovered how a friendsourcing framework could play a role in helping PVIs solve these tasks. Then, we synthesized several design goals to support PVIs with an accessible tool to solve Google reCAPTCHA, one of the most commonly used task-based CAPTCHA on the web. Based on these design goals, we developed WebAlly: a collaborative friendsourcing prototype implemented to assist PVIs in solving the Google reCAPTCHA.

3.1 Challenges

Through a survey of related work and existing systems, we identified three key challenges that PVIs face when attempting to solve task-based CAPTCHAs:

C1 - Providing PVIs with more direct help. Existing tools such as BeMyEyes and Eyecoming provide PVIs with remote assistance services: e.g., providing descriptive guidance on how to operate an interface and navigation guidance via smart glasses. However, these collaborative assistance services are limited to providing indirect help. PVIs rely on helpers' textual or audio guidance, either synchronously or asynchronously, to solve the ongoing task on their own. This type of assistance is helpful, fosters independence, and is widely used by PVIs for everyday life tasks []. However, when the task requires precise hand-eye coordination (e.g., moving the mouse and click a specific object), text-based assistance becomes challenging. For instance, to solve task-based CAPTCHAs, users would need to pick the right tile images, drag a puzzle piece to the right place, or click some letters and numbers in the right order. These actions are not easy for PVIs to perform independently even with descriptive guidance. One opportunity to address this challenge is to afford a remote helper direct control of the PVIs system to solve the task-based CAPTCHA on behalf of the PVI. However, it is still challenging to make remote control assistance secure and accessible for PVIs.

C2 - Protecting PVIs' privacy and security while getting help. While remote control assistance could help PVIs overcome task-based CAPTCHAs, it might also cause privacy and security concerns As many PVIs may be unable to receive visual feedback or otherwise monitor helpers' behaviors, remote helpers could perform malicious actions on PVIs' devices without their awareness, become aware of what the PVI is trying to do online, or be able to see sensitive personal information that may be present on the PVIs screen. Thus, there is a need to protect PVIs' privacy and security as they receive help through remote assistance systems and services — both for indirect descriptive guidance (when PVIs usually need to point their cameras to the computer screen) and direct remote control.

C3 - CAPTCHA restrictions. In CAPTCHA design, there is usually a trade-off between security and user experience. Task-based CAPTCHA can be difficult to solve even for humans to achieve its original purpose of differentiating humans from bots. The complex and diverse design of task-based CAPTCHA poses another challenge. For example, Google reCAPTCHA, the most common type of task-based CAPTCHA, has a solving time limit of two minutes. It also expires within one minute before submission, such that users must complete and submit a form protected by reCAPTCHA before it expires, lest they have to solve another reCAPTCHA puzzle. Remote assistance services usually take more than two minutes to post requests, find volunteers, synchronize with the helper, get help, and get notifications when the session is complete. Often the case is that PVIs need to wait for someone to answer their requests. It is naturally challenging to source help in a short amount of time before the current CAPTCHA expires.

3.2 Design Goals

To address these challenges, we highlighted several goals that we identified as essential for designing an efficient and accessible CAPTCHA-solving tool for PVIs. Our high-level design goals were to integrate social support, reduce human effort, source help efficiently, protect PVIs' privacy, and still maintain the security utility of CAPTCHAs — differentiating between humans and bots.

G1 - Limited remote control. To provide PVIs with more direct help and maintain their privacy and security at the same time, our goal is to design a limited, sandboxed remote control system in which helpers are restricted in the actions they can perform and the screen information they can see. Specifically, helpers should only be able to perform actions necessary to solve the CAPTCHA, and should only be able to see parts of the PVI's screen that is relevant to the CAPTCHA. In this case, helpers can not access any sensitive information or perform other actions on the PVIs' personal devices.

G2 - Sourcing help in time. Task-based CAPTCHAs like Google reCAPTCHA have time limits, after which they expire and a new puzzle is issued. Our goal is to source help within this time limit. To accomplish this goal, our system must: (1) send the PVIs' request, (2) synchronize with a helper, (3) have the helper complete the task. For a fast solving experience, the user interface for the helpers should be simple and straightforward so they can minimize the completion time to avoid expiration. It is a long-term design goal that requires design iterations and field studies, which is not a major focus of our initial prototype.

G3 - Accessible for PVIs. To ensure that our remote assistance tool is accessible, our goal is to make all its functions available via keyboard shortcuts Another design goal to enhance accessibility is to provide audio feedback at every stage of helping, to notify PVIs about the current state of the task and what the helper is doing.

G4 - Complement not replacement. We also try to maintain PVIs' independence when they could not bypass the CAPTCHAs and need to actually solve them. Instead of replacing the current solving-flow of PVIs, our tool was designed as a complement mechanism. Our tool offers an additional option when people find solving CAPTCHA difficult on their own.

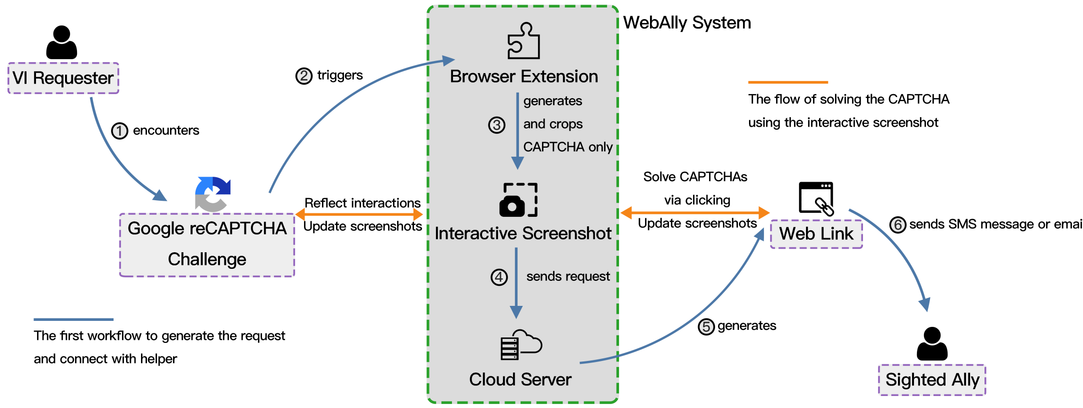

Fig. 1. Workflow of WebAlly: VI requester triggers the browser extension and sends the request, and the tool builds a channel between the requester and the helper via an interactive screenshot: Reflect helper's clicking on requester's screen and update screenshots back to helper's interface until the challenge is passed

3.3 System Design

Guided by our design goals, we designed and implemented WebAlly — a proof-of-concept friendsourcing tool that connects PVIs with their friends or family members when they encounter a Google reCAPTCHA task but have difficulty solving it. WebAlly creates an interactive screenshot of the reCAPTCHA task and sends it to the helper. This interactive screenshot serves as a canvas on which helpers' can perform actions to solve the reCAPTCHA task (e.g., clicking on tiles, dragging UI elements); these actions, in turn, are reflected on the PVIs screen. However, access to the source device is controlled and limited — helpers see only task-relevant screen information, and only pre-specified interactions (e.g., clicks) are reflected on the PVI's screen. In the following sections, we will use the terminology "requester" to refer to the visually-impaired users sending requests for help, and "helper" to refer to their friends and family offering help.

3.3.1 System Overview. WebAlly is implemented as a browser extension, written in JavaScript and executable on Chromium-based browsers (such as Google Chrome, Microsoft Edge, Opera and Brave). We also incorporated OpenCV to pre-process images, and WebSocket as a channel to transmit messages in real time. The workflow contains a one-way request from the requester and a synchronous collaboration process between the requester and the helper (see figure 1).

The system stores helpers' contact information in requesters' browser storage via extension API. This contact information is also encrypted and stored temporarily in the back-end server and deleted regularly. The interactive screenshots are also stored on the back-end server and deleted after each helping session. However, the screenshots do not contain any sensitive information since they are only the CAPTCHA image part of the screen.

3.3.2 Requester-side Interface. The requester interface is not graphical. After installing the browser extension (add-on), the requester must enter and store the contact information for a helper using a keyboard shortcut to activate the function. The requester will type in the information via the screenreader's prompts and save it. After finishing the preparation process, the requester can activate the extension and send the request to the preset helper using an editable keyboard shortcut combination. The WebAlly system then takes a screenshot of the current browser tab and uses the template-matching feature in OpenCV.js to crop the screenshot down to just the region that contains the Google reCAPTCHA task. Only the cropped image will be sent to the helper. After sending the request, the requester does not need to perform additional actions until the system gives audio messages about success or failure. If the reCAPTCHA

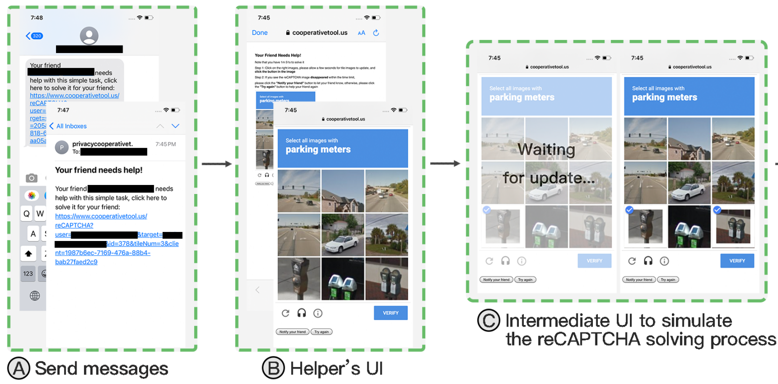

Fig. 2. User Interface from the helper's side: Requester sends message, helper opens the link, solve the CAPTCHA as usual, and click "VERIFY" to complete (The solving process might take multiple visual challenges to pass)

task is successfully solved, the requester can end the collaborative session and proceed with the task at-hand, like submitting a form or completing a transaction. If the helper failed to pass the test in time, the requester could ask again using the same keyboard shortcut.

3.3.3 Helper-side Interface. When a requester asks for assistance, the helper will receive both SMS messages and emails with a URL to a secure page hosted on WebAlly's backend server. The helper can then click on the web link and complete the requested task on their own web browser. The helper's interface contains instructions on what to do and the cropped interactive screenshot sent from the requester. WebAlly establishes a connection between the helper and the requester via a WebSocket. To reflect the helper's actions on their into the requester's side, WebAlly will record, transmit, and simulate the helper's clicks on the interactive screenshot. As the reCAPTCHA interface will update the tile images after each click, the system will also detect the updated part, crop it using OpenCV.js, and reflect these changes back on the interactive screenshot presented to helpers. These changes happen in real-time through the synchronous WebSocket connection — thus, the helper's experience mimics how they might solve a CAPTCHA for themselves.

The helper will click on the right tile images as described in the reCAPTCHA task, and click the "VERIFY" button when the task is complete. After each click, there will be a short delay in transmitting messages to the requester's side and reflecting back. Since the helper's selections might not be accurate to pass the task, the Google reCAPTCHA interface will ask the user to try again, which will also be transmitted back and forth until the task is complete or the time is up. As we mentioned in the challenges, the Google reCAPTCHA task has a two-minute time limit before it expires. WebAlly set a timer in the helper's interface for tracking the remaining time. If the helper solved the task successfully within the time limit, WebAlly will send an audio message of success to the requester. If the time is up before solving it correctly, WebAlly will also send an audio message to the requester and ask for another try. The system covered all possible scenarios and edge cases in use and provided audio feedback to the requester to acknowledge the ongoing process.

4 USER STUDY

Based on our literature review and design goals, we were interested in how WebAlly could help PVIs solve Google reCAPTCHA well, and how different settings of compensation and help source would impact users differently. We

Table 1. Participant demographics, including requesters' age range, visually impaired status, screen reader of use, and their relationship with the helpers

Participant ID	Age range	Requester Gender	VI status	Screenreader of use	Paired Helper	Helper Gender	Relationship
R01	25-34	Male	Blind	NVDA	H01	Male	Brother
R02	18-24	Male	Blind	NVDA	H02	Female	Sister
R03	55-64	Female	Deafblind	NVDA	H03	Male	Brother
R04	25-34	Female	Blind	JAWS	H04	Female	Brother
R05	25-34	Male	Blind	JAWS	H05	Female	Friend
R06	25-34	Male	Low Vision	NVDA	H06	Female	Friend
R07	25-34	Male	Low Vision	NVDA, JAWS	H07	Male	Friend
R08	35-44	Male	Blind	JAWS	H08	Female	Friend
R09	45-54	Female	Blind	JAWS	H08	Female	Friend
R10	35-44	Female	Blind	JAWS	H08	Female	Friend

conducted a two-by-two (crowdsourcing vs. friendsourcing, paid vs. free) within-subjects experiment with 18 participants (10 PVIs and 8 helpers) to evaluate WebAlly and answer our research questions. Our study was IRB-approved.

4.1 Method

Participants. We recruited participants to join our study in pairs: one PVI and one helper who the PVI would consider a close friend or family member. In total, we recruited 18 participants (see table 1), including 10 participants (6 males, 4 females) with visual impairments (referred to as requesters R1-R10) and 8 participants (3 males, 5 females) without visual impairments who are friends or family members of the PVIs (referred to as helpers H1-H8 just for simplicity). For the participants with visual impairments, one self-identified as deafblind, two self-identified as low-vision, and seven self-identified as blind. One helper (H3) self-identified as having some hearing impairment. For the screen readers they were using, one was using both NVDA [45] and JAWS [44], four were using NVDA, and five were using JAWS. For the study sessions, requesters R1-R7 were in pairs with helpers H1-H7 accordingly in our study, and H8 was a mutual friend to requester R8-R10, who joined the study with them for three times. For their relationship, one helper is the sister of the PVI, three helpers are brothers of the PVIs, and four helpers were friends of the PVIs.

Apparatus. We used the WebAlly prototype to conduct the study. We asked requesters to install WebAlly on their Chrome browsers before the study. To trigger the Google reCAPTCHA task during the study, we used an example website (https://www.google.com/recaptcha/api2/demo) that reliably triggers reCAPTCHA image testing (see figure 3). We asked helpers to have a device through which they could join the online video conference and receive SMS messages and/or emails.

Procedure and Data Collection. The user studies were all conducted remotely via video conference calls. Before the study, we confirmed from the participants that they had regular access to a laptop or desktop computers, and that they used Mozilla Firefox or Google Chrome to surf the web in their daily life. All sessions lasted about an hour, including a post-study interview to learn the requesters' and the helpers' feedback separately.

At the beginning of each study session, two researchers and two participants (the requester and the ally) would join the online meeting. Participants were introduced to the purpose of the study and the study procedures. In some cases, the requesters did not or failed to install the tool in advance. The researchers then helped them install the tool via the online meeting and store the helper's information in the tool to send all the requests to the ally during the study. After all the preparation was complete, the researchers would divide the two participants into two breakout rooms to simulate remote collaboration (i.e., they did not need to be physically co-located to use the system). The two researchers who helped conduct the study went into the each of the two breakout rooms in order to observe helper

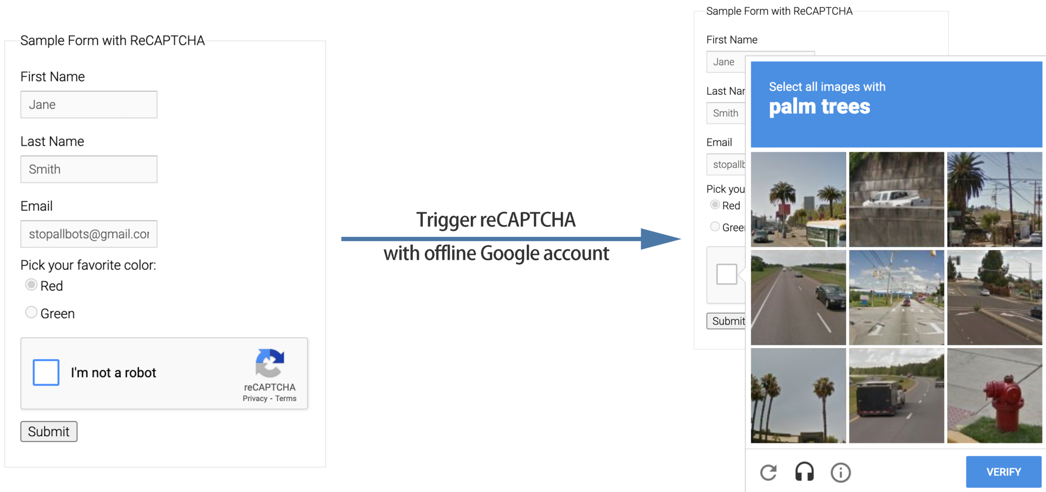

Fig. 3. The demo website with a login form to trigger Google reCAPTCHA

Table 2. Factorial design of 2 x 2 conditions, including factors of compensation and source of help. Each factor has two values for comparison and controlled experiment

Factors	Value 1	Value 2
Compensation of help	Free	Paid (free choice of rewards)
Source of help	Friendsourcing	Crowdsourcing

and requester behaviors and answer their questions. Splitting requesters and helpers up into two separate breakout rooms helped approximate a real-world scenario in which a PVI would need to to solve a Google reCAPTCHA task and choose to source remote help from friends.

After the researchers and the participants settled in different rooms, the researcher in the requester room (referred to as Researcher 1) introduced the tasks and asked the participants questions from a pre-study questionnaire about their experience with CAPTCHAs and Google reCAPTCHA. The questionnaire was designed to understand how participants generally solve reCAPTCHAs and the challenges they faced in solving these task-based CAPTCHAs.

After the questionnaire, in the requester room, Researcher 1 asked the requester to imagine that they are under one of four different scenarios (details below). We had a 2 x 2 within-subjects experimental design with two factors: Source of help (ally vs. stranger), and compensation for help (free vs. paid) (see table 2). In the helper's room, the researcher (referred to as Researcher 2) also introduced the different conditions to the helper and asked them to behave accordingly under different scenarios (e.g., imagine that a blind person whom you do not know asks your for free, voluntary help to solve a reCAPTCHA). We used the 2 x 2 design as an initial probe to dig valuable insights from such speculations [20].

The researchers randomized the order in which the four conditions were presented to reduce order effects. Under each condition, researchers asked participants to use WebAlly to solve a reCAPTCHA task together. Broadly, the PVI asks their helper for assistance, the helper solves the reCAPTCHA on his/her own screen, which would be automatically transmitted back to the helper through the limited remote control functionality we implemented in WebAlly. To reliably trigger a reCAPTCHA challenge under each condition, requesters used the aforementioned demo website. Then the requester would send the challenge to the helper through a keyboard shortcut. The helper could receive the help request message via SMS or email (see figure 2 A). The message would contain a broad description of the requested remote assistance task (see figure 2 D) and a web link containing the interactive screenshot of the Google reCAPTCHA (see figure 2 B, C). In ideal scenarios, each condition contained just one task-solving process. However, if the helper failed to complete the task within the time limit and the reCAPTCHA expired, the researchers asked participants to repeat the process again under the same condition.

After completing all of the four conditions, the researchers conducted an exit interview in which we asked about participants' detailed experiences with the system and their preferences among the four conditions. The questions include general feedback, suggestions on improvement, which condition they chose as their favorite and why, their privacy and security concerns in details, and how the request might alter the relationship dynamics between requesters and helpers.

4.2 Analysis

We video-recorded all the online sessions and took notes from the procedure and the interview. We transcribed the videos and used thematic analysis [15] to qualitatively analyze the study. Broadly, our analysis was driven by our core research questions and covered both PVIs and helpers perspectives. For PVIs, our codes cover prior experience with solving CAPTCHAs (e.g., tools used or methods for sourcing help), their general feelings about WebAlly, concerns on using WebAlly and its potential privacy/security risks, their preferences with respect to crowd/friendsourcing and paid/unpaid versions of WebAlly, their desire to use WebAlly in the future (willing to install and recommend), and their perception of WebAlly might affect their relationship with friends and family. For helpers, our analysis covered their overall feeling and suggestions, the willingness and general availability to help PVIs or even a broader user group with compensation or not, their choice on different conditions to help, and related feelings in detail.

5 RESULTS

5.1 RQ1: System's Task Performance

To answer our first research question of WebAlly's performance, we asked all participants about their general impressions and suggestions in terms of usability. We also recorded each task's success rate and time duration for quantitative analysis on the system performance.

General Impression. We first aimed to understand how PVI requesters and their helpers thought about the use of WebAlly to solve Google reCAPTCHA. Since we have four different controlled conditions, the overall feedback we gathered was more focusing on the usage and the framework of sourcing help from friends and transferring tasks to them, not impressions on different controlled conditions. The participants generally liked the tool but also identified its pros and cons. For the benefits, many thought the solving process was successful, and they were excited about the overall idea and how the tool was easy and efficient (see figure 5). For instance, R1 said it was a *"really bright idea,"* and R2 thought it was a *"really nice way"* as *"many websites do not have audio CAPTCHAs, and many audio ones just do not work"* and R2 *"really appreciate this invention."* Many helper participants also thought positively about the tool. For instance, H4 mentioned that *"it is great especially from a privacy perspective,"* and H6 said that *"it is simple and helpful, and I feel great to help someone."*

However, some users (3 out of 10 requesters and 4 out of 8 helpers) also brought up some usability issues that could compromise the user experience. H1 mentioned that *"The server response time could be improved for a bit"* and H2 said *"Sometimes I need to wait for a while for images to update and I got confused when I clicked other squares too quick"*. R8

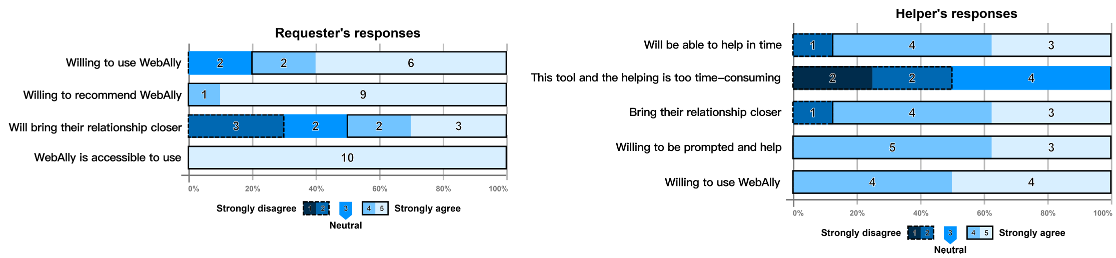

Fig. 4. Requesters' and helpers' responses on the acceptance of the tool, possible change in their relationship, and the tool's accessibility level

also mentioned that *"It is limited to Google CAPTCHA, and it takes some time. It also expires sometimes"* Two users also expressed their concerns about friends' availability. For instance, they hesitated in sending request messages to their friends when the timing is not right.

Performance Metrics. In general, the WebAlly tool performed well in the study under the four different conditions since the only differences across conditions were what we asked the participants to imagine (e.g., imagine you are seeking help from a stranger without any pay). Most of the solving sessions were successful. We consider a session successful if the ally could help the requester pass the Google reCAPTCHA task via the interactive screenshot within the time limit. Overall, the success rate of solving the reCAPTCHA task (see figure 5) under all four conditions is 88% (SD=0.10). Most times, the helpers could successfully solve the case within one minute. The average solving time for each task (if the task was successfully solved, also see figure 5) was 37.9 seconds (SD=12.65). Since all conditions were randomized in order during the sessions, and each condition was simulated the same way for participants, we only offered the overall metrics in every study session. For the three study sessions with the same helper (requester R8-R10 and helper H8), the decreased solving time (27.25 and 31.5 seconds) and increased success rate (100% for both) also showed that the tool became even easier to use after training. H8 became familiar with the tool quickly and used it independently without raising any question with the researcher in the study session. In the meantime, some participants brought up usability and user experience (UX) issues, which are illustrated in the next section.

Usability and UX Issues. From the helper's perspective, the primary issue was about the server response time. 3 out of 10 helpers (H1, H2, H5) mentioned that the server response time is not quick enough for smooth interactions. They experienced a noticeable pause after each click to wait for image updating. Unlike the real reCAPTCHA tasks where they can click on multiple images quickly, the interactive screenshots would require helpers to wait for 1-2 seconds. One helper (H3) also mentioned that the link for help seems suspicious when it was sent via text message or email without prior knowledge.

The requesters were comfortable with the tool's performance, mostly. However, they experienced some user experience issues. 2 out of 10 requesters (R6, R7) said that the tool's notification of success was overlapped with their screenreader's voice message. Initially, the implementation neglected the fact that screen readers like NVDA would notify the users when they pass the CAPTCHA test. 3 out of 10 requesters (R1, R9, R10) also mentioned that they want to have a cross-platform tool available on both laptops and mobile devices. One requester (R5) also pointed out a keyboard shortcut conflict with existing combinations provided by JAWS, which could be easily avoided in browser

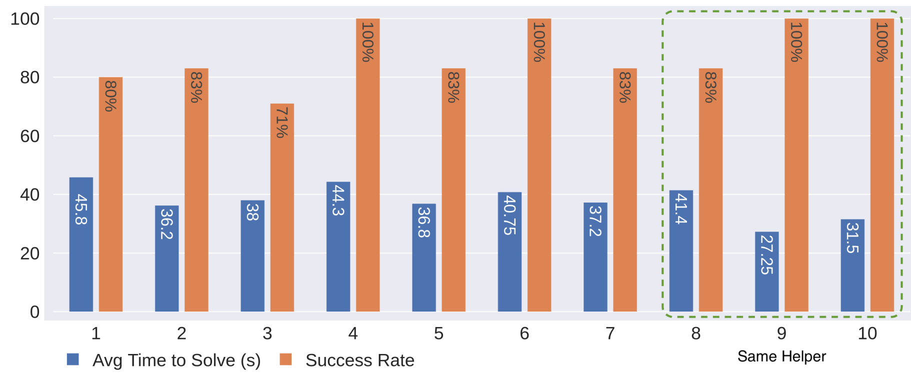

Fig. 5. WebAlly's performance on success rate and average solving time, with an overall success rate of 88% (SD=0.10) and average time to solve of 37.9 seconds (SD=12.65). H8 participated in session 8-10, and session 9 and 10 showed significant improvement after the helper got familiar with the tool

extension settings. However, it also indicated that all configurations of the tool, including changing keyboard shortcuts, should be accessible to PVIs.

5.2 RQ2: Privacy and Security Concerns

To understand how WebAlly's feature of protecting privacy and security is perceived by the participants and how more general friendsourcing tools would raise their concerns in privacy and security, we designed several open-ended questions in the exit interview. The questions were mainly about their concern levels and detailed feelings in using WebAlly and under different conditions like friendsourcing or crowdsourcing.

Concerns Regarding the Tool. At first, requesters were not aware of the image cropping feature of WebAlly because the cropped interactive screenshots were only available to the helpers. Without preset knowledge, 3 out of 10 requesters (R3, R4, R8) proactively asked researchers if the helpers (no matter friends or stranger volunteers) could see their entire screen since they have this impression from traditional remote control assistance. After researchers have explained the feature that WebAlly used computer vision techniques to crop the screenshot and saved the CAPTCHA image part only, all requesters stated that WebAlly's feature on protecting their privacy and security was acceptable. In the meantime, the researchers also further expanded that image cropping may not always be available on every task, and it worked on Google reCAPTCHA because it was easy to template matching the reCAPTCHA image areas. With this explanation, the requesters further talked about their concerns in detail.

Concerns Regarding Sourcing Help in General. Participants still had concerns over more general scenes of using crowdsourcing or friendsourcing to solve small daily tasks. Four out of ten requesters (R4, R8, R9, R10) initiatively mentioned privacy and security issues, but they also stated that they would not be bothered much whether their privacy or security was compromised during the helping session. They were more concerned with *"getting the job done"* (R4). Other requesters showed concerns about privacy- and security-related risks when asking strangers on the Internet for help, especially when the situation is about monetary transactions or signing up for new accounts. Although they felt that volunteers on the Internet might have good intentions to help them, some PVIs still had some concerns, e.g., R8 said *"You can never be too careful"*. This implies that they had their own priorities and preferences. We also observed that all five US requesters voluntarily brought up privacy/security before the researchers reached the privacy/security questions, compared with one out of four Indian requesters who did that. There might be a cultural dimension, and future research is needed to further investigate the potential impact of culture on people's tendency to trust others and their privacy and security concerns.

Our participants also showed different concerns in specific scenes. We found that requesters were more concerned with financial-related situations using crowdsourcing. They also have security concerns about what volunteer strangers can access on their computers. In comparison, requesters had more concerns using friendsourcing when it comes to private information like browsing history. We asked requesters about any concerns using friendsourcing to solve more general small online tasks, where cropping exact screen parts may not always be possible. They expressed a shared opinion that they would prefer not to let their friends see the screenshot and know what websites they are visiting (R8: *"I don't want my friends to know where I am looking, but I don't care if strangers see it. They don't know me anyway"*). However, they do not care if the crowd workers could access their browsing histories since they are not familiar. This result also implies a separate privacy/security consideration of different scenes when using friendsourcing or crowdsourcing.

Helpers' trust in the tool. From the helpers' perspective, when we asked them to imagine they were helping PVIs that they do not know on the Internet, some helpers (H3, H6) mentioned that they would not trust some random texts from some app, which might be insecure like phishing links. Although the web link sent to helpers was encrypted using Secure Sockets Layer (SSL) certificates, helpers without much technical background would not recognize these features and show less trust in the tool. They suggested that the tool should also provide a channel for helpers to register beforehand (like BeMyEyes) even though they do not need to install the extension to help PVIs. A registration process would give helpers some presupposition and increase their trust in the tool when receiving messages.

5.3 RQ3: Crowdsourcing or Friendsourcing

Another main research goal of this study was to compare the friendsourcing approach with the crowdsourcing approach in solving inaccessible small online tasks like reCAPTCHA. Most participants enjoyed sourcing help from their friends in small favors. When asked about their choice on the favorite condition of help source, 7 out of 10 requesters chose a friendsourcing-based tool to solve real-life tasks. They mentioned that asking friends for help would make them feel more secure than involving a crowd worker, and small favors like solving a CAPTCHA would not bother friends much. In comparison, other three requesters (R7, R8, R10) who preferred crowdsourcing-based tools expressed concerns about bothering their friends (R7: *"I don't want to interrupt my friend when it's midnight"*) and privacy issues with them as mentioned in the last subsection. In general, participants showed a trend in preference of having the option to choose from friendsourcing or crowdsourcing, which could help them in more complex scenarios, and save crowdsourcing as an extra alternative plan when they do not want to bother friends in certain situations.

5.4 RQ4: Compensation to Helpers

Besides investigating the source of help, we also evaluated users' acceptance from the compensation angle. In general, different PVIs have different preferences for compensating their friends who help them solve CAPTCHA tasks. 3 out of 7 requesters who preferred to use a friendsourcing-based tool mentioned that they would like to save the money because CAPTCHA is a simple task for their friends. In comparison, 4 out of 7 requesters still preferred to compensate their friends for helping them, even for small favors. As for the compensation types, we found that nearly every requester (9 out of 10), even the ones prefer to receive small helps for free, would prefer to compensate their helper by extrinsic rewards like *"a cup of coffee"* or *"buy him/her a beer."* They believe paying their friends a small amount of money is not an ideal option, which would make them *"feel weird,"* or nearly an *"insult."* R2 mentioned that a subscription service would also be acceptable for both requesters and helpers since a routined and fixed payment would cause less embarrassment between the requesters and helpers. Nearly all participants would not prefer to introduce the money-transferring process into friendsourcing scenes.

For the 3 out of 10 requesters who would rather choose to crowdsource help, they also chose to use a paid service rather than a free tool. Some requesters (R8, R9) were already using paid remote assistance to help them with any technical issues they encounter while using computers. These requesters expressed that they would trust the crowd workers more when they paid for the service as the helpers would be *"trained or professional workers."*

From the helpers' side, they expressed similar thoughts. Most helpers found that being treated with some extrinsic rewards was more acceptable. For solving a small task like Google reCAPTCHA, they would also settle for completely voluntary work with no payments or rewards. These results imply that small favors' payments are of less concern to users, and if compensations are involved, requesters would prefer to avoid monetary payments and use extrinsic rewards to avoid awkwardness or embarrassment. We also designed similar questions to ask whether helpers are willing to help strangers using such a tool. Most of them presented goodwill towards both their friends and also requesters online that requested crowdsourcing help. Only one helper (H3) mentioned that helping strangers might be overwhelming because *"you need to pick up random messages"*. Most helpers mentioned that they would not feel bothered by a small number of PVIs with a small number of requests. As for the compensation of help, most helpers would not expect getting paid or getting paid much because *"solving a CAPTCHA only takes seconds"* (H8).

5.5 RQ5: Impact on Social Relationships

While helpers showed a willingness to help their friends, one concern is whether using this friendsourcing tool would cost friends' social relationships. We designed questions on a possible relationship change after the participants were asked to use the friendsourcing tool several times. In general, most participants (5 out 10 requesters and 7 out of 8 helpers) thought the friendsourcing tool would bring them closer in their relationship (see figure 4). H3 believed this tool could give them *"more contact opportunities,"* and there is *"value of knowing that I helped my sister."* Some participants thought the tool would not change their relationship since they were already friends, and there were not *"many additional interactions"* (R5). Also, R5 pointed out this tool's social value, which could raise the awareness of sighted people that PVIs need help in these everyday tasks.

In addition, we also added questions about the boundary and limit for helpers. Most of them indicated a number between 5 to 8 times a day that they would feel comfortable solving tasks for friends or strangers who need help. If the requests exceed this number, helpers would feel bothered, and if the request was from a friend, it would potentially cost their relationship to some extent. Comparing it to a question we asked in the pre-study interviews about how many times PVIs need to solve a CAPTCHA, the number was usually within 10 times weekly, indicating that solving tasks like CAPTCHA with low frequency but great difficulty might not bother their friends.

5.6 Other Findings

Besides answering the research questions we proposed, we also concluded other ad-hoc study findings based on both pre-study and post-study interviews.

Additional Effort. We asked both requesters and helpers whether this tool was accessible or time-consuming (see figure 4). All requesters found it accessible enough to use without additional effort. Most of them liked the audio feedback at different stages, including when the message has been sent, the helper started to work on this task, and the helper has successfully completed it. These notifications gave them useful signals to tell where's the helper at. For helpers, most of them did not think the tool was time-consuming because it usually took them less than one minute to solve the task. Some helpers gave a neutral answer to the time-consuming question. However, the reasons were (1) some reCAPTCHA tasks asked them to solve multiple challenges because they failed at the first one, and (2) some helpers expressed concerns on request times each day and afraid that others may request for help frequently. With some training on the reCAPTCHA tests and proper communication between PVIs and their friends, the helpers could avoid or eliminate these time-consuming concerns.

Comparison with Existing Tools. Prior to this study, most of the participants used remote assistance services or CAPTCHA-solving tools. They mentioned the similarities and differences between WebAlly and tools like BeMyEyes and WebVisum. The results showed the uniqueness of WebAlly, which could adopt helpers' direct visual perception and corresponding action into solving task-based CAPTCHAs. Services like BeMyEyes provided a crowdsourcing solution to gather descriptive guidance. WebVisum, one of the earliest methods designed for solving visual CAPTCHAs, also provided a solution to solve visual CAPTCHAs independently. However, many participants mentioned that it stopped maintenance long ago, and they have been relying on friends physically around them to help them solve Google reCAPTCHAs.

Furthermore, although Google reCAPTCHA provided an audio alternative, most requesters (9 out of 10) mentioned that audio CAPTCHAs are not always accessible. It is highly likely that they still need help from a person. Based on our pre-study interview, a typical flow for PVIs to solve a Google reCAPTCHA is: (1) See if they can pass without solving challenges, and (2) when they were signing up for new accounts or did not logged into Google account, they would still face a challenge and try to solve the audio version, and (3) in many cases participants can not get the right words and failed to identify themselves as humans, so they would (4) Ask friends around them to click the right tile images for them. Half of the requesters mentioned that they could not solve the challenge if there were no friends around. This result highlights the practical value of a tool like WebAlly.

These results also indicate that WebAlly could sufficiently serve as a backup plan when they do not have other options. However, most requesters declared that they would prefer to use the friendsourcing tool first if such a tool is available publicly. By using an online friendsourcing assistive tool, PVIs could save the effort of trying audio CAPTCHAs (which they may fail to pass) and ask people around them for help.

PVIs' independence in solving challenges. In comparison to other requesters' acceptance of friendsourcing, one requester specifically mentioned that it is preferred to try solving the challenge independently first rather than asking for help. This finding also implies further considerations on how friendsourcing tools would affect PVIs' feeling of independence. We designed WebAlly as an accessible complement, not a replacement to the existing reCAPTCHA mechanism.

6 DESIGN IMPLICATIONS

Based on our study findings, we provide several practical implications for designing more accessible CAPTCHAs and friendsourcing tools to solve them.

6.1 For CAPTCHA Designers

Make Tasks Transferable. In our studies, the response time of WebAlly can be delayed because it was a proof-of-concept system built upon Google reCAPTCHA. As CAPTCHA tasks were proven to be inaccessible even for the audio ones, designers should consider an accessibility feature of transferring tasks to their pre-registered friends or trained workers, so that the response time would be much shorter. In addition, designers can further expand on transferring more than the visual challenge itself. Transferring the task of identifying humans from robots to PVIs' friends or trained workers can also be feasible. By adding similar features in remote assistance services like BeMyEyes, the other party of human can identify whether the requester is a human via some voice or short video message attached by the visually-impaired requester. This has also been observed in some study sessions, where requesters send some voice message via another channel and inform the helper that a request is coming.

Trade-off Between Security and Accessibility. CAPTCHA was originally designed to distinguish humans from online bots. Making CAPTCHAs more accessible for PVIs would also introduce security-related issues. Making tasks transferable would also compromise the entire mechanism if robots can easily trigger this request. Designers should specifically consider the trade-off between PVIs' online security and accessibility.

6.2 For Friendsourcing Tools

Our paper majorly focused on applying friendsourcing to help PVIs solve Google reCAPTCHA. However, our findings also gave implications on designing friendsourcing tools for PVIs to solve other small online tasks that might not be accessible enough. For instance, when posting images to social media or preparing a video conference, PVIs need to check the image or their surroundings for sensitive information. Friendsourcing tools can be helpful in such scenarios.

Give Option of Sourcing Help. Our study findings revealed that most PVIs would like to have a mixed option towards sourcing help from the crowd or their friends. Giving them the ability to choose would also increase their sense of independence.

Maintain Active Friend List. Most PVI participants in our study have more than one close friends who were available to help. An active friend list would help increase the chance of tasks getting picked up and solved within the time limit. If the current friend did not respond in a few seconds, the request would be transferred to the next available friend on the list. The tool could also detect websites that frequently ask users to solve specific challenges and check friends' availability after users enter this website, and before they need to solve the challenge.

Explore Different Forms of Compensation. Our findings also proved that users tend to compensate the helpers with extrinsic rewards rather than monetary payments when asking for small favors like solving a Google reCAPTCHA. Designers should also consider different compensation systems to fulfill the user's needs in this. If the system uses monetary payments for compensation, a regularly paid subscription service is also more acceptable by users.

7 DISCUSSION

7.1 Limitation

To run the lab study, we developed a research prototype limited to the Google reCAPTCHA challenge and on the Chrome browser only. However, we believe that the Google CAPTCHA challenge is one of the most popular CAPTCHAs and could be an ideal example of how friendsourcing works in small daily tasks. Through some technical implementation, our framework and the tool could be easily extended to different scenarios that are challenging to PVIs.

For the methodology of our study, we did not ask PVIs to solve CAPTCHA themselves as a controlled experiment, which could make the session much longer and exhaust the participants. However, since our tool was meant to be a complement rather than a replacement to current solutions, direct quantitative comparison in performance became less important. Also, we asked both requesters and helpers to imagine themselves in hypothetical scenarios that either request or offer helps using friendsourcing or crowdsourcing, and we also asked them to imagine they were paying or getting paid in different controlled experiments. However, in real life, there are more conditions with other factors like friends' availability. Moreover, the hypothetical conditions might be limited since participants were not experiencing it in real life. For instance, even if researchers have randomized conditions to neutralize the possible affections of learning curves and preset minds, both requesters and helpers might still unconsciously believe they are doing the tasks with their friends or families. In this case, we are unsure whether people would behave similarly, given the same scenario in their daily lives. Furthermore, our study is also limited due to the hardship of recruiting participants. Coordinating studies with PVIs and their friends or families at the same time was extremely challenging. Our study also showed a possible effect of cultural differences, which suggested another limitation since our participants were majorly from the US and India.

Besides, the WebAlly system requires the interdependence of PVIs and their friends, which might affect PVIs' feeling of independence in solving day-to-day tasks. However, when PVIs need to solve a task-based visual CAPTCHA, they would still need to ask for help using a hacking CAPTCHA tool, a crowdsourcing, or a friendsourcing method to pass this challenge unless the CAPTCHA's mechanism changes for PVIs. Also, our tool serves as an additional complement to maintain PVIs' independence as much as possible.

7.2 Future Work

The tool could expand to two parts in the future, one of which is to improve the current functionality and bring more convenience to both requesters and helpers in the way as a multi-functional platform does. For instance, we will optimize the server response time by improving the technical framework to provide a better user experience. We would also integrate helpers' usage and management into the system so that helpers could register and increase their trust in the tool rather than randomly picking up text messages. Based on our study results, it is also necessary to implement the participants' suggestions, including making the availability transparent, adding a friend list to increase the response rate, and providing a crowdsourcing alternative when PVIs need to.

On the other hand, the tool can help people with visual impairments in other tasks. For instance, future researchers and investigators might be interested in helping PVIs screen images for private information before posting to social media or check surroundings before attending some virtual meetings. By combining machine learning algorithms and friend-sourced human intellect, the tool could efficiently detect privacy-related information in the given visual materials.

8 CONCLUSION

To help PVIs solve task-based visual CAPTCHAs that frustrate and encumber their daily web use, we designed a friendsourcing tool called WebAlly to involve friends' direct help in solving Google reCAPTCHA challenges. We proved through a lab study that this novel tool is accurate and fast for solving the tasks, and also doing well in protecting PVIs' privacy and security by using image processing features. We also used a factorial design to explore their preferences further and compared how different sources and compensation types could impact PVIs and their friends' perceived feelings. In conclusion, our tool could serve as a preferred alternative for PVIs when they cannot solve reCAPTCHAs independently. PVIs and their helpers also think that doing small favors would have a positive impact on their relationships. In the future, we want to study further how the tool works in a real-life setting when friends are not always available and investigate and for other small daily tasks.

REFERENCES

[1] Dina A Abdrabo, Tarek Gaber, and M Wahied. 2016. Assistive Technology Solution for Blind Users Based on Friendsourcing. In *The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015, Beni Suef, Egypt*. Springer, 413–422.

[2] Tousif Ahmed, Roberto Hoyle, Kay Connelly, David Crandall, and Apu Kapadia. 2015. Privacy Concerns and Behaviors of People with Visual Impairments. In *CHI2015*. 3523–3532. https://doi.org/10.1145/2702123.2702334

[3] Tousif Ahmed, Apu Kapadia, Venkatesh Potluri, and Manohar Swaminathan. 2018. Up to a Limit? Privacy Concerns of Bystanders and Their Willingness to Share Additional Information with Visually Impaired Users of Assistive Technologies. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–27.

[4] Tousif Ahmed, Patrick Shaffer, Kay Connelly, David Crandall, and Apu Kapadia. 2016. Addressing Physical Safety, Security, and Privacy for People with Visual Impairments. In *SOUPS2016*.

[5] TASLIMA AKTER, TOUSIF AHMED, APU KAPADIA, and MANOHAR SWAMINATHAN. [n.d.]. Privacy Considerations of the Visually Impaired with Camera Based Assistive Technologies: Misrepresentation, Impropriety, and Fairness. ([n. d.]).

[6] Taslima Akter, Bryan Dosono, Tousif Ahmed, Apu Kapadia, and Bryan Semaan. 2020. " I am uncomfortable sharing what I can't see": Privacy Concerns of the Visually Impaired with Camera Based Assistive Applications. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*.

[7] Paul Baecher, Niklas Büscher, Marc Fischlin, and Benjamin Milde. 2011. Breaking reCAPTCHA: a holistic approach via shape recognition. In *IFIP International Information Security Conference*. Springer, 56–67.

[8] Daniel Robert Bateman, Erin Brady, David Wilkerson, Eun-Hye Yi, Yamini Karanam, and Christopher M Callahan. 2017. Comparing crowdsourcing and friendsourcing: a social media-based feasibility study to support Alzheimer disease caregivers. *JMIR research protocols* 6, 4 (2017), e56.

[9] Cynthia L Bennett, Erin Brady, and Stacy M Branham. 2018. Interdependence as a frame for assistive technology research and design. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. 161–173.

[10] Michael S Bernstein, Desney Tan, Greg Smith, Mary Czerwinski, and Eric Horvitz. 2008. Personalization via friendsourcing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 2 (2008), 1–28.

[11] Rudy Berton, Ombretta Gaggi, Agnieszka Kolasinska, Claudio Enrico Palazzi, and Giacomo Quadrio. 2020. Are CAPTCHAs preventing robotic intrusion or accessibility for impaired users?. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 1–6.

[12] Jeffrey P Bigham and Anna C Cavender. 2009. Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1829–1838.

[13] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and others. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 333–342. http://dl.acm.org/citation.cfm?id=1866080

[14] Kevin Bock, Daven Patel, George Hughey, and Dave Levin. 2017. unCaptcha: a low-resource defeat of recaptcha's audio challenge. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*.

[15] Richard E Boyatzis. 1998. *Transforming qualitative information: Thematic analysis and code development.* sage.

[16] Erin Brady, Meredith Ringel Morris, and Jeffrey P Bigham. 2014. Friendsourcing for the greater good: perceptions of social microvolunteering. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

[17] Erin L Brady, Yu Zhong, Meredith Ringel Morris, and Jeffrey P Bigham. 2013. Investigating the appropriateness of social network question asking as a resource for blind users. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 1225–1236.

[18] Anjali Avinash Chandavale, Ashok M Sapkal, and Rajesh M Jalnekar. 2009. Algorithm to break visual CAPTCHA. In *2009 Second International Conference on Emerging Trends in Engineering & Technology*. IEEE, 258–262.

[19] Bryan Dosono, Jordan Hayes, and Yang Wang. 2015. "I'm Stuck!": A Contextual Inquiry of People with Visual Impairments in Authentication. In *Symposium on Usable Privacy and Security (SOUPS)*.

[20] Anthony Dunne and Fiona Raby. 2013. *Speculative everything: design, fiction, and social dreaming.* MIT press.

[21] Valerie Fanelle, Aditi Shah, Sepideh Karimi, Bharath Subramanian, and Sauvik Das. [n.d.]. Blind and Human: Exploring More Usable Audio CAPTCHA Designs. *people* 17, 28 ([n. d.]), 32–34.

[22] Jennifer LoCascio Gauvreau. 2017. Accessibility gotchas with CAPTCHAs. *Journal of Digital & Social Media Marketing* 5, 4 (2017), 379–390.

[23] Danna Gurari, Qing Li, Chi Lin, Yinan Zhao, Anhong Guo, Abigale Stangl, and Jeffrey P Bigham. 2019. VizWiz-Priv: a dataset for recognizing the presence and purpose of private visual information in images taken by blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 939–948.

[24] Jordan Hayes, Smirity Kaushik, Charlotte Emily Price, and Yang Wang. 2019. Cooperative Privacy and Security: Learning from People with Visual Impairments and Their Allies. In *Proceedings of Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. https://www.usenix.org/conference/soups2019/presentation/hayes

[25] Kieran Healy. 2010. *Last best gifts: Altruism and the market for human blood and organs*. University of Chicago Press.

[26] J. Holman, J. Lazar, and J. Feng. 2008. Investigating the Security-related Challenges of Blind Users on the Web. In *Designing Inclusive Futures*, Patrick Langdon BSc, CEng John Clarkson MA MIEE, and Peter Robinson MA Ceng (Eds.). Springer London, 129–138. https://doi.org/10.1007/978-1-84800-211-1_13

[27] Mohit Jain, Rohun Tripathi, Ishita Bhansali, and Pratyush Kumar. 2019. Automatic Generation and Evaluation of Usable and Secure Audio reCAPTCHA. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 355–366.

[28] Kiranjot Kaur and Sunny Behal. 2014. CAPTCHA and its techniques: a review. *International Journal of Computer Science and Information Technologies* 5, 5 (2014), 6341–6344.

[29] Sushama Kulkarni and Hanumant Fadewar. 2018. Audio CAPTCHA Techniques: A Review. In *Proceedings of the Second International Conference on Computational Intelligence and Informatics*. Springer, 359–368.

[30] Jonathan Lazar, Jinjuan Feng, Tim Brooks, Genna Melamed, Brian Wentz, Jon Holman, Abiodun Olalere, and Nnanna Ekedebe. 2012. The SoundsRight CAPTCHA: an improved approach to audio human interaction proofs for blind users. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2267–2276.

[31] Kaitlin Mahar, Amy X Zhang, and David Karger. 2018. Squadbox: A tool to combat email harassment using friendsourced moderation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[32] Il'dar Nailevich Manashev. 2018. Breaking Google reCAPTCHA v2. *Prikladnaya Diskretnaya Matematika. Supplement* 11 (2018), 99–101.

[33] Lourdes Moreno, María González, and Paloma Martínez. 2014. CAPTCHA and Accessibility. *Is this the best we can do* (2014).

[34] Greg Mori and Jitendra Malik. 2003. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, Vol. 1. IEEE, I–I.

[35] Meredith Ringel Morris, Kori Inkpen, and Gina Venolia. 2014. Remote shopping advice: enhancing in-store shopping with social technologies. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 662–673.

[36] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. What do people ask their social networks, and why? A survey study of status message Q&A behavior. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1739–1748.

[37] Donatella Pascolini and Silvio Paolo Mariotti. 2012. Global estimates of visual impairment: 2010. *The British Journal of Ophthalmology* 96, 5 (May 2012), 614–618. https://doi.org/10.1136/bjophthalmol-2011-300539

[38] Frances L Rapport and CJ Maggs. 2002. Titmuss and the gift relationship: altruism revisited. *Journal of advanced nursing* 40, 5 (2002), 495–503.

[39] Software. Retrieved. 2020-10-15. Anti Captcha: captcha solving service. Bypass reCAPTCHA, FunCaptcha, image captcha, GeeTest, hCaptcha. https://anti-captcha.com/

[40] Software. Retrieved. 2020-10-15. Be my eyes - see the world together. https://www.bemyeyes.com/

[41] Software. Retrieved. 2020-10-15. Buster: captcha solver for humans. https://chrome.google.com/webstore/detail/buster-captcha-solver-for/mpbjkejclgfgadiemmefgebjfooflfhl

[42] Software. Retrieved. 2020-10-15. Eyecoming website. https://www.eyecoming.com/english/

[43] Software. Retrieved. 2020-10-15. Geetest captcha | demo. https://www.geetest.com/en/demo

[44] Software. Retrieved. 2020-10-15. Jaws® – freedom scientific. https://www.freedomscientific.com/products/software/jaws/

[45] Software. Retrieved. 2020-10-15. Nv access. https://www.nvaccess.org/

[46] Software. Retrieved. 2020-10-15. Recaptcha. https://www.google.com/recaptcha/about/

[47] Software. Retrieved. 2020-10-15. Webvisum: download. https://www.webvisum.com/en/main/download

[48] Jeffrey M Rzeszotarski and Meredith Ringel Morris. 2014. Estimating the social costs of friendsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2735–2744.

[49] Sajad Shirali-Shahreza and M Hassan Shirali-Shahreza. 2011. Accessibility of CAPTCHA methods. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. 109–110.

[50] Suphannee Sivakorn, Jason Polakis, and Angelos D Keromytis. 2016. I'm not a human: Breaking the Google reCAPTCHA. *Black Hat* (2016), 1–12.

[51] Online Survey. [n.d.]. Webaim: screen reader user survey #7 results. https://webaim.org/projects/screenreadersurvey7/

[52] Jennifer Tam, Jiri Simsa, David Huggins-Daines, Luis Von Ahn, and Manuel Blum. 2008. Improving audio captchas. In *Symposium On Usable Privacy and Security (SOUPS)*.

[53] Kathleen D Vohs, Nicole L Mead, and Miranda R Goode. 2006. The psychological consequences of money. *science* 314, 5802 (2006), 1154–1156.

[54] Michael Walzer. 2008. *Spheres of justice: A defense of pluralism and equality.* Basic books.

[55] Jacob O Wobbrock, Shaun K Kane, Krzysztof Z Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing (TACCESS)* 3, 3 (2011), 1–27.

[56] Huichuan Xia and Brian McKernan. 2020. Privacy in Crowdsourcing: a Review of the Threats and Challenges. *Computer Supported Cooperative Work (CSCW)* (2020), 1–39.

[57] Jeff Yan and Ahmad Salah El Ahmad. 2007. Breaking visual captchas with naive pattern recognition algorithms. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*. IEEE, 279–291.

[58] Yuan Zhou, Zesun Yang, Chenxu Wang, and Matthew Boutell. 2018. Breaking google recaptcha v2. *Journal of Computing Sciences in Colleges* 34, 1 (2018), 126–136.

[59] Haiyi Zhu, Sauvik Das, Yiqun Cao, Shuang Yu, Aniket Kittur, and Robert Kraut. 2016. A market in your social network: The effects of extrinsic rewards on friendsourcing and relationships. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 598–609.