Let The Right One In: Attestation as a Usable CAPTCHA Alternative

Tara Whalen, Thibault Meunier, and Mrudula Kodali, *Cloudflare Inc.;* Alex Davidson, *Brave;* Marwan Fayed and Armando Faz-Hernández, *Cloudflare Inc.;* Watson Ladd, *Sealance Corp.;* Deepak Maram, *Cornell Tech;* Nick Sullivan, Benedikt Christoph Wolters, Maxime Guerreiro, and Andrew Galloni, *Cloudflare Inc.*

https://www.usenix.org/conference/soups2022/presentation/whalen

This paper is included in the Proceedings of the Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022).

August 8–9, 2022 • Boston, MA, USA

978-1-939133-30-4

Open access to the Proceedings of the Eighteenth Symposium on Usable Privacy and Security is sponsored by USENIX.

Let The Right One In: Attestation as a Usable CAPTCHA Alternative

Tara Whalen, *Cloudflare Inc.*
Thibault Meunier, *Cloudflare Inc.*
Mrudula Kodali, *Cloudflare Inc.*
Alex Davidson, *Brave*
Marwan Fayed, *Cloudflare, Inc.*
Armando Faz-Hernández, *Cloudflare Inc.*
Watson Ladd, *Sealance Corp.*
Deepak Maram, *Cornell Tech*
Nick Sullivan, *Cloudflare Inc.*
Benedikt Christoph Wolters, *Cloudflare Inc.*
Maxime Guerreiro, *Cloudflare Inc.*
Andrew Galloni, *Cloudflare Inc.*

Abstract

CAPTCHAs are necessary to protect websites from bots and malicious crawlers, yet are increasingly solvable by automated systems. This has led to more challenging tests that require greater human effort and cultural knowledge; they may prevent bots effectively but sacrifice usability and discourage the human users they are meant to admit. We propose a new class of challenge: a Cryptographic Attestation of Personhood (CAP) as the foundation of a usable, pro-privacy alternative. Our challenge is constructed using the open Web Authentication API (WebAuthn) that is supported in most browsers. We evaluated the CAP challenge through a public demo, with an accompanying user survey. Our evaluation indicates that CAP has a strong likelihood of adoption by users who possess the necessary hardware, showing good results for effectiveness and efficiency as well as a strong expressed preference for using CAP over traditional CAPTCHA solutions. In addition to demonstrating a mechanism for more usable challenge tests, we identify some areas for improvement for the WebAuthn user experience, and reflect on the difficult usable privacy problems in this domain and how they might be mitigated.

1 Introduction

In a CAPTCHA challenge, a client is presented with a human-targeted puzzle requiring an interaction that no algorithm should be able to provide. A puzzle solved correctly is understood to be a puzzle solved by a human.

In practice, the association between puzzle and person has been broken by advancements in machine learning and artificial intelligence techniques that solve CAPTCHAs with high degrees of accuracy [39]. In response, new CAPTCHAs emerge with increasingly specific (or challenging) signals and characteristics to distinguish human users from bots. The natural consequence of puzzles that focus on very specific traits of "humanness" is a set of laborious tests that can be solved by a decreasing number of humans [22]. This creates a cycle of increasing user frustration.

How, then, can the burden of proof that a client is not a bot be reduced for the human user? One approach reduces the number of challenge-response tests by extensive server-side user behaviour modeling and analysis [26]. This is accomplished with the use of cookies to track and profile users, alongside automated tests such as canvas rendering [35]. These tools are used to fingerprint client behaviour at the cost of privacy.

Alternatively, we can revisit the question: How can a human prove that they are not a program? The motivation to do so stems from two observations. First, CAPTCHA challenges are fundamentally connected to a design [38] born in a decades-old Internet ecosystem, more culturally homogeneous and with less capable hardware and software. Second, today's Internet infrastructure consists of, indeed relies upon, cryptographic constructs and systems. Remote attestation is one such bedrock of increasing importance to Internet systems and protocols [31]. In cryptography, remote attestation involves supplying evidence to an appraiser over a network, in support of a claim about the properties of a target [10].

In this paper we explore remote attestation as the foundation for a new class of challenge-response that can attest to the presence of a person. Rather than identify tasks that bots are incapable of completing, our focus shifts to tasks that a human can complete. Thus we ask the following question: *what is the smallest task that separates a human from a bot?* The answer, we claim, is a *physical* interaction such as a touch or a look. We note that support for such interactions

USENIX Symposium on Usable Privacy and Security (SOUPS) 2022.
August 7–9, 2022, Boston, MA, United States.

is increasingly ubiquitous on even lower-end mobile phones and computers via *privacy-preserving* biometric sensors, and is additionally supported by USB and NFC hardware keys. These are authenticator devices that "attest" to the interaction. Their functionality is also widely accessible via the World Wide Web Consortium's (W3C) Web Authentication API (WebAuthn) [23].

Motivated by these observations, we architected and implemented a challenge in which the response is a cryptographic and WebAuthn-compliant attestation. We note that WebAuthn functionality is increasingly available on the lower-end devices that are the primary means for connecting to the Internet for most of the world's population [9, 32, 20]. Our design is guided by the W3C guidelines and requirements for replacing CAPTCHAs, with privacy-preservation made an explicit priority [22].

We evaluated the feasibility of our WebAuthn-based challenge, called the "Cryptographic Attestation of Personhood" (CAP), through a set of user studies. After a pilot study using USB security keys, we created a demo compatible with a wider range of hardware and released it for public testing and feedback. We found that, given the required hardware and browser environment, users were able to quickly and easily pass a challenge, and most said they were likely to use CAP if it were available as an option.

Our results were drawn from an analysis of 1896 sessions in which users tried our CAP challenge, testing it with their own hardware; a subset of these users (n=93) provided additional details via a survey. In our demo evaluation, a large proportion of users were able to complete the CAP challenge, with approximately half of the attempts being successfully passed. Task completion was quick, at 10.6 seconds—approximately half the time needed to solve a picture-selection CAPTCHA. Our survey results indicated that the majority of respondents (75%) were likely to use CAP when possible.

Overall, CAP shows great promise as a usable CAPTCHA alternative, although there are some barriers to adoption. These include privacy concerns (which are a challenge for WebAuthn in general); the difficulty of clear communication; and inconsistencies across different browsing environments.

2 Background: Users vs. CAPTCHAs

CAPTCHAs have been routinely identified as problematic by both researchers and others in the wider technical community [22]. The first CAPTCHA defined the puzzle as a challenge-response mechanism that involves a user and a challenge provider [38] (most often a content server or service). The puzzle has one requirement: a correct solution should assure the provider of an interaction that only a human could have performed. Interactions that could be completed by bots and algorithms are excluded by definition. The definition and intention notwithstanding, automated solvers have since emerged [34], prompting increasingly complex CAPTCHAs that place ever-higher demands on people to solve them.

One major problem is accessibility. CAPTCHA tasks frequently involve visual identification, which makes them unusable by users with visual impairments [22]. Audio recognition tasks [15] may be an improvement for some user needs but still demand a heavy task burden. In addition, many task types rely on language or cultural knowledge that is far from universal. This can create barriers—for example, if taxicabs in the images look nothing like those in the user's country [13] or for users who have never seen a fire hydrant. Mathematics, seemingly universal, is a far from trivial type of challenge for many users [18].

Privacy is another area of concern. For example, reCAPTCHAv3 calculates an "adaptive risk analysis" to assess the likelihood that a site visitor is a human, and may refrain from presenting a task if there is high enough score [26]. These approaches rely on background data collection— the specific details of which are rarely made public [33, 19]. In this context, some loss of privacy may be unavoidable, despite being undesirable.

The reliability of CAPTCHAs increasingly suffers in response to AI algorithms that continue to improve. Levels of complexity have been added to tests in response, as well as server-side tracking and profiling mechanisms to reduce their appearance to users. Reliability is an important requirement as any test that insufficiently prevents a bot from solving it has little utility as a security mechanism in the Internet setting. This worsens, in turn, accessibility. Among audio CAPTCHAs, for example, the gap between human and robot performance has shrunken dramatically, with bots reporting higher scores than humans [37, 2, 36].

In contrast to the available set of CAPTCHAs, our hardware challenge establishes proof of personhood with no cognitive burden and relies instead on a minimal set of possible physical interactions. The criteria, the components, and overall architecture are presented in the next section.

3 A WebAuthn challenge architecture

In this section, we describe a challenge platform with the Web Authentication standard's API for attestation [23]. The platform is intended to be easily deployable so that smaller service providers can benefit.

3.1 Design requirements

Our design is guided by work at the W3C [22] and the experience with CAPTCHAs at Cloudflare, a service that provides

security features, including bot management, for a large proportion of the Internet [1]. Based on these, we believe any proof of a person attached to a device must meet the following goals:

1. **Ephemerality:** Solutions cannot be precomputed.
2. **Browser-based:** The challenge task must work in the browser without client modifications.
3. **Usability:** Internet-using humans should be able to prove their proximity to the device with minimal burden.
4. **Integrity:** The task has no solution without a human, otherwise the task fails to ensure security.

Standard CAPTCHAs clearly adhere to the two criteria of being *ephemeral* and *browser-based*. Each puzzle is randomly generated, and usually consists of a visual or audio challenge that can be displayed in an Internet browser. However, CAPTCHAs often fail to adhere to *usability* and *integrity*, as previously discussed.

In response to the diminishing *integrity* of CAPTCHAs, tools such as reCAPTCHAv3 [26] use sophisticated server-side modelling of client behaviour and anomaly detection. In some cases, this may preserve a degree of usability, but transforms the independent presentations of a challenge across websites into a connected web of user tracking, and motivates an additional requirement:

5. **Privacy:** Tests and challenges should reveal no information about users, nor be substitute identifiers.

We note that *privacy* is one attribute in which CAPTCHAs excel if executed in isolation. Absent the extra analytics pipelines that are, or can be, built on top of them, there is no information to tie a puzzle solution to an identity. Given the Internet context that we are operating in, the main privacy considerations that we examine in this work relate to ensuring that user identities are never revealed. In addition, we regard as unacceptable any challenge framework that can track users across visits. Even in situations where a user's identity is never directly revealed, the presence of such tracking potential may be used to identify the user via other means.

3.2 A challenge that trusts cryptographic attestation of human signals

We propose that one simple task that can differentiate a human from a bot is a *physical* interaction. Interactions may include biometric verification of a fingerprint or face, or a registering a touch on a secure hardware key. In this context an interaction challenge is deferred to a trusted platform to correctly and cryptographically *attest* to some attribute or action. This idea is the bedrock of trusted computing platforms.

Internet browsers have recently acquired the interfaces needed to support cryptographic attestation, which are exposed via the W3C's Web Authentication standard (WebAuthn) [23]. The WebAuthn protocol is supported in all major Internet browsers [12]. It is also supported by many authenticator devices, including FIDO-supporting touch hardware keys, as well as biometric sensors increasingly available in Android, iOS, macOS, and Windows devices.

The WebAuthn protocol consists of two information flows, one for registration and another for authentication. The authentication flow is used to log in to an account without a password after an account has been created or registered. For our purposes, the authentication flow is ignored, thus there is no account against which to authenticate. Our design *relies solely on the attestation flow*. The attestation flow is similar to the registration flow (see Figure 1), but omits information that would bind an account to a user, such as an email address or name. Since there is no account-related information, there is no relationship between an account and a user for the attestation to expose.

We instead isolate the cryptographic attestations from within the WebAuthn framework's standard registration flow, as depicted in Figure 1. The standard flow has three high-level stages: (i) A server first requests an attestation challenge from a client in response to a username; (ii) the client then requests a credential from the WebAuthn-supported device, for which a person must take an action; (iii) the client receives a credential containing a proof of the action (usually, an attestation in the form of a digital signature), and sends it to the server, where the attestation is verified and stored.

Our changes omit the first and last stages of the WebAuthn standard registration flow. Figure 1 shows the standard flow, with greyed boxes depicting our omissions. The standard registration process is initiated by a username. Instead, our challenge is initiated by the server, which requests an attestation from the client without being prompted by a username. Note that this invocation is otherwise a standard WebAuthn registration interaction. During the last stage, the public keys are discarded once the attestation is verified, in contrast to their being stored after registration. These omissions preserve the integrity of the attestation itself.

In our challenge platform, the contents of the challenge string include a timestamp to limit the validity period of the response, together with information about the browser and user such as IP address, enabled Javascript APIs, etc. This prevents use of the response from any user agent in subsequent scripted interactions. Furthermore, successful completion of this or any other challenge to prove humanity only grants access if other aspects of the request are consistent with human interaction.

We emphasize that the cryptographic elements of the flow are untouched, so security aspects are preserved. Conversely, the

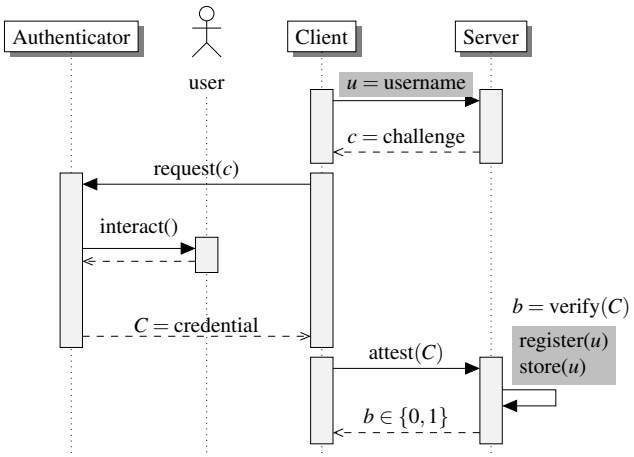

Figure 1: A high-level overview of the WebAuthn registration flow, with the minor omissions that enable our challenge: Portions encapsulated in grayed boxes are required for registration, and unnecessary for attestation verification. Ignoring the registration components preserves the privacy of users. Our challenge flow is otherwise identical to the standard.

omissions from the typical WebAuthn flow pertain *only* to user data. The deviations from the exact specification of the protocol leave the attestation and its verification untouched. Our hardware challenge is then characterized by the following properties:

- No user data is stored at the server.
- There are no user identifiers: users never specify a username, display names are replaced with generic text and unique IDs with random values that go unused.
- Attestations are provided directly by authenticators to ensure that they can be validated.

The availability of WebAuthn as a web API among Internet browsers enables us to build a human attestation system with the same *ephemerality* provided by a CAPTCHA. It is instructive to revisit the ability of our challenge to fulfill the remaining design goals, below and summarized in Table 1.

Usability Our WebAuthn challenge supports the same set of devices as does the W3C standard API, including Apple and Android biometric sensors and hardware security keys. The user gesture, such as presenting one's face or touching a USB key, was expected to be easier to perform than a CAPTCHA interaction, and fits the profile of CAPTCHA alternatives envisioned in the recent W3C technical report highlighting CAPTCHA inaccessibility [22]. The usability assessment forms the bulk of this paper, in which we confirmed that this interaction was quick and easy in the majority of cases for which users had the necessary hardware and web-browsing environment.

Table 1: Design requirements comparison between our approach and CAPTCHAs.

Challenge	Usability	Security	Privacy
CAPTCHA	✗	✓[1]	✓[2]
Hardware attestation	✓[3]	✓	✓

[1] Reliant on continual upgrade of CAPTCHA challenges to prevent attacks from bots of ever-increasing capability.
[2] Only for those CAPTCHAs that do not use wider user browsing analytics to make inferences on the user's humanness.
[3] Usability is ensured for those that own applicable hardware.

The drawback of using this approach as a challenge is somewhat obvious: it is only available to those individuals with applicable hardware that implements the WebAuthn standard. As mentioned previously, WebAuthn is currently supported by a variety of devices including security keys, smartphones, and personal computing devices. With this in mind, we believe that it is reasonable to expect that WebAuthn, and Internet-based hardware attestation, will become more prevalent across the globe in the near future.

Integrity Our challenge should be difficult for bots to bypass. The integrity of the interaction is tied to the integrity of the WebAuthn standard, and devices' ability to maintain keys securely. The attestation is generated by the device using a secret key that is embedded in hardware, tamper-resistant, and can only be extracted manually. Such a task is engineered to be difficult by design. Notably, the secret key is embedded in a batch during manufacture across a cohort of devices. In this manner, a batch key is shared, for example, among the same device model or devices manufactured in the same year. The key batching makes it possible for the challenge provider to only accept attestations from selected classes of devices. Similarly, attestations for device classes can be revoked if they fail some set of criteria, or if keys are known to have leaked. One weakness in touch devices *may* be that they can be circumvented by an automated physical device[1], against which biometric sensors are resilient.

Privacy Any viable challenge solution must reveal no details about the user identity, nor provide avenues for tracking the user across multiple websites or challenges. Our challenge reduces the registration to an attestation that is non-specific to the user. As a result, the attestation reveals no personally identifiable information. However, each attestation does reveal a hard-coded certificate associated with the device class. If the certificate were unique, it could be used to track a user's

[1] See, for example, https://bert.org/2020/10/01/pressing-yubikeys/.

attestation across multiple challenges and make inferences about that user's browsing patterns.

Fortunately, the expected privacy impact incurred by revealing this certificate is very small, as described in the standardisation document [23, Section 14.4.1]. The standard recommends that these certificates (and their associated cryptographic keys) are batched and shared across multiple manufactured devices. The result is that each user belongs to a large *anonymity set*, as no given hardware device can be identified by the revealing of this certificate alone. For example, the FIDO UAF standard [4] requires that at least 100000 authenticator devices share the same attestation certificate in order to produce sufficiently large groups. (When considering mobile device classes we expect the anonymity set to be orders of magnitude larger.) The knowledge revealed to a provider is limited to the type of device and its batch or model.

Note that the WebAuthn challenge proposed here is built on an open standard. This is not a proprietary solution, but can be deployed by anyone needing to roll out a human challenge in their systems. They can learn from our evaluation (detailed below), and adapt and extend this solution in the ways that are most suitable for their own requirements.

4 Pilot study

We explored the possibilities of our hardware attestation mechanism in the context of Cloudflare, which provided opportunities for real-world evaluation as well as the potential of large-scale deployment as a challenge solution for a substantial number of websites. As a starting point, we carried out a pilot study to assess whether our idea had merit, particularly in terms of its usability.

We evaluated our hardware attestation mechanism with a usability experiment, assessing effectiveness, efficiency, user satisfaction and gathering feedback about the overall user experience. We compared this hardware key method, using Yubico YubiKey 5 Series security keys, against a standard CAPTCHA method currently protecting millions of sites: hCaptcha [21]. hCaptcha presents a 3x3 grid of pictures and prompts the user to select a subset matching specific criteria (e.g., "Please click each image containing a boat").

In the experiment, 17 participants (Cloudflare employees) performed a simple webpage access task, where they visited two public webpages protected by hCaptcha or hardware attestation. For hCaptcha, participants identified objects from a set; for hardware attestation, they launched the proof-of-concept challenge and touched their YubiKey when prompted by their browser.

Each task was followed by a System Usability Scale (SUS) [7] satisfaction questionnaire. Participants were also provided with a post-session questionnaire to measure preference between the two methods during a short, closing debrief.

Results of the usability experiment We instrumented the testing environment used by our participants to record *errors*, measure *success rate* (task completion), and *time-on-task*. Effectiveness was high for both conditions: all participants successfully completed both the YubiKey and hCaptcha conditions with no errors. Our participants rated the hardware challenge as easier to use with an SUS score of 77.1, and hCaptcha with an SUS score of 65.3. For SUS scores, "better products scor[e] in the high 70s to upper 80s", and "[p]roducts with scores of less than 70. . . should be judged to be marginal at best" [5].

Measurements and analysis indicated significantly shorter completion times for the hardware challenge. A Wilcoxon signed-ranks test indicated a mean task time of 13.5 s for the hardware challenge, and 25.0 s for hCaptcha: $V=115$, $p < 0.001$. Note that the hardware challenge completion time is not just the time taken for the physical interaction with the key, but also includes the time taken to read and respond to informational pop-up messages spawned in-browser by the WebAuthn flow.

15/17 participants (88%) preferred the hardware challenge, with only one participant preferring hCaptcha and another having no preference. Participants who preferred the YubiKey expressed frustrations with CAPTCHAs and commented on the ease and speed of the YubiKeys. The two participants who did not prefer the YubiKey voiced concern and fear about security and privacy. Similar concerns were shared by some participants who favoured the hardware key. Participant feedback also identified wider user-communication challenges with browser prompts and messaging, where the information presented was viewed as uninformative or confusing.

5 Evaluation: Public demo study

The results of the preliminary user study indicated that our proposed solution was promising enough to develop further. We therefore developed a public demo of our "Cryptographic Attestation of Personhood" (CAP), which we deployed at Cloudflare for wider evaluation. Unlike our pilot study, which was limited to YubiKeys, the challenge on the demo site could be passed used a wide variety of hardware, such as biometric readers (e.g., Face ID and Touch ID) and multiple models of secure hardware keys. The site accepted any USB or NFC key certified by the FIDO Alliance, as long as it had no known security issues according to the FIDO Alliance Metadata service (MDS 3.0) [3]. A summary of supported hardware can be found in Table 2.

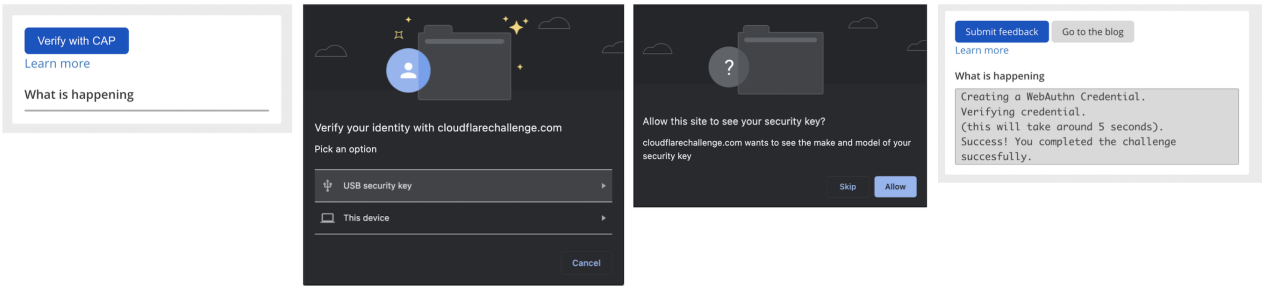

Figure 2: CAP demo: stages of CAP interaction, including browser WebAuthn prompts

5.1 Experiment details

We created a demo website where users could click on a button to "Verify with CAP", which would prompt them to complete the WebAuthn challenge, whose main stages are illustrated in Figure 2. The start panel (with the "Verify" button) was displayed on a web page; this panel included a "Learn More" link, which brought the user to a separate "Frequently Asked Questions" page for assistance. The space below the button, labelled "What is happening" displayed progress through the verification process, until its conclusion: success or failure.

Once the user clicked on the button, additional pop-up windows were spawned by their browser as part of the standard WebAuthn process. Note that the specific text and design of these pop-ups is determined by the browser, and not by CAP. The examples in Figure 2 are from Chrome v98 on MacOS 12. The first browser pop-up prompts the user to "verify your identity" on the Cloudflare demo website, and gives them a list of WebAuthn authenticator options. In this example, the user can pick from the built-in Touch ID sensor on their Macbook, or they could use a portable USB key. The user selects their preferred option, and performs the user gesture (e.g., touches the fingerprint sensor). Because an attestation has been requested for this WebAuthn interaction—as this is an integral part of CAP—a prompt is displayed that asks whether the user wishes to disclose the make and model of their security key to the site. If the user selects "allow", then the attestation is sent for verification; if it passes, then the user successfully passes the challenge (as shown in the final image in Figure 2). The user might also fail to pass, in which case an error message is shown in the panel. Technical details of the error are shown, and the user is informed "It seems there was an error completing the challenge! You can retry or share your feedback with us." After each challenge attempt, users have the option of clicking "Retry" or "Submit Feedback". The latter takes them to a user survey (described in 5.3 below).

This demo site was launched in conjunction with a blog post about CAP published on Cloudflare's blog, which often discusses new features and experiments being run [28]. It was expected that this post would spark readers' interest in trying out CAP, which would provide us with useful information. The blog post explained how the underlying WebAuthn technology worked, at both a non-expert and a more technical level for those who might prefer such details. The post included information about privacy considerations, which we anticipated would be of concern to users (and had been demonstrated in our preliminary study). The privacy explanation highlighted the size of the anonymity set (e.g., your key is indistinguishable from a large batch of others) and stressed how WebAuthn strictly limits what is sent to the server—for example, biometric data never leaves the device. The blog post concluded with a link to the demo site, and invited people to try out this experimental feature. In a later expanded version of the demo, with an associated blog post [14], when a user completed an attempt at a CAP verification, they had the option of giving feedback through a survey. This provided richer information on what they liked and disliked, general concerns, and suggestions for improvementc̃ite. We evaluated the demo version of CAP through a combination of data logged from online users and feedback gathered from the online survey. For each interaction with the CAP demo, timing and error data was logged.

Note that we adopted a minimal data collection approach for the data logging of these interactions. Because we were concerned that users might be uneasy about disclosing information when testing this new feature (despite the protections being provided), particularly given privacy sensitivities (e.g., biometrics), we strongly limited what we stored. This meant that we did not collect details such as browser user agent strings and did not store any information about the authenticator (such as make or model); this attestation information was not logged.

Ethics Institutional review boards (IRBs) are uncommon in most workplaces, including ours. Nonetheless, care was taken to follow appropriate experimental procedures throughout (e.g., obtaining user consent for participation and data collection). No identifying information was logged in the interaction

phase, only timing and error data related to each stage of the CAP process. (This also means that duplicates may occur in our dataset, since repeat visits could not be identified.) For the survey, respondents provided explicit consent and were not required to provide any identifying information. They were permitted to provide an email address if they wanted to be contacted for further studies; they also were given the separate option of providing details on their environment (such as their browser's user agent). They were also asked whether they consented to having their responses quoted, without attribution, in research publications. No participant compensation was offered.

5.2 Logged Interaction Data

We analyzed 1896 user sessions, collected over eight days. A single session was defined as any instance in which a user clicked on the "Verify with CAP" button at least once, which ended in a failed or successful verification, and include multiple attempts at verification (if any) within the same session.

5.2.1 Results

Completion Time For cases in which a person successfully validated with CAP at any point in a session, the mean completion time—from button click to completed validation—was 10.6 s. In the case of a failed validation, the mean time was 2.8 s. Failure is faster than success because the process terminated earlier without completing further steps; note that this also includes cases in which there is no further user response after the button click, which leads to a failure upon timeout (whose duration is environment-dependent).

For comparison purposes, we analyzed the time taken to complete real-world hCaptcha interactions (which could be from bots or humans), based on a sample of 8262 interactions recorded in Cloudflare's logs. (hCaptcha uses an object identification challenge involving a 3x3 picture grid [21].) The mean hCaptcha solving time was 24.5 s, over twice the time of a successful CAP challenge; this timing difference is statistically significant (Wilcoxon rank-sum test: $W = 1476154$, $p < 0.001$).

Success Rate Out of 1896 sessions, 919 (48.5%) included a successful validation, with the majority of these (818, 43%) having no errors. (Recall that a person could retry multiple times per session.) In most cases, people tried only once: in the 1078 sessions with at least one failed attempt, only 24% had more than one failed attempt. (Note that we do not have any details about users' environments in this dataset, owing to our minimal data collection in this part of the experiment.)

5.3 User Feedback: Survey

When a person completed a CAP validation attempt, they were given the option of completing a survey to provide additional feedback. This survey was deliberately kept brief, to encourage people to complete it, and focused on the key elements we wished to measure. We collected 93 survey responses during our evaluation period.

Likert scales The first set of questions asked for responses to 5-point Likert scales ("strongly agree" to "strongly disagree"):

- I am likely to use this when possible (I have a security key/biometric sensor)
- Assuming I have what I need, I prefer using this instead of a CAPTCHA
- It's frustrating how often I have to prove I'm a human
- I feel confident that this preserves my privacy

The Likert scale responses are shown in Figure 3.

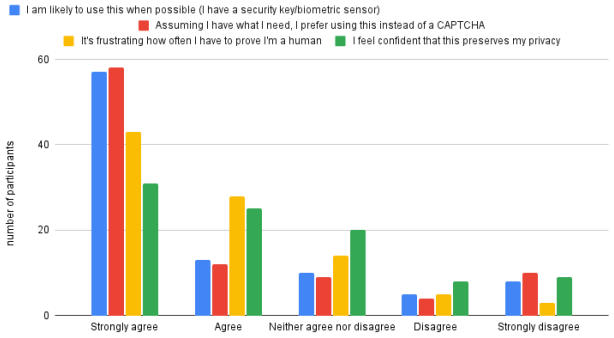

Figure 3: Results from 5-point Likert questions in CAP survey

The majority of respondents indicated they were likely to use CAP when their hardware allowed this option: 70 (75%) agreed or strongly agreed. Similarly, 70 respondents (75%) said they preferred CAP to a CAPTCHA (agreeing or strongly agreeing). Respondents indicated a high level of annoyance with having to complete human challenges, with 71 respondents (76%) agreeing or strongly agreeing that it was frustrating to do this task often. On the question of privacy, responses were more mixed. Respondents had some confidence in CAP's privacy protections, although this was not as high as for the other items: 56 respondents (60%) agreed or strongly agreed with the statement "I feel confident that this preserves my privacy"; a further 20 (22%) neither agreed nor disagreed.

Free-form responses Respondents could provide free-form responses to four further questions: (i) What do you like

the most about this? (ii) What one thing would you change about how this works? (iii) If you have any accessibility needs, please let us know how well or poorly this caters to those needs, and (iv) If there's anything else you'd like us to know, please tell us here.

If desired, the respondent could send information about their environment: browser user agent; hardware device issuer; attestation format and type. Additionally, we collected the time taken for their most recent verification attempt and the number of errors encountered during their session. The free-form responses provided us with greater detail on what users liked and disliked about CAP. These were manually coded, which involved three researchers collectively identifying a set of initial themes, then coding independently and finally comparing results to achieve consensus.

The most commonly cited strengths ("What do you like the most about this?") were ease of use; speed; and improvement over other types of challenges (e.g., traditional picture-selection CAPTCHAs):

- "Honestly, it's quite fast. Works great, while proving the same thing that regular captchas do" [P6]
- "this is much much quicker than selecting all the buses....and trucks. . . " [P15]
- "Passed the challenge with just my fingerprint. Very convenient." [P43]
- "Easy as ABC. Love it!" [P54]

There were a number of suggestions that people had about how to improve CAP, primarily around clarifying the communication; preventing errors and failures; and reducing UI pop-ups. On the theme of communication, respondents recommended improvements in explaining some aspects of CAP, primarily the privacy protections:

- "Maybe making it clearer that the model of your key doesn't go out to the internet?" [P6]
- "Maybe add some explanation of how this works, what information do you guys collect during this process"[P21]
- "will Cloudflare store my 2FA key?" [P39]
- "how is this not a unique identifier? and how are you gonna explain that this is not surveillance to 'the normie folks'?" [P91]

Others suggested a need for better explaining some of the WebAuthn process and components, which may be hard to understand:

- "The options that are available on Android can be overwhelming for a non-technical audience. Most people won't know what a Yubikey is or understand that 'unlock with screen lock' means finger print sensor." [P37]

Some users had problems completing the CAP challenge because they did not have a compatible setup, so they wanted better support for their devices (e.g., "Make it work with Windows Hello PIN" [P3]).

Although we did not specifically evaluate the accessibility aspects of CAP in this phase of study, we did wish to solicit feedback from anyone with these user requirements. Three suggestions were provided: two for larger UI elements and one for improved contrast.

Finally, respondents could provide us with any additional comments. Again, there was a call for extended support (on more devices and browsers), particularly to avoid failed attempts; recommendations for clearer communication to users; and requests for removing inefficiencies (such as pop-ups) where possible.

- "Chrome Android requires few more steps to actually choose which authentication system to use (NFC, security key or fingerprint). It doesn't automatically save my preferences so that I don't have to choose again" [P43]

Some other people wanted to simply express satisfaction with our approach:

- "I hope every website on the internet adopts this method" [P22]
- "I do wonder how well this will work to prevent farms of Captcha solving bots [...] if you can truly prevent that or stop it, this will be an amazing alternative." [P6]

Environment and Completion Time Of the 93 survey respondents, 82 provided details about their environment (browser, security hardware) along with the number of errors encountered during their entire CAP session and the time taken for their most recent verification attempt. Based on User Agent String, 39 were on mobile devices and 43 on desktop; the most commonly-reported browser was Chrome (41), followed by Safari (17), Firefox (10), and Edge (9). In terms of errors and timing, 50% of these respondents (41) had no errors at all in their session; 27 (33%) had one error in their session, and the remaining 14 (17%) had two or three errors. This is similar to the distribution found in the larger set of logs described previously, although there is a slightly higher success rate in the survey respondents.

Task completion timing was recorded, but note this measured duration from the initial JavaScript load event until the verification attempt ended, while the task time in the log dataset previously discussed was measured only from when the user clicked the button, which is a much shorter set of events. We analyzed the log data to give the same baseline for comparison: for a successful attempt, the survey participants took

15.1 s (vs 15.7 s) and 8.9 s for a failed attempt (vs. 7.0 s); again, this is similar to the larger dataset.

6 Discussion

6.1 Availability and Ease of Use

For successful validation cases, the completion time is quick (half that of hCaptcha), with few errors, and has high perceived efficiency. However, this is not the situation for all users: as noted, about half of them were unable to validate. The main difference of note comes down to environment: the biggest obstacle was having (or using) the correct combination of security hardware, OS, and browser. A summary of supported hardware and browser combinations is shown in Table 2. Survey respondents reported problems with validation when using MacOS with non-Safari browsers, and on Android mobile outside of Chrome, along with a few users having Windows compatibility issues.

Table 2: Overview of hardware support (based on testing in this study)

Hardware	Browser support	WebAuthn support	Secure attestation[1]
macOS (11 onwards)	Safari	✓	✓
	Major browsers	✓	✗
iOS 15 devices	Major browsers	✓	✓
Windows Hello	Microsoft Edge	✓	✓
	Other browsers	✓	✗
Android mobile	Chrome	✓	✓
	Other browsers	✗	✗
Hardware keys in FIDO MDS (e.g., YubiKeys)	Major browsers	✓	✓

[1] Secure attestation refers to attestation formats [23] that allow validation with a global issuing certificate.

For example, a person with a MacBook equipped with Touch ID would need to use Safari with CAP in order for the attestation to work properly; if they tried with Chrome, it would fail, as the Apple attestation sent with the Touch ID platform authenticator for WebAuthn is only compatible with Apple's browser (Safari). In some cases, the user might lack the necessary hardware, although this is becoming less of an issue given the deployment of built-in WebAuthn-compatible devices in mobile devices (e.g., Face ID), and the growing adoption of hardware security keys for multi-factor authentication [6].

As noted in the survey responses, the majority of respondents (75%) were likely to use CAP if they had the necessary hardware. These results suggest that CAP is a good solution in the right circumstances: given the appropriate environment, users prefer it to traditional CAPTCHAs. However, CAP is best positioned as an alternative challenge method for those equipped to take advantage of it, rather than it being presented as the sole option, given the number of users for whom it would not be possible or practical to use for a human challenge.

6.2 Communication Challenges

Explaining functionality Although the majority of our survey participants stated that they were likely to use CAP when possible, and many commented on how easy it was to use, it is important to consider that CAP involves a number of elements that are likely to be unfamiliar to many users. This is an entirely new human challenge method, which does not resemble the more familiar puzzle-based tasks. WebAuthn is itself a fairly new technology as well, and even those who may be comfortable with WebAuthn may be confused by its application in this unusual way. Those trying the CAP demo had the opportunity to review a substantial blog post with explanations of the technology before they tried it out; this would not be the most common scenario in a real-world deployment. Users need to know what this new feature does, and whether or not they are equipped to use it, as well as any additional considerations (such as privacy, discussed below). This is a lot to convey in a limited user interface. We have used the results of the study to refine our design and to augment customer support materials to assist users; these additions will be evaluated and refined iteratively as we continue to test CAP in deployment, as discussed in Section 7.

CAP as novel WebAuthn application CAP leverages the capabilities of WebAuthn and extends its functionality into the human challenge space; this is a benefit, and could provide additional incentives for people to obtain and use hardware security keys in order to mitigate their frustrations with CAPTCHAs. However, there are always challenges with novel technology, and in the CAP scenario, WebAuthn is being used for quite a distinct purpose from its usual application. Most people using WebAuthn are doing so for *authentication* purposes, and elements such as browser messages are designed with that in mind. As one example, consider the pop-up example shown when describing the public demo, in Figure 2. Note the text used when prompting the user: "Verify your identity with [example.com]". Often, this is what users are doing: verifying their identity as part of an authentication process, such as logging into their account. Because CAP does not include this component (as it never registers a user and does not handle credentials), this message does not properly describe what is about to happen. It is understandable that the WebAuthn browser designs prioritize the majority use case, but it would be helpful to accommodate other applications.

WebAuthn: inconsistent experiences Additionally, the design choices of CAP are only one part of the entire WebAuthn user experience; many of the messages displayed during user interaction are under the control of the browser, not CAP. If there is a confusing message displayed, or excessive popups, this also has an effect on the overall user experience. At best, one can anticipate and explain some of the confusing elements of the WebAuthn ecosystem. This is compounded by the number of different configurations that a person may be using: WebAuthn via Face ID on an iPhone using Safari is not identical to WebAuthn via Yubikey on a Windows laptop using Chrome. These are similar, but not identical, and the inconsistencies in these experiences can lead to a sub-optimal user experience: some may lead to a failed verification, while others might simply provide unclear information.

6.3 Privacy Considerations

The survey results indicated that not all users were confident in the privacy protections provided by CAP. While the overall sentiment was positive (with 60% of respondents expressing confidence), this shows an area where improvement is needed. Very few respondents (only four) who had low confidence in privacy provided any comments about this topic at all, so the source of their concerns is not clear. Two of these discussed privacy in the content of how communication might be improved, whereas the other two were more concerned about the actual data collection risks (i.e., what is the website collecting?). In one case, the participant was confused by the specific Firefox messaging that appears when attestation is requested: "Firefox displays a warning that the site 'is requesting extended information about your security key, which may affect your privacy'. I wouldn't necessarily trust this if I didn't know for sure that the request was coming from Cloudflare (which, in general, as a user, one doesn't)." [P72]. In other browsers, the message is different, despite it being for the same type of request: for example, as shown in Figure 2, Chrome v98 says that the site "wants to see the make and model of your security key". This example shows the importance of communication, and also the stark differences that users can experience between different environments.

7 Enhancements and Future Work

The findings from our user studies have highlighted areas where CAP could be improved, along with some promising new directions. We have also identified some research questions that we will continue to explore.

7.1 Improving Communication

When we conducted our usability evaluations, we provided explanatory material (such as blog posts) that assisted users in learning about this new human challenge approach that is enabled by secure hardware; this also explained the underlying technology and its privacy and security capabilities. This was workable for experiments, but is not realistic outside of this situation. In the more usual scenario, a user would be browsing the web and then encounter a human challenge, such as an interstitial page containing a CAP prompt. A first-time user would have no previous experience with this type of attestation challenge, and perhaps would have no previous experience with WebAuthn at all. They would need to know how they might pass this challenge, including whether or not they had the right hardware and environment to do so successfully. They might also wonder about the security and privacy risks associated with using secure hardware to pass this challenge. Note also that those with WebAuthn experience in its more common *authentication* situation might have specific expectations about CAP that are not true: for example, they may expect they need the same hardware device to pass a challenge on repeat visits to a particular website.

This situation presents many significant challenges for user communication, and we are continuing to work on solutions. We began by revising the visual elements of the CAP prompt panel, so that it gives a suggestion that this is a task you perform with secure hardware; the first version of our new design displays a graphic with a fingerprint (to suggest a biometric reader) plus a USB key. We are also developing new customer support materials, which might involve videos to demonstrate the technology and how to use it; this would be readily accessible from the challenge page, in context with the CAP prompt. Providing explanations for WebAuthn through richer interactions, such as video, is consistent with recommendations provided in recent research on WebAuthn adoption; this was shown to be beneficial for mitigating misconceptions (such as where biometrics are stored) [25]. We expect to iterate on our designs as we have begun to run small-scale tests in a production environment and can evaluate the results.

7.2 Privacy: Zero-Knowledge Proofs

We have continued to explore how we might improve the privacy story for CAP and WebAuthn. In an extension of this work, we investigated how one might disclose the minimum possible amount of information: not the make and model of the security key, but simply the proof that the key being used is trustworthy. We developed an in-browser zero-knowledge proof to provide this functionality [17]. In brief: instead of sending the signature, the client sends a proof that the signature was generated by a key on a server-provided list. Because only the proof is sent, the server learns only that the attestation exists, and not which hardware security key generated it. An efficient proving and verification system was developed for this scenario, which is currently being evaluated. Results to date demonstrate that a zero-knowledge proof can be generated in approximately 10 seconds, which is extremely

promising as an efficient, privacy-preserving solution.

Ideally, this solution could be integrated into the WebAuthn standard, as an attestation type, so that it could signal to browsers that sending this particular attestation type would not disclose the make and model of hardware key (given the underlying zero-knowledge proof). In that case, there would be no need for the consent pop-up that users must click through, as there is no disclosure in this case. Not only would this be a more robust privacy solution, but it would also make WebAuthn interactions more efficient and less confusing for users, for any instance in which attestations were used (which is not restricted to CAP).

7.3 Exploring Privacy Concerns

While one path of our ongoing privacy improvements involves zero-knowledge proofs, we would also like to explore what some of the underlying privacy concerns are that could impede the adoption of CAP. Our survey touched on this question, as we anticipated its importance, but as noted above, this was designed to be a short questionnaire that did not delve deeply into any one specific area—including privacy. However, given the complexity and persistence of privacy considerations of WebAuthn in general, we feel it would be valuable to deepen our understanding of this problem. There are many potential sources of unease, some of which may be unrelated to the human challenge itself. For example, a person might choose not to use CAP because they do not want to use a biometric reader, and their underlying discomfort may be due to the biometric component in itself, which would be the case in *any* online context (not just for CAP). A better understanding of these factors would help us determine how to improve designs for this specific application, as well as how to contribute to WebAuthn adoption more broadly. This would be informed by, and build on, ongoing user research in this domain (e.g., [25, 29, 27]).

7.4 Security Considerations

In designing new methods for attesting to personhood, we must be mindful of security issues that arise when *malicious* clients attempt to provide false proofs of humanity. In the following, we attempt to build an overview of the threat model and potential methods for calculating adversarial costs of providing false proofs. Valuable future work would establish a thorough security analysis of using such attestations widely before establishing a large-scale deployment of these technologies.

Threat model We can split the attack surface into the following two types of attacks:

- *Human-assisted*: These type of attacks involve an adversary proxying attestation requests to a real human being, who provides the proof based on their own inherent characteristics and returns the proof to the adversary to be returned to the requester.
- *Automated*: These attacks involve constructing mechanisms (either physically or in software) that allow generating valid attestation proofs from hardware authenticators, without a real person interacting with them.

All challenges that attempt to provide attestation of humanity—including all CAPTCHAs and related technologies—are vulnerable to human-assisted attacks. This assumes, however, that a challenge that an adversary receives can always be forwarded to a different real person that can solve the challenge instead. Currently, it is an open problem whether forwarding of hardware attestations is possible, and to what extent that compares to existing challenge systems.

In addition, software-based challenges are vulnerable to automated attacks that involve no human participation. As mentioned previously, CAP authenticators that rely solely on touch (rather than biometric identification, such as Yubico Yubikeys) may be vulnerable to automated attacks that involve constructing physical devices that generate valid interactions with the device. It is more difficult to circumvent biometric authenticators; such biometrics have not yet been mimicked in a similar manner (again, assuming it is possible to forward hardware attestations).

Adversarial costs A common way of establishing the security of a human-based challenge system is identifying the cost of buying a single valid attestation. These attestations can be provided either by real humans (who are paid for solving each challenge), or by an adversary that controls a resource that is able to provide automated proofs. Generally speaking, vulnerability of a challenge mechanism to automated attacks is quite damaging, since it is likely that such proofs can be provided much more cheaply than those that require human assistance.

In the case of CAPTCHAs, various services are known to price a single solution of a standard Google reCAPTCHA at $0.003[2]. Therefore, even human-assisted challenges are very cheap to acquire solutions for. Hardware-based authenticators such as Yubico Yubikeys require an initial up-front cost of between $45 and $85.[3] Assessing the cost of launching an automated attack on top of these authenticators would be a valuable task for future work, but is likely to involve another one-time cost of setting up the tools that are required for automation, plus the much lower cost of continued usage. CAP authenticators that rely on biometrics are likely to involve much higher costs. Firstly, devices such as smartphones and

[2] According to https://www.f5.com/labs/articles/cisotociso/i-was-a-human-captcha-solver.

[3] See https://www.yubico.com/us/store/ (accessed 23 May 2022).

laptops that provide valid signals incur very significant one-time costs. Moreover, they will also require paying for human subjects to provide valid proofs of personhood, which will further incur per-usage costs.

A rough analysis using the above figures could suggest that it might be economically advantageous to launch automated physical attacks on touch-based authenticators via commodity hardware. However, servers can tip the economic balance against attackers, by leveraging the asymmetry of information about the types of authenticators being used. While leveraging this asymmetry remains an open research topic, our system provides visibility into the global breakdown of device types, which attackers do not have. As mentioned previously, authenticators are typically associated with coarse-level batches of a specific model by their attestation certificates (Section 3.2). Thus, a server has the ability to collect and maintain a view of different device types. An attacker may invest in a particular model of security key that could be removed from the list of allowed devices (e.g., if it was uncommon and mainly used for attacks). This adds an additional risk for the attacker, who may find their investment wiped out with one configuration change. The attack cost is higher to maintain for a diverse profile of security keys that matches up with the global distribution. The server could remove support for specific device keys if a farm of them was discovered; it is worth noting that this would affect legitimate users that share devices within the same batch as the attacker, but the diversity of keys used in practice means CAP would still be effective for most of the other users. Note also that unlike human-assisted farms, where the cost is per-CAPTCHA, security key farms have an upfront cost that is amortized over time. The ability for the server to selectively support the feature for specific devices or regions introduces a significant downside risk for any capital investment by attackers. In summary, valuable future work would establish whether using such mechanisms as a viable mitigation is possible, without introducing significant overheads to legitimate users.

8 Related Work

CAPTCHA-related research that has motivated and guided our explorations is described throughout this work and includes studies of usability [15, 18, 24] and security [34, 35]. Recent and related streams of study on security key usability identify many strengths along with some weaknesses [16, 8]. Their results are highly encouraging and report that users are readily able to physically interact with YubiKeys. Minor and occasional problems included key touches that fail to be recognized [16], or the key being inserted incorrectly [8]. Users are also concerned about being able to locate or losing such small devices [30]. These occurrences will be familiar to any user of touch and biometric devices (e.g., mobile device fingerprint sensors). We anticipate their reductions with practiced use, improvements in hardware sensors, and further hardware integration.

Many security key usability challenges emerge as part of a two-factor authentication (2FA) [11, 16]. Our hardware challenge task has lower barriers to entry since (i) there are no passwords or user accounts involved, and (ii) a failed challenge can fall back to a CAPTCHA. However, the same works identified inconsistencies and inadequacies in messaging and best practice for WebAuthn among Internet browsers [16]. This observation is in keeping with our own and deserves further attention.

There have also been some recent studies about FIDO and WebAuthn usability more broadly, which are helping highlight specific challenges and potential solutions. A study of mobile phones as roaming authenticators [29] suggested that users wanted to take advantage of the user presence features (such as facial recognition) available on their smartphones for authentication; the convenience of these features could also be leveraged for the attestation-only variant of WebAuthn (as in CAP).

An exploration of user misconceptions about WebAuthn biometrics [25] provides useful insights into some of the persistent points of confusion and gives recommendations for mitigating these (e.g., by providing more explicit guidance about where biometric data is stored, and providing users with more than just simple notification messages when explaining the technology). We have identified similar issues and are experimenting with ways of improving the user experience, particularly in terms of communication.

9 Concluding Remarks

The balancing act between security and usability places undue hardship on users to complete frustrating, impenetrable CAPTCHAs that have a number of serious shortcomings. Based on our user study we believe that a cryptographic attestation to a physical interaction provides a better solution for users without degrading bot detection.

We hope that others will be able to apply this solution in their own environments, leveraging the open WebAuthn standard to benefit from cryptographic attestations for human challenges. Our evaluation provides us with confidence that this is a fruitful approach for those users poised to take advantage of it; the necessary hardware is already widely available and is being rolled out even further. We have identified a number of barriers to adoption, however, primarily in the areas of privacy, clarity of communication, and consistency of user experience with WebAuthn. We will continue to pursue research into these areas, in hopes that cryptographic attestations will be more widely adopted and provide users with better ways of completing human challenges.

Acknowledgments

We gratefully acknowledge the assistance of our study participants and our anonymous shepherd and reviewers. We would also like to thank our Cloudflare colleagues for their valuable support.

References

[1] Usage statistics and market share of Cloudflare. https://w3techs.com/technologies/details/cn-cloudflare.

[2] William Aiken and Hyoungshick Kim. Poster: Deepcrack: Using deep learning to automatically crack audio CAPTCHAs. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 797–799, 2018.

[3] FIDO Alliance. FIDO Alliance Metadata Service v3.0. https://fidoalliance.org/metadata/. Accessed Feb 2022.

[4] Dirk Balfanz, Alexei Czeskis, Emil Lundberg, J.C. Jones, Jeff Hodges, Michael Jones, Rolf Lindemann, Akshay Kumar, and Huakai Liao. FIDO UAF Protocol Specification v1.0. FIDO Alliance Standard, FIDO, December 2014. https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-protocol-v1.0-ps-20141208.html.

[5] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[6] Garrett Bekker and Matthew Utter. Work-from-home policies driving MFA adoption, but still work to be done, Apr 2021. "https://pages.yubico.com/work-from-home-policies-driving-mfa-adoption. Accessed Feb 2022.

[7] John Brooke. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, page 189, 1996.

[8] Stéphane Ciolino, Simon Parkin, and Paul Dunphy. Of two minds about two-factor: Understanding everyday FIDO U2F usability through device comparison and experience sampling. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS)*, 2019.

[9] J Clement. Mobile internet usage worldwide - statistics & facts. Statista, Jul 12 2021. https://www.statista.com/topics/779/mobile-internet/. Accessed Feb 2022.

[10] George S. Coker, Joshua D. Guttman, Peter A. Loscocco, Amy Herzog, Jonathan Millen, Brian O'Hanlon, John Ramsdell, Ariel Segall, Justin Sheehy, and Brian Sniffen. Principles of remote attestation. *International Journal for Information Security*, 10(2):63–81, 2011.

[11] Sanchari Das, Andrew Dingman, and L Jean Camp. Why Johnny doesn't use two factor: a two-phase usability study of the FIDO U2F security key. In *International Conference on Financial Cryptography and Data Security*, pages 160–179. Springer, 2018.

[12] MDN Web docs. Web authentication API. https://developer.mozilla.org/en-US/docs/Web/API/Web_Authentication_API#Browser_compatibility. Accessed Feb 2022.

[13] Josh Dzeiza. Why CAPTCHAs Have Gotten So Difficult. *The Verge*, Feb 2019. https://www.theverge.com/2019/2/1/18205610/google-captcha-ai-robot-human-difficult-artificial-intelligence.

[14] Wesley Evans and Tara Whalen. More devices, fewer CAPTCHAs, happier users, August 2022. https://blog.cloudflare.com/cap-expands-support.

[15] Valerie Fanelle, Sepideh Karimi, Aditi Shah, Bharath Subramanian, and Sauvik Das. Blind and human: Exploring more usable audio CAPTCHA designs. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS)*, pages 111–125, 2020.

[16] Florian M Farke, Lennart Lorenz, Theodor Schnitzler, Philipp Markert, and Markus Dürmuth. "You still use the password after all" – Exploring FIDO2 Security Keys in a Small Company. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 19–35, 2020.

[17] Armando Faz-Hernández, Watson Ladd, and Deepak Maram. ZKAttest: Ring and group signatures for existing ECDSA keys. In *International Conference on Selected Areas in Cryptography*, pages 68–83. Springer, 2022.

[18] Ruti Gafni and Idan Nagar. CAPTCHA – Security affecting user experience. *Issues in Informing Science and Information Technology*, 13:063–077, 2016.

[19] Google. Google reCAPTCHA: Register a site. https://www.google.com/recaptcha/admin/create. Accessed Feb 2022.

[20] Lucy Handley. Nearly three quarters of the world will use just their smartphones to access the internet by 2025, Jan 2019. "https://www.cnbc.com/2019/01/24/smartphones-72percent-of-people-will-use-only-mobile-for-internet-by-2025.html. Accessed Feb 2022.

[21] hCaptcha. hCaptcha Developer Guide. Available at https://docs.hcaptcha.com/.

[22] Scott Hollier, Janina Sajka, Matthew May, Michael Cooper, and Jason White. Inaccessibility of CAPTCHA. W3C note, W3C, December 2019. https://www.w3.org/TR/2019/NOTE-turingtest-20191209/.

[23] J.C. Jones, Akshay Kumar, Alexei Czeskis, Vijay Bharadwaj, Dirk Balfanz, Hubert Le Van Gong, Huakai Liao, Michael Jones, Jeff Hodges, Rolf Lindemann, and Arnar Birgisson. Web Authentication: An API for accessing Public Key Credentials Level 2. W3C working draft, W3C, July 2020. https://www.w3.org/TR/webauthn-2/.

[24] Kat Krol, Simon Parkin, and M Angela Sasse. Better the devil you know: A user study of two CAPTCHAs and a possible replacement technology. In *NDSS Workshop on Usable Security (USEC)*, volume 10, 2016.

[25] Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. "It's stored, hopefully, on an encrypted server": Mitigating users' misconceptions about FIDO2 biometric WebAuthn. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 91–108, 2021.

[26] Wei Liu. Introducing reCAPTCHA v3: the new way to stop bots. Google Webmaster Central Blog, October 2018. https://webmasters.googleblog.com/2018/10/introducing-recaptcha-v3-new-way-to.html. Accessed Feb 2022.

[27] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication. In *IEEE Symposium on Security and Privacy*, pages 268–285, 2020.

[28] Thibault Meunier. Humanity wastes about 500 years per day on CAPTCHAs. It's time to end this madness, May 2022. https://blog.cloudflare.com/introducing-cryptographic-attestation-of-personhood.

[29] Kentrell Owens, Olabode Anise, Amanda Krauss, and Blase Ur. User perceptions of the usability and security of smartphones as FIDO2 roaming authenticators. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 57–76, 2021.

[30] Joshua Reynolds, Nikita Samarin, Joseph Barnes, Taylor Judd, Joshua Mason, Michael Bailey, and Serge Egelman. Empirical measurement of systemic 2FA usability. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 127–143, 2020.

[31] Michael Richardson, Carl Wallace, and Wei Pan. Use cases for Remote Attestation common encodings. https://datatracker.ietf.org/doc/html/draft-richardson-rats-usecases-08, November 2020. Work in Progress.

[32] Max Roser, Hannah Ritchie, and Esteban Ortiz-Ospina. Internet. *Our World in Data*, 2015. https://ourworldindata.org/internet.

[33] Catherine Schwab. Google's new reCAPTCHA has a dark side. Fast Company, June 2019. https://www.fastcompany.com/90369697/googles-new-recaptcha-has-a-dark-side.

[34] Chenghui Shi, Shouling Ji, Qianjun Liu, Changchang Liu, Yuefeng Chen, Yuan He, Z Liu, R Beyah, and T Wang. Text Captcha is dead? A large scale deployment and empirical study. In *The 27th ACM Conference on Computer and Communications Security*, 2020.

[35] Suphannee Sivakorn, Jason Polakis, and Angelos D. Keromytis. I'm not a human: Breaking the Google reCAPTCHA. In *Black Hat ASIA*, 2016.

[36] Saumya Solanki, Gautam Krishnan, Varshini Sampath, and Jason Polakis. In (cyber)space bots can hear you speak: Breaking audio CAPTCHAs using OTS speech recognition. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 69–80, New York, NY, USA, 2017. Association for Computing Machinery.

[37] Jennifer Tam, Sean Hyde, Jiri Simsa, and Luis Von Ahn. Breaking audio CAPTCHAs. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, page 1625–1632, Red Hook, NY, USA, 2008. Curran Associates Inc.

[38] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. CAPTCHA: Using Hard AI Problems for Security. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, pages 294–311, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[39] Guixin Ye, Zhanyong Tang, Dingyi Fang, Zhanxing Zhu, Yansong Feng, Pengfei Xu, Xiaojiang Chen, and Zheng Wang. Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 332–348, New York, NY, USA, 2018. Association for Computing Machinery.