"If I could do this, I feel anyone could:" The Design and Evaluation of a Secondary Authentication Factor Manager

Garrett Smith, Tarun Yadav, and Jonathan Dutson, *Brigham Young University;*
Scott Ruoti, *University of Tennessee Knoxville;*
Kent Seamons, *Brigham Young University*

https://www.usenix.org/conference/usenixsecurity23/presentation/smith

This paper is included in the Proceedings of the 32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

Open access to the Proceedings of the 32nd USENIX Security Symposium is sponsored by USENIX.

"If I could do this, I feel anyone could:" The Design and Evaluation of a Secondary Authentication Factor Manager

Garrett Smith*
Brigham Young University

Tarun Yadav*
Brigham Young University

Jonathan Dutson
Brigham Young University

Scott Ruoti
University of Tennessee Knoxville

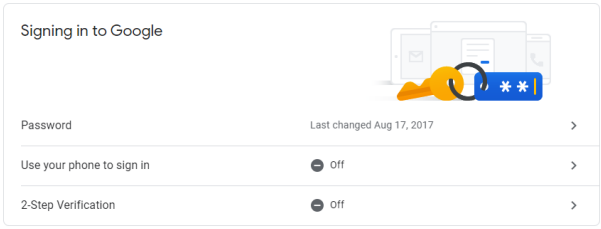
Kent Seamons
Brigham Young University

Abstract

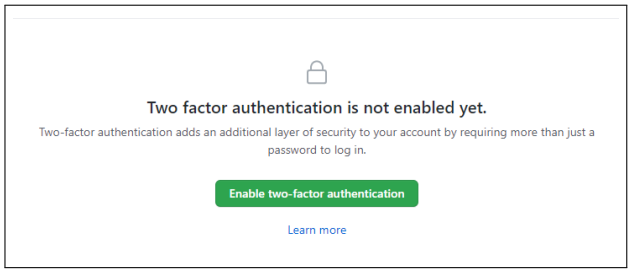
Two-factor authentication (2FA) defends against account compromise by protecting an account with both a password—the primary authentication factor—and a device or resource that is hard to steal—the secondary authentication factor (SAF). However, prior research shows that users need help registering their SAFs with websites and successfully enabling 2FA. To address these issues, we propose the concept of a SAF manager that helps users manage SAFs through their entire life cycle: setup, authentication, removal, replacement, and auditing. We design and implement two proof-of-concept prototypes. In a between-subjects user study (N=60), we demonstrate that our design improves users' ability to correctly and quickly setup and remove a SAF on their accounts. Qualitative results show that users responded very positively to the SAF manager and were enthusiastic about its ability to help them rapidly replace a SAF. Furthermore, our SAF manager prevented fatal errors that users experienced when not using the manager.

1 Introduction

Password-based authentication remains the dominant form of authentication on the Web. However, attackers steal passwords using various means, such as phishing and password database leaks. Once stolen, attackers can use a password to impersonate that user from anywhere in the world. To address this threat, websites can require that users demonstrate ownership of some device or resource that is difficult to steal remotely—for example, a hardware security token or a phone number. This approach is known as two-factor authentication (2FA), with passwords serving as the primary authentication factor and the difficult-to-steal device or resource acting as the *secondary authentication factor (SAF)*. While 2FA does not entirely prevent remote account compromise, it does reduce the likelihood of such an attack and mitigate the impact of a successful attack [20, 21, 49].



(a) Google



(b) GitHub

Figure 1: Examples of differences in 2FA terminology and interfaces

Studies of 2FA have shown that users often need help configuring their online accounts to use a SAF [3, 13, 38]. One issue is that there are many different types of SAFs, and even for a given SAF, there may be multiple ways to use it. For example, for a phone, users may need to enter a code displayed on the phone into the website, a code displayed on the website into the phone, or tap a button displayed on the phone in a push notification. This heterogeneity can lead to significant confusion for users [36].

Similarly, websites supporting 2FA use different terminology, interfaces, and workflows (see Figure 1). This variation limits the ability of users to transfer experience from one site to another directly and may even cause

*These authors contributed equally to this work.

mistakes if sites function differently from each other [38]. This issue is further exasperated when users need to update many accounts at the same time, such as when first adopting 2FA, replacing an existing SAF (e.g., getting a new phone), removing a lost SAF, or adding a SAF (e.g., giving a spouse or child access). Such large-scale operations can both be time-consuming and confusing, or as Matt Blaze recently tweeted: *"Upgraded to a new phone, which is like taking a 12 hour refresher course in configuring 2FA"* [8].

To address these issues, we propose the concept of a *SAF manager*. Much like a password manager improves the usability and security of passwords, a SAF manager can improve the usability and security of SAFs. It does so by helping users manage their SAFs through the entire SAF life cycle: setup, authentication, removal, replacement, and auditing. Critically, throughout this process, it uses unified terminology, interfaces, and workflows, regardless of the websites or SAFs in use.

As a first step towards testing our vision of a SAF manager, we analyzed various websites' 2FA workflows. Using this information, we designed and implemented a proof-of-concept SAF manager that unifies and partially automates the setup and removal of SAFs for popular websites. The source code for the proof-of-concept prototype is provided open source at https://bitbucket.org/isrlauth/saf_manager/src/master/. **Even if SAF managers are never adopted, our design improvements could be integrated into existing websites to significantly improve the usability and convenience of 2FA across the Web.** To evaluate our design, we conducted a between-subject user study (N=60) designed to answer the following research questions:

RQ1 Does our SAF manager increase the success rate of setting up and removing a SAF?
RQ2 Does our SAF manager reduce the completion time when setting up and removing a SAF?
RQ3 Does our SAF manager increase the perceived usability of the setup and removal process?
RQ4 While using SAF manager, does prior 2FA experience (1) increase the success rate and perceived usability, and (2) reduce the completion time of setting up and removing a SAF?

Our results show that

- Users with no prior experience using SAF face challenges (such as the discoverability of settings and inconsistency) while setting up and removing SAF on multiple websites:
  - only 75% of such users complete the SAF setup on three websites, with a mean time of 7 min 52 secs.
  - 93% of such users complete the SAF removal on three websites, with a mean time of 2 min 42 secs.
- A SAF manager improves users' ability to correctly and quickly set up and remove SAFs on multiple accounts (RQ1, RQ2, and RQ3).
  - 100% of users complete the SAF setup and removal on three websites, with a mean time of 5 min 15 secs and 1 min 2 sec, respectively.
  - Qualitative results show that users found our manager easy to use and were enthusiastic about its ability to help them rapidly replace their SAFs if they lost one.
- Prior 2FA experience does not have a significant impact on users' abilities or experiences to use a SAF manager. (RQ4)

2 Related Work

2.1 Two-Factor Authentication (2FA) Usability

Studies of 2FA usage in laboratory studies [13, 14, 18, 50, 51] and field studies [11, 22, 23, 36, 38] have shown that users are generally happy with the day-to-day usage of 2FA [38]. However, evidence strongly suggests that setup has usability challenges that have been difficult to address.

Usability of 2FA setup A repeated theme from multiple user studies is that users often are confused about whether the setup process is complete. Das et al. [13] conducted a think-aloud experiment for setting up a security key on Gmail. All users stopped prematurely and failed to complete the setup due to a confusing Yubico demonstration tool. Acemyan et al. [3] conducted a lab study of four Google 2FA methods with a setup success rate of only 68%. Pandey et al. [33] conducted a lab study of the setup process for two of Google's 2FA methods, with errors due to some participants not following written instructions. After setup, 64% of the participants were unsure whether they had completed the setup task.

Ciolino et al. [11] conducted a lab study and a diary study comparing three different hardware security keys and a one-time password. A major source of failures was the setup process. Reynolds et al. [38] also conducted a lab usability study of the Yubikey setup process for Google, Facebook, and Windows. They identified significant problems with the Facebook and Windows setup process that resulted in many users (1) not completing the setup process on Facebook because they believed they had setup the security key when they had not, and (2) locking themselves out of their Windows account. They recommended that a standardized 2FA setup process could improve usability.

Two lab studies used mock websites and observed positive results. Reese et al. [36] conducted a study of five SAFs, while Lyastani et al. [26] conducted a study comparing passwords to passwordless logins using a security key. To address the usability challenges of setup, the latter study included an instructional video to guide users through the setup process.

User perceptions Some early studies focused on 2FA usability in UK banking systems. Gunson et al. [18] studied a hardware fob and showed that although most customers

believed 2FA increased their security, they also perceived it significantly lowered usability and convenience. Fagan et al. [17] show that 71% of users reported inconvenience and cost as the reason not to use 2FA. Krol et al. [22] also found that participants were generally dissatisfied with using hardware tokens. They recommend reducing the number of steps required for authentication to as few as possible to provide a more usable authentication experience.

Three surveys have been conducted at U.S. universities deploying 2FA to understand user attitudes about 2FA [2, 12, 15]. They found that new 2FA users anticipate it being more inconvenient and difficult to use than it is. Thus, they recommend that organizations require 2FA because there will be fewer usability concerns after using it. A takeaway from these studies is that users prefer to use 2FA to protect only the most sensitive accounts; it is overkill for less-sensitive accounts. Also, a *remember-me* option is desirable to reduce 2FA frequency and increase user acceptance.

Reynolds et al. [37] did a quantitative study inspired partly by these university surveys. They analyzed two universities' authentication log entries and support tickets related to 2FA. Nearly half of the 500 support tickets were related to setup, removal, and update of a second factor.

Relation to our work Previous work shows significant usability problems with setting up 2FA, motivating our goal to standardize the setup process. Unfortunately, we observed some of the same failures reported in earlier studies, indicating that flawed designs persist, such as participants assuming they finished when they did not.

Our work also contains recommendations for improving the 2FA experience, such as reducing the number of steps needed to authenticate with a SAF [22] and standardizing the setup process [36, 38]. In this paper, we incorporate these recommendations into the concept of a SAF manager. Our prototype SAF manager also demonstrates the positive effects of simplifying and standardizing the setup and removal ceremonies for SAFs.

We took a different approach than the Lyastani study by exploring whether we could have novice users succeed at setup without prior training or an instructional video to guide them. Also, to our knowledge, we are the first study to measure the usability of SAF removal.

2.2 Password Managers

The motivating hypothesis behind our SAF manager is that by making it easier to register, track, update, and remove SAFs, users will be more likely to adopt SAFs, thereby increasing users' online security. This hypothesis was partly inspired by password managers, which help users generate, store, and fill passwords to improve password strength and reduce password reuse.

Adoption Research has shown that the adoption of password managers is often driven by a desire for increased usability [16, 34, 45]. In contrast, non-adoption is often driven by concerns with security [4, 5, 10, 27]. Additionally, research into password manager adoption and usage among older adults has shown that different populations also have different motivations and needs [35].

Usability Simmons et al. [44] systematized password manager use cases, finding that many use cases were poorly supported by today's managers and that even when supported, they were often targeted at experts rather than the lay users the tools claimed to support. Relatedly, research has shown that poor usability stymies the adoption of password manager security-critical functionality [25, 32], such as generating passwords.

Integration Usability issues with password managers often arise from poor integration between password managers and the websites and apps that use passwords. For example, the highly heterogeneous nature of website implementations causes managers to frequently fail to properly automate password fill [19]. A similar issue exists between password managers on mobile devices and apps on that devices [42].

Security While password managers have the potential to provide strong security benefits, they also act as a potential single point of failure. For example, repeated studies [24, 31, 43, 46] have shown that poor manager implementations leave users vulnerable to credential harvesting attacks wherein all of a user's passwords are clandestinely stolen. This problem can be made even worse when the operating system enforces incorrect behavior, which is the case for mobile autofill frameworks [30].

Relation to our work The above research demonstrates that improving the usability of authentication can encourage users to adopt better authentication technologies (e.g., password managers). Our results demonstrate that a SAF manager improves the usability of authentication, and so, like password managers, they have the potential to be widely adopted, which could have a knock-on effect of increasing the adoption of 2FA. Still, as is the case with password managers, future research will be needed to refine and enhance the usability, utility, and security of SAF managers. In this regard, our work does not answer the question of whether SAF managers are the right step forward but instead opens the door to this compelling area of inquiry, similar to the research currently being done on password managers.

3 SAF Manager Concept

We envision a SAF Manager as a tool to enhance the usability and security of SAFs, comparable to how password managers assist users with their passwords. A SAF Manager helps users manage their SAFs throughout their life cycle, including setup, authentication, removal, replacement, and auditing. As much as possible, a SAF manager would seek to automate tasks, reducing user burden and making it easier and quicker to make mass changes across all of their accounts—such as replacing an existing SAF (e.g., a hardware security key) across all their accounts at once when they get a new one.

Critically, a SAF manager would unify the terminology, interfaces, and workflows users encounter, regardless of which SAFs they use and which websites they want to enable 2FA. This vision stands in stark contrast to the current state of affairs, where every website implements the SAF life cycle differently. These differences range from small (functionality in different parts of the website) to large (entirely different terminology), but regardless of their size lead to a degraded user experience, user confusion, and a limited ability to transfer experiences managing a SAF on one website to others [38].

The benefits of a SAF manager are not limited to usability but also extend to security. By addressing UI/UX inconsistencies, a SAF manager can reduce the burden for initial 2FA usage, potentially increasing 2FA adoption by the masses. Similarly, by ensuring that the SAF setup completes correctly, a SAF manager can avoid situations where users think they have enabled 2FA but have not [38]. Also, by auditing a user's SAF management, the manager can help users adopt 2FA on as many websites as possible and help them address any security concerns that may arise.

3.1 Life Cycle

A SAF Manager helps users manage their SAFs through the entire life cycle, including setup, authentication, removal, replacement, and auditing.

Setup The initial setup includes registering a SAF with a website and enabling 2FA for that account. While seemingly simple, prior research has shown many usability impediments with initial setup [13, 38]. A SAF manager could help address the issues by simplifying and partially automating the setup process and ensuring it completes correctly. Automating this process as much as possible enables the mass adoption of 2FA across all user accounts.

Authentication Once a SAF is setup with a website, it will then be used to authenticate to that website. A SAF manager could ensure that the authentication ceremony is consistent across the SAFs and websites the user employs.

Removal If a user loses access to their SAF or no longer wants to use 2FA for an account, they need to remove it from the account. The SAF manager can help automate removal.

Replacement Replacing a SAF may be desirable in response to changing security needs, such as replacing SMS with a hardware security key. It also happens when users need to replace existing SAFs with a new model, such as when getting a new phone. A SAF manager can automate the process of replacing SAF devices by first adding the new device and then removing the old device. It can also handle any per-website inconsistencies that may make this process problematic, such as websites allowing only the registration of a single SAF at a time.

Auditing Finally, similar to password health checks in password managers [44], a SAF manager can help users audit their usage of SAFs and 2FA, helping them improve their security posture over time. For example, a SAF manager could monitor the websites a user authenticates to and notify users of accounts for which they could enable 2FA with their existing SAFs. Similarly, a SAF manager could help users identify situations where they are using a weaker SAF when the website also has support for a stronger SAF that the user also uses. Also, it could track how frequently users need to provide their SAF to login to various websites, allowing users to adjust their remember me settings as they feel appropriate. These and more auditing features can significantly increase the security of 2FA users using a SAF manager.

4 Design/Implementation of a SAF Manager

As a first step towards testing our vision of SAF managers, we designed a SAF manager that supports setting up and removing SAFs. Our design provides a standardized set of terminology, interfaces, and workflows to improve the usability of SAFs and promote the adoption of 2FA on websites. Three main goals informed our design:

1. Improve the success rate for setting up and removing SAFs from an account.
2. Reduce the time needed to setup or remove SAFs from an account.
3. Increase perceived usability of the setup and removal processes and 2FA generally.

4.1 Design

To inform the design of our SAF manager, we analyzed a range of 2FA implementations at popular websites, including Google, Facebook, Dropbox, Twitter, Pinterest, LinkedIn, Yahoo, Reddit, and GitHub. We derived common terminology, interfaces, and workflows for setting up and removing SAFs.

We used a minimalist design style to keep our UI simple and professional. The design supports enabling and removing SAFs on multiple sites at a time. Ideally, enabling 2FA on one site would have a nearly identical experience to enabling 2FA on another site. However, this is impossible because our manager cannot override a site's 2FA policies. For example, Google requires enabling SMS before enabling stronger 2FA methods such as TOTP or security keys. Our design reduces and automates the steps necessary to setup or remove a SAF wherever possible. Prior experience shows these efforts can significantly improve usability [22]. **Even if SAF managers are never adopted, these simplified and consistent designs could be integrated into existing websites to significantly improve the usability and convenience of 2FA across the Web.**

4.1.1 Setting up a SAF

Across the websites we evaluated, five steps are needed to setup a SAF. The first four steps occur on every website, while the fifth only happens on some.

1. The user selects a website and logs in We did not find any websites that let users create their account with 2FA already setup. Instead, to setup an account with 2FA, a user would first create an account with a password. They would then log into that account and initiate the SAF setup workflow (Steps 2–4). If the user was already logged into the account, then this step can be skipped as long as the website does not require re-authentication before modifying login settings.

Instead of having users navigate to individual websites to initiate, in our design, we provide users with a list of websites that our manager supports. They can then select one or more of these websites to setup their SAF (see Figure 2). While our current prototypes support only a handful of sites, future versions could support an arbitrary number of sites by: (a) allowing users to search through the list of supported sites, (b) automatically detecting the set of sites a user uses and suggesting setting up 2FA on those sites, or (c) allowing users to trigger the SAF setup workflow when viewing the target website in the browser.

After selecting a website(s), the SAF manager will help users log into the websites. It can fully automate this process by storing user credentials, like a password manager. If stored credentials are unavailable, then the SAF manager will provide consistent terminology, interfaces, and workflows for entering the user's credentials. We created two different implementations of our SAF manager (see §4.2), one that was integrated with a password manager and took the former approach and another that was a browser extension that took the latter approach.

2. The user selects a SAF and authentication method to register with their account Next, users must select which SAF and authentication method to register with their account. For example, they might use their phone as their SAF to receive a phone call or an SMS message. In our design, the SAF manager lists all supported SAFs and methods, provides a brief description, and allows users to select which one they will use (see Figure 3).

In the future, the design could allow users to identify their preferred SAF and authentication method. If this SAF and method are available for the website, step 2 could be fully automated, further simplifying the setup process for the user.

3. The SAF's identifier is shared between the SAF and the website A unique SAF identifier ensures that only valid SAFs are accepted during authentication. The identifier must be shared between the SAF and the website to register the SAF with the website. Most commonly, the SAF creates its identifier—for example, a public key generated by a hardware security token—or intrinsically linked to the SAF—such as a phone number. In some cases, the website generates the identifier and sends it to the SAF—for example, a website generates and sends a shared secret to a time-based one-time password (TOTP) SAF, which uses this secret to generate one-time passwords.

Where possible, our manager fully automates this process. If users are required to enter information or take action, such as scanning a QR code, our design provides unified terminology and interfaces for this process (see Figure 4).

4. The website conducts a challenge-response exchange to prove the user's possession of the SAF Before finishing the registration of the SAF, websites conduct a challenge-response exchange to verify that the user has access to the SAF. This exchange helps prevent errors leading to account lockout. The details of this challenge-response differ for each SAF and method, though it generally matches the normal authentication flow.

5. Activate 2FA for the user's account For many websites, 2FA is enabled for the user's account after successfully completing the challenge-response exchange. However, even after registering a SAF, 2FA is not enabled on some websites until it is explicitly turned on. This extra step is a source of confusion for users and may leave them thinking their account is protected using 2FA when it is not [38], a significant security issue. Our design addresses this issue by automatically enabling 2FA for the website once the SAF is registered.

4.1.2 Removing a SAF

Across the websites we evaluated, SAF removal requires three steps. Steps 1 and 2 are identical to their counterparts when setting up a SAF: (1) the user selects a website and logs in, and (2) the user navigates to the security settings and selects a

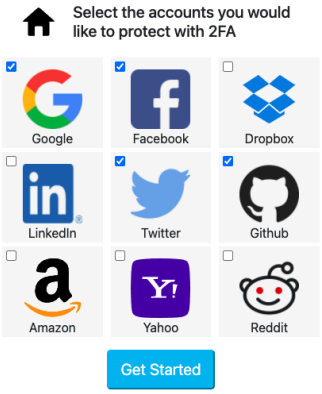


Figure 2: Select accounts to enable 2FA

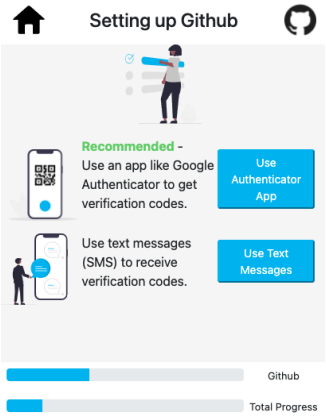


Figure 3: Select SAF Method

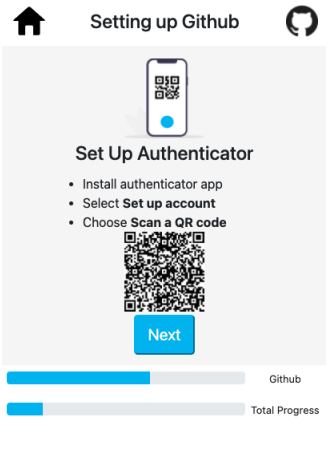


Figure 4: QR code and instructions during GitHub SAF setup

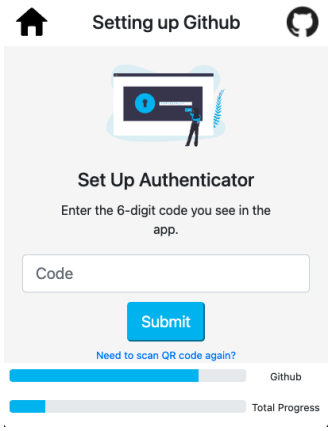


Figure 5: Challenge entry for GitHub

SAF and authentication method to remove from their account. Unlike when setting up a SAF, Step 1 can require users to use their SAF when they are logging in. Also, in Step 2, some websites may require the user to enter their password again before modifying login settings, even if they are already logged into the website.

Step 3 is the removal of the SAF from the user's account. In all the websites we evaluated, this could be fully automated by our SAF manager design. While some websites did require the user to enter the password again, our design temporarily stores the password from when it is entered in Step 1, allowing this second password entry to be automated. However, this introduces a potential vulnerability to a local attacker that gains access to the SAF manager between steps 1 and 3. They will not be required to re-enter the password. None of the websites required a second interaction using the SAF.

4.2 Proof-of-Concept Implementations

We created two proof-of-concept 2FA managers to test the feasibility and usability of our design. The prototypes used the same terminology and workflows, although they had slightly different interfaces that we kept consistent within each prototype. First, we integrated our SAF manager design into a password manager, the popular open-source password manager KeePass. We chose to integrate our design with a password manager as it allows Step 1 of setting up a SAF to be fully automated. In most cases, Step 1 of removing a SAF will also be fully automated. Moreover, password managers already store the list of a user's accounts, allowing the SAF-enhanced password manager to recommend to a user which sites they could register a SAF.

Second, we implemented a stand-alone SAF manager as a browser extension. A browser extension is a natural fit, as users will likely use 2FA within the browser context. One benefit of an extension is that it can monitor the websites a user is visiting and prompt them to increase security by registering a SAF.

Each prototype supported setting up or removing SAFs from nine websites. Limited development resources drove this limitation. Different user interfaces and 2FA setup workflow between websites require implementing a specialized automation script for each supported website. Changes to a website's code may necessitate updates to its automation script. Maintaining these prototypes requires monitoring the supported websites and fixing automation scripts when necessary. Later (see §8.2), we discuss how to scale our design to many websites.

5 Methodology

To understand whether our SAF manager improved the usability of the SAF setup and removal workflows and to elicit feedback on our SAF manager design, we conducted an IRB-approved 60-person between-subjects study. While we developed two working prototypes we designed the study to specifically test the browser extension prototype instead of the password manager. Standalone password managers already lack widespread adoption [27] which could lead to a confounding factor if we did not specifically recruit standalone password manager users. Prior work has shown that those with higher cybersecurity knowledge are more likely to adopt password managers, thus recruiting password manager users would likely lead to a non-generalized sample. Due to the COVID-19 pandemic and a desire to recruit a generalizable sample, we conducted the study remotely using Zoom.

Group	Tool used	Prior Usage
A	Existing website interfaces	No prior experience using 2FA
B	SAF manager	No prior experience using 2FA
C	SAF manager	Prior experience using 2FA

Table 1: User Study Conditions

5.1 Study Design

Our study was designed to answer our four research questions. RQ1–3 directly arise from our SAF manager's design goals. RQ4 arises due to research by Colnago et al. [12] that showed the prior usage of 2FA affects users' acceptance of 2FA generally. As RQ4 intersects with each of RQ1–3, we ended up with six hypotheses we tested through our user study:

H_1 Our SAF manager increases users' ability to setup and remove a SAF.

H_2 Prior experience using 2FA increases users' ability to setup and remove a SAF using the SAF manager.

H_3 Our SAF manager reduces the time needed to setup and remove a SAF.

H_4 Prior experience using 2FA reduces the time needed to setup and remove a SAF using the SAF manager.

H_5 Our SAF manager increases the perceived usability of setting up and removing a SAF.

H_6 Prior experience using 2FA increases the perceived usability of setting up and removing a SAF using the SAF manager.

Note that $H_{1,3,5}$ are comparing performance against existing websites and $H_{2,4,6}$ are comparing performance against users without prior experience using 2FA. For each hypothesis, there is also a corresponding null hypothesis that performance is the same or rose for either SAF manager usage or users with 2FA experience, respectively.

5.1.1 Participant Groups

We grouped participants into one of three groups as shown in Table 1. By comparing results between Groups A and B, we can test $H_{1,3,5}$ and by comparing results between Groups B and C, we can test $H_{2,4,6}$. In a pre-study survey, participants were asked if they had prior experience using time-based one-time passwords (TOTP) or hardware security keys. If so they were assigned to Group C; otherwise, they were randomly assigned to either Group A or B.

5.1.2 Study Setup

Studies were conducted over Zoom, with a study coordinator available to introduce the study, describe the tasks for users to complete, and answer questions about the study itself. Participants' feedback was gathered using a Qualtrics questionnaire. All study materials, including screening survey, study survey, and interview guide, are available online at https://bitbucket.org/isrlauth/saf_manager/src/master/. Using Zoom, we also recorded the participant's screen for later analysis.

Informed consent was obtained at the beginning of each session using the questionnaire. We also confirmed participants' prior experience with 2FA by asking them to look at the apps installed on their devices to see if they were using an authenticator app such as Twilio's Authy or Google Authenticator. Based on their responses we corrected group assignments as necessary. The questionnaire gave instructions on installing the Google Authenticator app (a TOTP SAF) if it was not already installed. Participants in Groups B and C were also instructed to install a browser extension that simulated our extension-based SAF manager prototype (see §5.2 regarding the need to use a simulation). Study coordinators offered assistance if participants encountered issues installing the app or the browser extension. Study coordinators did not help participants complete the SAF setup and removal tasks. Finally, participants were given unique login credentials for the Google, Facebook, and Dropbox test accounts and asked to log in to each account.

Before starting the SAF setup and removal tasks, participants were given a brief introduction to 2FA and TOTP in particular. Study coordinators also read a short description of 2FA and TOTP, explaining the security benefits that 2FA provides.

After the study, participants were debriefed and informed that the extension used in the study was not a working tool but a simulation of the SAF manager we had developed.

5.1.3 Tasks

Participants in all three groups were tasked with setting up three test accounts using the installed Google Authenticator app. Participants were allowed to use any resources they liked but were told that the study coordinators could not assist them with the task. Once this task was completed, participants were instructed to answer questions about their experience in the Qualtrics questionnaire. Next, participants were instructed to remove the Google Authenticator app from the test accounts. In describing the task, study coordinators explained that removing a 2FA method can be helpful if a user obtains a new phone or a device is lost or stolen. After completing the removal task, participants provided feedback about their experience in the Qualtrics questionnaire.

Participants in Group A completed these tasks using each website's existing interfaces and workflows. Participants in Groups B and C completed the task using our SAF manager.

5.1.4 Questionnaire

The setup and removal questionnaires both used the System Usability Scale (SUS) to measure participant sentiment about the usability of the setup processes as a whole (i.e., not the usability of specific websites, but rather the usability of setting up 2FA across multiple accounts at one time). SUS is composed of 10 Likert scale questions, with 1 being "Strongly disagree" and 5 being "Strongly agree" [9]. The responses are used to calculate a SUS score from 0 to 100. An analysis of over 5,000 users across 446 studies found that the average SUS score is 68, with a standard deviation of 12.5 [41]. These scores are often interpreted using percentile rankings or assigned letter grades.

The questionnaire also included questions adapted from Colnago et al.'s [12] work measuring participants' intentions to adopt 2FA in their personal accounts and 2FA's perceived usefulness. Finally, the questionnaire included open-ended questions to investigate what aspects of the setup and removal process participants liked or disliked.

5.2 Pilot Study

We conducted a pilot study over Zoom with 13 participants to test the functionality of our working prototype and the study protocol. Of the 13 participants, 7 used the prototype. During pilot testing, we experienced significant issues with our extension-based SAF manager. We discovered that many websites' 2FA user interfaces varied dramatically depending on the browser version and operating system of the machine running the extension. These changes in the 2FA interfaces caused issues with our automation scripts.

We also found that Chrome throttles browser windows running in the background, which on participants' (slower) machines can cause our automation scripts to time out. We identified two ways to address this issue. First, we could reverse engineer the HTTP requests and responses necessary to interact with services directly, obviating the need for automation scripts. Second, the browser could add a permission for extensions that allows them to run code in the background without rate limiting. Both approaches would require substantial engineering effort. Instead, we chose to create a browser extension that simulated how our prototype would have functioned for the three websites under test.

We built the simulation to match our SAF prototype exactly. For example, TOTP codes expired after 30 secs, real text messages were sent to the phone using Authy, and all credentials were properly validated. Moreover, we used our working prototype to setup 2FA for Google, Facebook, and Dropbox 5 times on the same device using screen capture software to record the process, measuring all network latency, and modifying our simulation to match the recorded latency. Ultimately, the simulation matches the working prototype so well that we believe if users used both they would not be able to tell the difference. In our study, participants assigned to Groups B and C used the simulated SAF manager.

Measure	Items	A	B	C	Total
Gender	Female	13	12	4	29
	Male	7	8	16	31
Age	18-24	4	1	1	6
	25-34	7	5	9	20
	35-44	5	4	2	11
	45-54	2	5	4	11
	55+	2	4	2	9

Table 2: Demographics by Group

5.3 Recruitment

We recruited participants using Prolific. Participants had to be at least 18 years old, fluent in English, and willing to participate in a video interview. Initially, we ran a screening survey. We collected 152 qualifying responses and systematically invited participants to the full study to keep the groups gender-balanced. Unfortunately, only 5 of the 152 qualifying participants accepted an invitation to participate further. Therefore, we abandoned the screening survey and used a Qualtrics scheduling questionnaire indicating participants had to attend an initial scheduling survey and interview to receive compensation. In this questionnaire, participants reported their current 2FA usage and selected a time to participate in a live Zoom call. Participants were compensated 8.55 USD for completing the study. The average completion time was approximately 30 minutes.

5.4 Demographics

In our final study, we specifically recruited equal numbers of male and female participants through Prolific because we did not expect a significant difference in 2FA adoption by gender. However, we found in our scheduling questionnaire that males were more likely to report using an authenticator app or hardware security key. This resulted in Groups A and B having more female participants than males and Group C having more male participants than females. We attempted to balance the gender of participants in Groups A and B. Table 2 gives a complete breakdown of the demographic data for each group.

5.5 Analysis Approach

To test our hypotheses, we analyzed the success rate, task completion time, and perceived usability for each group. We used the STATA 14.0 statistical software package for all quantitative analysis.

Regarding task completion, participants were considered to have successfully completed the task if and only if they could setup or remove the Google Authenticator as the SAF method for all three accounts. To compare the success rate across groups we used a two-sample test of proportion.

Using the recorded video of each participant, we timed how long it took for each participant to complete the setup or removal tasks. We started timing the task when participants took their first action on each website and stopped it when they finished. We then summed up these times to get the participant's overall time. We subtracted the time that the participant asked the coordinator any questions, offered feedback, or if they decided to switch to a new website. We calculated task completion time data only for participants that completed the task successfully.

Perceived usability was measured using the System Usability Scale (SUS).

To reduce the likelihood of false positives due to multiple comparisons, we calculated the False Discovery Rate using Benjamini and Hochberg's method [7] and obtained a significance threshold of 0.0167. All tests reported as significant are below this corrected threshold. In addition, within each group, we tested the mean completion time and average SUS score for normality using the Shapiro-Wilk test. None of them were normally distributed, so we used the Mann-Whitney test for comparisons.

Besides quantitative measures, we also gathered qualitative feedback. After completing all tasks, participants answered open-ended questions about their experiences with setup and removal. Two researchers used inductive coding [47] to identify common themes and ideas expressed by participants. First, the researchers independently read all responses, listing common themes and ideas they found. The researchers then discussed their lists and generated an initial codebook. The researchers then independently coded one-sixth of the data, met to resolve differences, and finalized the codebook. Finally, each researcher coded each response independently, then met to resolve differences. The final codebook is in Appendix A.1.

5.6 Limitations

There are potential limitations related to our sampling method. First, all of our participants were from the United States, which may limit the generalizability of our findings outside the US. Second, the dramatic gender-based difference in prior 2FA experience caused males to be overly represented in Group C, which may have impacted our comparison of Group C to Group B. Third, requiring participants to participate in a video call may have introduced sampling bias. Fourth, using a simulation instead of an actual implementation may have biased results. Nevertheless, we believe the high quality of our simulation was indistinguishable from our SAF manager implementation.

We were primarily investigating whether prior 2FA experience affects the experience with a SAF manager, not the general question of whether prior 2FA experience impacts 2FA usage. Thus, we chose not to investigate a fourth condition that includes users with previous 2FA experience required to set up TOTP manually. Therefore, we cannot make any claims about that condition.

While the results seem promising, using a fictitious task of setting up 2FA on multiple accounts raises questions about the ecological validity of the results. The actual usage of a manager could be much different in practice.

6 Results—Manual

Group A (baseline) helped us identify some challenges users still face while performing manual setup and removal of SAF, with 25% (5/15) failing to complete the setup process on at least one of the three websites. Two failed to set up the SAF for any account, two failed to set up the SAF only on Dropbox, and one failed only on Google. The average time to complete setup on all three websites was 7 min 52 secs. The median SUS score for the setup task was 72.5. Of the 15 participants who completed the setup on all three websites, only one failed to remove the SAF from Dropbox. The average time to remove SAF on all three websites was 2 min 42 secs. The median SUS score for removal was 77.5.

Participants believe it is a lot of work to set up SAF on multiple accounts, which hinders SAF adoption.

> P11 (Group A): *"It would be a lot of work to set it up for all of the accounts that I have..."*

Discoverability of SAF settings Participants using the manual method had difficulty finding the 2FA settings page for each account. On Facebook, for example, participants often looked in the Privacy settings instead of the Security and Login settings. On Google, participants often looked in Gmail settings instead of Google Account settings. As a result, many participants gave up navigating to the settings page and resorted to using a search engine. Nine participants(45%) mentioned that they disliked searching for the correct setup page:

> P3 (Group A): *"Some of the sites have their security settings hidden in a lot of menus."*

For some participants in Group A, finding the correct settings page was the only major concern. Once they found it, they had no trouble.

> P26 (Group A): *"Once you found the right place, it was easy to do."*

Participants also had a hard time finding removal settings.

P83 (Group A): *" I didn't like that Google has so many different security pages/boxes, which makes it a bit confusing to locate the button you are looking for."*

Instructions The final two participants in the manual group that failed to complete the setup task did not understand how TOTP worked. The first set up SMS for each account and incorrectly informed the coordinator that they had added the authenticator app. The second was unable to scan the QR code from the authenticator App. Ten percent of the participants in Group A reported issues with the instructions.

P1 (Group A): *"I did not like that the app did not tell me where to start, it just gave me two options but I originally couldn't figure out where to find the QR code it was referring to."*

Inconsistency – setup Once a participant finds the 2FA settings page, the next issue they would likely encounter is on the *select a 2FA method* page. Unlike Dropbox and Facebook, Google allows users only to enable TOTP once they set up SMS or Google Prompt (push-based). There is no indication that Google even supports TOTP until they do. One participant gave up trying to set up TOTP on Google because they could not find the page to add TOTP.

Dropbox had fewer pain points than Facebook and Google. However, there was a significant issue in the final step of Dropbox. Unlike Google and Facebook, Dropbox has more steps after entering a TOTP code from the authenticator app. Once participants enter their secret code, Dropbox provides them with backup codes to save for when they lose access to their SAF. The two participants that failed to set up TOTP on Dropbox assumed they were at the end of the process and closed the 2FA popup window. This premature setup exit resulted in 2FA not being enabled on the account without notification that setup was incomplete. As such, users believe they have secured their account when it is still only protected with a password. This phenomenon was previously identified [13, 38] but remains an issue five years later.

Inconsistency – removal Each website had two primary methods for removing a 2FA method from an account, (1) a *Turn Off* 2FA button and (2) an *edit* or *manage* button for each 2FA method on the account. Participants' main pain point for this task was that edit buttons behaved differently. For Google, when participants tried to use the edit/remove button to remove TOTP, an error message stated, "Two-Step Verification is not allowed without this method," even though participants had previously set up SMS as a backup method. The error message led all participants to turn off 2FA, removing both methods instead of just TOTP. On Facebook, the manage button allowed users to add a new TOTP app or turn off 2FA entirely. All participants were able to remove TOTP from Facebook. On Dropbox, the edit button allowed users to choose a different 2FA method in place of the one they had enabled. One participant did not notice the small slider for turning off 2FA above the edit button. Upon clicking the edit button and seeing the 2FA setup page, they assumed they had successfully removed the authenticator. This error is especially concerning. If a user believes they have removed a 2FA method from their account and subsequently discards their authenticator (e.g., uninstall their authenticator app or change phone numbers), they will be locked out of their account.

P1 (Group A): *"I did not like that it was inconsistent across the platforms."*

7 Results—SAF Manager

Below we detail the success rate, completion time, and perceived usability for using our SAF manager and how it relates to the manual setup. We also share qualitative results and observations from these studies.

7.1 Success Rate —$H_{1,2}$

Table 3 summarize the success rates for setting up and removing SAF tasks. We found a statistically significant higher setup success rate for participants using our SAF manager (Group A vs. B, H_1, $p = 0.0084$). However, we did not find a statistically significant difference between participants with different prior 2FA experiences (Group B vs. C, H_2, $p = 0.93$). Moreover, we did not find a statistically significant higher removal success rate for participants using our SAF manager (Group A vs. B, H_1, $p = 0.13$). Since there were no failures in Groups B or C, we can not test proportions based on differences in prior 2FA experience (H_2). These results confirm H_1 for the setup phase and fail to confirm H_1's removal phase and H_2.

7.2 Completion Time —$H_{3,4}$

Table 3 summarize the completion time for setting up and removing SAF tasks. We found a statistically significant lower completion time for participants using our SAF manager (Group A vs. B), to (1) setup SAF (H_3, $t(33) = 2.767, p = 0.0057$) and (2) remove SAF ($H_3$, $t(32) = 3.7111, p = 0.0002$), confirming H_3. We did not find a statistically significant difference between participants with different prior 2FA experiences (Group B vs. C), for setup H_4, $t(36) = 1.769, p = 0.0769$) and removal ($H_4$, $t(36) = 1.508, p = 0.1316$), failing to confirm H_4.

7.3 Perceived Usability—$H_{5,6}$

Our SAF manager resolved the concern of discovering the settings by including them as part of its semi-automated

Group	Tool used	Setup Success		Setup Completion Time	Setup SUS	Removal Success		Removal Completion Time	Removal SUS
A	Existing website	75%	(15)	7 min 52 sec	72.5	95%	(19)	2 min 42 sec	77.5
B	SAF manager	100%	(20)	5 min 15 sec	77.2	100%	(20)	1 min 2 sec	95.0
C	SAF manager	90%	(18)	4 min 22 sec	87.5	100%	(20)	1 min 0 sec	91.3

Table 3: Task success rate, mean completion time, and median SUS Score for SAF setup and removal

process, with no users from Groups B or C reporting any discoverability issues.

Five participants (25%) in Group C mentioned the convenience of using the manager to setup SAF on multiple accounts simultaneously:

> P10 (Group C): *"This was *incredibly* easy. Install the extension, select multiple websites, and go. Setting up multiple websites is great, as I can just plug in credentials from my password manager."*

35% of the participants in Groups B and C mentioned the manager's usefulness while setting up a SAF.

Most participants in Group B (90%) and Group C (80%) indicated that the removal process was easy, whereas 55% of the participants in Group A mentioned it was easy. Participants in Group C again recognized that the SAF manager could be helpful when managing SAFs for multiple accounts. Five participants (25%) mentioned liking removing SAFs en masse from accounts:

> P55 (Group C): *"I liked that I could select multiple types of accounts (from multiple services) at one time instead of having to go through the whole process, repetitively, one service at a time."*

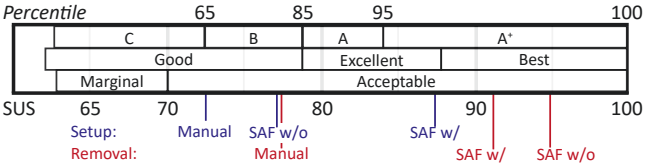
The ease of removing a SAF when switching devices encouraged some participants to use more 2FA:

> P23 (Group B): *"This process makes the disabling of 2FA so easy that I'd want to use it on more websites."*

Many participants who used the SAF manager (groups B and C) showed an interest in using it to proactively set up SAF on other accounts.

> P52 (Group C): *"I would want to secure more of my accounts."*

Figures 7 and 8 provide SUS score distributions for the setup and removal tasks, respectively. The median SUS score for setup was 72.5 for Group A, 77.2 for Group B, and 87.5 for Group C. The median SUS score for removal was 77.5 for Group A, 95.0 for Group B, and 91.3 for Group C.



Manual—Group A, SAF w/o—Group B, SAF w/—Group C

Figure 6: Interpretation of SUS Scores for Setup and Removal

We found a statistically significant increase in perceived usability for participants using our SAF manager to remove SAF (Group A vs. B, H_5, $t(37) = 2.562, p = 0.0104$), confirming the removal phase of H_5. However, we did not find a statistically significant increase in perceived usability for the setup phase and failed to confirm the setup phase of H_5. Also, we did not find any statistically significant increase in SUS between groups B and C, and failed to confirm H_6. Using the work of Bangor et al. [6] and Sauro et al. [40] we contextualize these results within the body of work using SUS to measure a system's usability. These results are presented in Figure 6.

7.4 Pain Points of SAF manager

Every participant in Group B successfully set up TOTP on their accounts. However, because our extension created a popup window, some participants had difficulty finding the window when navigating to different windows. The two participants in Group C that failed setting up TOTP could not figure out how to scan the QR code in the authenticator app. This failure was surprising since both participants indicated they had already used TOTP on their accounts.

One participant from Group C mentioned that they disliked the inconsistency in the requirements across different websites, as explained in Section 6

> P55 (Group C): *"The only thing I didn't like was that the Google account had me verify my number at the same time as the log-in credentials were entered (which was 2FA) and then I had to again follow a similar process to actually set up the 2FA addon. "*

No participants had issues removing TOTP from their accounts using the 2FA manager. However, some

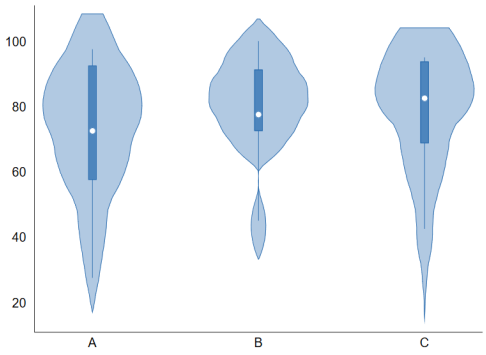


Figure 7: Setup SUS Score

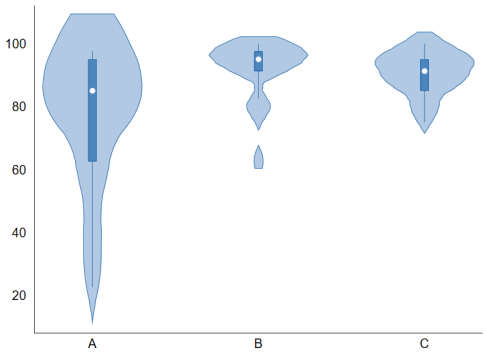


Figure 8: Removal SUS Score

participants were concerned with how fast they removed 2FA from multiple accounts.

> P30 (Group B): *"It seemed almost ... too easy. It took several minutes to initiate 2FA on three accounts and roughly one minute to remove the protection. Just a little too easy, as in: Did that really work?"*

While users were generally pleased with the usability benefits provided by our SAF manager, one-fifth (20%) of participants expressed some hesitation about using a browser extension implementation of a SAF manager:

> P40 (Group C): *"I would likely use this automated tool on my own accounts based on how seamless the whole setup process was without having to delve into the website settings. Although with most Chrome extensions, I don't exactly trust them given how some that I have used in the past have "went rogue". Nonetheless, I would likely use this automated tool for accounts that I use frequently that do not have sensitive information."*

Such hesitation is problematic as it could lower adoption. Still, password managers demonstrate that tens of millions of users are willing to install security-focused extensions [48]. Regardless, there is room for alternative implementations, such as standalone desktop clients or direct browser support. Password managers have adopted these approaches and successfully attracted different types of users [32]. Additional research will be needed to see what new functionality or impediments occur for these approaches.

8 Discussion

8.1 Fragmentation Is Problematic

Throughout our study, it was clear that the key problem facing manual SAF management was the fragmentary nature of the workflow and interfaces. To start, participants often struggled to find where they need to go to begin the process of registering a SAF and enabling two-factor authentication. Then, even if participants were adding the same SAF, they were presented with different workflows, language, and terminology. This injected unnecessary complexity and the potential for confusion into the process. Moreover, as each website is left to implement SAF management functionality by itself, some make it possible to set up a SAF without enabling 2FA, even though this issue was first identified nearly five years ago [13, 38].

Our SAF manager addressed these problems by providing consistent terminology, interfaces, and workflows, regardless of the websites or SAFs in use. For example, our SAF manager provides a consistent entry point for setting up and managing SAFs, removing the need for users to search through disparate web pages to find the 2FA settings. Additionally, the unified terminology, interfaces, and workflows gave participants confidence that they understood how to manage their SAFs. Critically, from a security perspective, our SAF manager prevented participants from incorrectly terminating the setup process before 2FA was correctly set up. Participants were aware of these advantages too, as one-third of participants (32%) self-reported their excitement for the concept of a SAF manager.

Taken together, these results suggest that our proposed SAF manager concept has strong potential and should be further explored and expanded upon in future work. However, even if the concept of a SAF manager never takes off, websites could adopt the unified terminology, interfaces, and workflows found in our SAF manager to improve their SAF management user experience. This alone would be a significant step in the right direction for 2FA deployments.

8.2 Scalability

Our SAF manager requires developing and maintaining a custom script for each website it supports, significantly limiting scalability. Similar issues have also impeded the usability and functionality of password managers [19, 42].

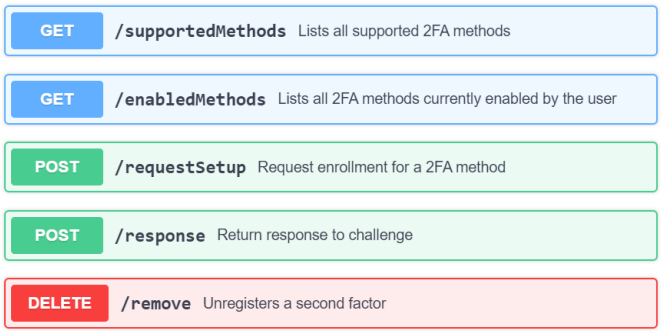


Figure 9: SAF Manager Web API sketch

Nonetheless, their success demonstrates that sufficient engineering effort can overcome these challenges. Below we describe three potential approaches for addressing scalability in our SAF manager (parallel approaches could also benefit password managers).

Crowdsourced automation scripts We could create a shared repository of SAF manager setup scripts that websites or individuals could provide. While this does not reduce the overall amount of work needed to create or maintain automation scripts, it does distribute it such that it would be easier to scale. Moreover, if websites participate, they can update their automation scripts whenever their site changes.

Web standards HTML5 offers standard elements to define user input fields required for authentication, including password and email [29]. Using these declared types allows password managers to automate authentication processes, such as auto-filling users' account credentials [44]. We propose adding additional input field types to support SAF authentication methods, such as displaying QR codes and standardized labels for 2FA progression buttons. Websites that use these standard types allow automated tools like our SAF manager to more easily interface with the website.

SAF manager Web API Even with standardized HTML elements, password managers still have issues interfacing with some websites that do not properly implement those elements or do not use the standards [19]. While expanding web standards to include 2FA fields could ease automation, it would only guarantee a consistent experience between websites if they standardized their 2FA setup flow. Therefore, we propose the creation of a Web API for SAF managers to interface automatically with the website. Such an API could remove any need to write custom scripts for websites.

We sketched a Web API for our SAF manager (see Figure 9). Even with just five endpoints, it supports nearly all of the functionality described in §3). While the message format must support many SAFs, this challenge is manageable. Most importantly, adoption would eliminate per-website customization.

8.3 Password Managers as a SAF

Some password managers, such as 1Password and BitWarden, offer a feature that allows the password manager to generate one-time-passwords [1]. Similarly, some managers are planning support for Passkeys. In contrast to our proposed design, which aids users in managing SAFs, these features turn the password manager into a SAF. While this approach has clear usability benefits, it is not as secure as a traditional 2FA setup, as an attacker who gains access to the manager immediately has access to both passwords and the SAF, which is at odds with how 2FA is supposed to work.

8.4 Recovery

Account recovery is necessary but difficult when a user loses their SAF [38]. For this reason, some websites require users to register multiple SAFs. To support recovery, we suggest two possible extensions to our SAF manager design. First, the SAF manager could include a TOTP code generator that could be registered with websites as a backup SAF in case the primary SAF is lost. Second, when a service requires a backup SAF or storage of recovery codes, the SAF manager could guide users through this process, unifying the backup experience across all websites. It could even be used to store recovery codes. Note, that having the SAF manager store recovery codes or generate TOTP codes could potentially impact security (see §8.3).

8.5 Future Work

Broadly applicable results Despite promising results showing how a SAF manager can improve the 2FA experience, there are fundamental challenges to 2FA adoption that could limit its practical impact. Many users are not concerned enough about security and lack the motivation to add additional steps to their logins [13, 36]. Also, a lack of trust in third-party software could impede adoption, much like password manager adoption [28, 35]. These adoption challenges also intersect with the scalability challenges previously mentioned, as it may not be possible to make headway towards creating and adopting a Web API until there is progress towards 2FA adoption generally.

Even if the vision of a SAF manager never materializes, there are facets of our work that individual websites can incorporate to improve existing systems. First, our user study confirms that users need help finding the 2FA system settings, which may limit adoption. Including 2FA setup in account registration or enrollment could improve adoption rates. Further, websites should make account security more prominent in their settings so it is easier for users to locate.

Second, some websites need clearer directions for completing tasks and warnings when only partially completed. Specifically, we discovered instances during the 2FA removal task where Dropbox users were locked out of their accounts by exiting prematurely. A similar issue occurred in prior work, and it is concerning that the problem persists [38]. Third, consistent terminology, workflow, and interfaces can improve the 2FA experience. Industry professionals and researchers should collaborate to settle on consistent terminology and workflow.

Standardized interfaces Our SAF manager shows clear benefits to unifying the terminology, interfaces, and workflows for SAF management, whether as part of a SAF manager or for individual websites. While our SAF manager provides one vision for structuring SAF management, future research should explore this design space.

As the first step, it would be helpful to systematize the existing terminology, interfaces, and workflows for SAF management, similar to what Simmons et al. [44] did for password managers. Not only would this identify ways in which users and websites already expect to be able to use and manage SAFs, but it would also help identify portions of the design space for SAF management that remain unexplored. Researchers could then build prototypes to explore these design concepts, comparing prototypes against established systems in head-to-head, comparative usability studies [39] to identify the relative strengths and weaknesses of different design approaches. Although the design space for SAF management might appear small at first glance, experience from password manager research (see §2.2) suggests that properly designing an authentication manager is a fundamentally challenging problem.

Measurement studies It would be informative to measure (1) which SAFs are supported by websites, (2) existing terminology, interfaces, and workflows for managing SAFs, and (3) which sites are potentially vulnerable to the design flaws we observed with Dropbox, such as users registering a SAF without enabling 2FA and locking themselves out of their accounts by exiting removal prematurely. Such studies would provide a deeper understanding of the heterogeneous nature of 2FA on the Web and inform standardization efforts.

Authentication managers Our vision is that SAF managers will help users manage SAFs like password managers help them manage passwords. Future research could explore the security and usability of a combined authentication manager. There may be new challenges, especially when it becomes a single point of failure.

9 Conclusion

This paper described a high-level concept for tools to help users manage secondary authentication factors (SAFs). We describe the design of a SAF manager that helps users setup and remove SAFs and detail two prototypes we built based on this design. To evaluate our design, we conducted a user study to measure the usability improvements our SAF manager design provides. Our results show that our design leads to fewer mistakes and reduced task completion time compared to manually setting up and removing 2FA methods. Qualitative results show that users found the semi-automated process easy to use and were enthusiastic about its ability to help them rapidly replace 2FA methods when they lose their 2FA device. Furthermore, our SAF manager prevented fatal errors that users experienced when not using the manager. Our results suggest that our SAF manager concept has strong potential and should be further explored in future work.

Acknowledgments

The authors thank the reviewers and our shepherd for their helpful feedback on the final version of the paper. We thank Xinru Page, Joshua Reynolds, and Daniel Zappala for their feedback on earlier drafts of the paper. We also thank Austin Kolander, Clinton McCardell, and Paul Spencer for their help developing research prototypes and assisting with user studies. This material is based upon work supported by the National Science Foundation under awards CNS-1816929 and CNS-2226404.

References

[1] 1Password. 1Password Features Examined. https://1password.com/product/features/.

[2] Jacob Abbott and Sameer Patil. How mandatory second factor affects the authentication user experience. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[3] Claudia Ziegler Acemyan, Philip Kortum, Jeffrey Xiong, and Dan S. Wallach. 2FA might be secure, but it's not usable: A summative usability assessment of google's two-factor authentication (2FA) methods. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 62(1):1141–1145, 2018.

[4] Nora Alkaldi and Karen Renaud. Why do people adopt, or reject, smartphone password managers? *First European Workshop on Usable Security*, 2016.

[5] Salvatore Aurigemma, Thomas Mattson, and Lori Leonard. So much promise, so little use: What is stopping home end-users from using password manager

applications? In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2017.

[6] 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.

[7] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: series B (Methodological)*, 57(1):289–300, 1995.

[8] Matt Blaze. Upgraded to a new phone, which is like taking a 12 hour refresher course in configuring 2FA. https://twitter.com/mattblaze/status/1181229052520218624, Oct 2019.

[9] John Brooke. SUS-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[10] Sunil Chaudhary, Tiina Schafeitel-Tähtinen, Marko Helenius, and Eleni Berki. Usability, security and trust in password managers: A quest for user-centric properties and features. *Computer Science Review*, 33:69–90, 2019.

[11] 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)*, pages 339–356, 2019.

[12] Jessica Colnago, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Lorrie Cranor, and Nicolas Christin. "It's not actually that horrible": Exploring adoption of two-factor authentication at a university. In *Conference on Human Factors in Computing Systems*, page 456. ACM, 2018.

[13] 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.

[14] Sanchari Das, Andrew Kim, Ben Jelen, Joshua Streiff, L Jean Camp, and Lesa Huber. Why don't older adults adopt two-factor authentication? *CHI Workshop on Designing Interactions for the Ageing Populations - Addressing Global Challenges*, 2020.

[15] Jonathan Dutson, Danny Allen, Dennis Eggett, and Kent Seamons. Don't punish all of us: measuring user attitudes about two-factor authentication. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 119–128. IEEE, 2019.

[16] Michael Fagan, Yusuf Albayram, Mohammad Maifi Hasan Khan, and Ross Buck. An investigation into users' considerations towards using password managers. *Human-centric Computing and Information Sciences*, 7(1):12, 2017.

[17] Michael Fagan and Mohammad Maifi Hasan Khan. Why do they do what they do?: A study of what motivates users to (not) follow computer security advice. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 59–75, 2016.

[18] Nancie Gunson, Diarmid Marshall, Hazel Morton, and Mervyn Jack. User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking. *Computers & Security*, 30(4):208–220, 2011.

[19] Nicolas Huaman, Sabrina Amft, Marten Oltrogge, Yasemin Acar, and Sascha Fahl. They would do better if they worked together: The case of interaction problems between password managers and websites. In *IEEE Symposium on Security and Privacy (SP)*, pages 1367–1381. IEEE, 2021.

[20] Troy Hunt. The only secure password is the one you can't remember. https://www.troyhunt.com/only-secure-password-is-one-you-cant/, Mar 2011.

[21] Troy Hunt. Passwords evolved: Authentication guidance for the modern era. https://www.troyhunt.com/passwords-evolved-authentication\-guidance-for-the-modern-era/, Jul 2017.

[22] Kat Krol, Eleni Philippou, Emiliano De Cristofaro, and M Angela Sasse. "They brought in the horrible key ring thing!" analysing the usability of two-factor authentication in UK online banking. *arXiv preprint arXiv:1501.04434*, 2015.

[23] Juan Lang, Alexei Czeskis, Dirk Balfanz, Marius Schilder, and Sampath Srinivas. Security keys: Practical cryptographic second factors for the modern web. In *International Conference on Financial Cryptography and Data Security*, pages 422–440. Springer, 2016.

[24] Zhiwei Li, Warren He, Devdatta Akhawe, and Dawn Song. The emperor's new password manager: Security analysis of web-based password managers. In *Proceedings of the 23rd USENIX Security Symposium*, pages 465–479, 2014.

[25] Sanam Ghorbani Lyastani, Michael Schilling, Sascha Fahl, Michael Backes, and Sven Bugiel. Better managed than memorized? studying the impact of managers on password strength and reuse. In *Proceedings of the 28th USENIX Security Symposium*, 2018.

[26] 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 (SP)*, pages 268–285, 2020.

[27] Raymond Maclean and Jacques Ophoff. Determining key factors that lead to the adoption of password managers. In *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, pages 1–7. IEEE, 2018.

[28] Peter Mayer, Collins W. Munyendo, Michelle L. Mazurek, and Adam J. Aviv. Why users (don't) use password managers at a large educational institution. In *Proceedings of the 31st USENIX Security Symposium*, pages 1849–1866, Boston, MA, August 2022.

[29] Mozilla. <input type="password"> - HTML: HyperText Markup Language | MDN. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/password, 2022.

[30] Sean Oesch, Anuj Gautam, and Scott Ruoti. The emperor's new autofill framework: a security analysis of autofill on iOS and Android. In *Proceedings of the 37th Annual Computer Security Applications Conference*. ACM, 2021.

[31] Sean Oesch and Scott Ruoti. That was then, this is now: a security evaluation of password generation, storage, and autofill in browser-based password managers. In *Proceedings of the 30th USENIX Security Symposium*, 2020.

[32] Sean Oesch, Scott Ruoti, James Simmons, and Anuj Gautam. "it basically started using me:" An observational study of password manager usage. In *Proceedings of the 40th ACM CHI Conference on Human Factors in Computing Systems*. ACM, 2022.

[33] Shivam Pandey, Tewodros Taffese, Michelle Huang, and Michael D Byrne. Human performance in google's two-factor authentication setup process. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 63, pages 2221–2225. SAGE Publications Sage CA: Los Angeles, CA, 2019.

[34] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, and Nicolas Christin. Why people (don't) use password managers effectively. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS)*, 2019.

[35] Hirak Ray, Flynn Wolf, Ravi Kuber, and Adam J. Aviv. Why older adults (don't) use password managers. In *Proceedings of the 30th USENIX Security Symposium*, 2021.

[36] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, and Kent Seamons. A usability study of five two-factor authentication methods. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS)*, 2019.

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

[38] Joshua Reynolds, Trevor Smith, Ken Reese, Luke Dickinson, Scott Ruoti, and Kent Seamons. A tale of two studies: The best and worst of YubiKey usability. In *IEEE Symposium on Security and Privacy (SP)*, pages 872–888. IEEE, 2018.

[39] Scott Ruoti, Brent Roberts, and Kent Seamons. Authentication melee: a usability analysis of seven web authentication systems. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015.

[40] Jeff Sauro. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC, 2011.

[41] Jeff Sauro and James R. Lewis. *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2012.

[42] Sunyoung Seiler-Hwang, Patricia Arias-Cabarcos, Andrés Marín, Florina Almenares, Daniel Díaz-Sánchez, and Christian Becker. "I don't see why i would ever want to use it:" Analyzing the usability of popular smartphone password managers. In *Proceedings of the 26th ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2019.

[43] David Silver, Suman Jana, Dan Boneh, Eric Chen, and Collin Jackson. Password managers: Attacks and defenses. In *23rd USENIX Security Symposium*, pages 449–464, 2014.

[44] James Simmons, Oumar Diallo, Sean Oesch, and Scott Ruoti. Systematization of password manageruse cases and design paradigms. In *Annual Computer Security Applications Conference*, pages 528–540, 2021.

[45] Elizabeth Stobert and Robert Biddle. Expert password management. In *International Conference on Passwords*, pages 3–20. Springer, 2015.

[46] Ben Stock and Martin Johns. Protecting users against XSS-based password manager abuse. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, pages 183–194. ACM, 2014.

[47] David R Thomas. A general inductive approach for qualitative data analysis. 2003.

[48] Aliza Vigderman. Password manager annual report 2022. https://www.security.org/digital-safety/password-manager-annual-report/, Dec 2022.

[49] Alex Weinert and Lee Walker. Breaking password dependencies: Challenges in the final mile at Microsoft. *RSA Conference*, 2020.

[50] Catherine S Weir, Gary Douglas, Martin Carruthers, and Mervyn Jack. User perceptions of security, convenience and usability for ebanking authentication tokens. *Computers & Security*, 28(1-2):47–62, 2009.

[51] Catherine S Weir, Gary Douglas, Tim Richardson, and Mervyn Jack. Usable Security: User Preferences for Authentication Methods in eBanking and the Effects of Experience. *Interacting with Computers*, 22(3):153–164, 2010.

A Appendix

A.1 Codebook

1. Setup Usability
   (a) Discoverability of 2FA Settings
   (b) Learnability - after doing once it gets easier
   (c) (In)Consistency
   (d) Visibility of System Status
   (e) Multiple Accounts
   (f) Backup Method
   (g) Number of steps
   (h) Specific Requirements
   (i) Adding account to Auth App
   (j) Time/Effort to setup
   (k) Instructions
   (l) Easy/Intuitive process
   (m) QR Code Scanning
2. Security Strategy
   (a) Selective Securing
   (b) Mandatory Security
   (c) Proactive Securing
   (d) Reactive Securing
3. Removal Usability
   (a) Discoverability of 2FA Settings
   (b) Learnability
   (c) (In)Consistency
   (d) Visibility of System Status
   (e) Multiple Accounts
   (f) Number of steps
   (g) Specific Requirements
   (h) Time to remove
   (i) Easy/Intuitive process
4. 2FA Usability
   (a) Concerns for Losing Device
   (b) Annoyance of login
   (c) Require multiple devices
   (d) Ease of Use
   (e) 2FA Method type
   (f) Easy removal leads to increased usage
5. Security concern
   (a) Easy Removal means Security Concern
   (b) Trust
      i. Authenticator App
      ii. Extension
      iii. Websites
   (c) Lack of Awareness about Security Benefit
6. Extension Implementation
   (a) Firefox Support
   (b) Other 2FA method Support
   (c) Extension
   (d) UI
7. Usefulness of Automated Tool
8. Usefulness of 2FA
   (a) Risk Based Authentication is enough
   (b) Security Benefits
   (c) User doesn't have accounts of value
   (d) Wasn't aware of 2FA offering or how easy it is