Knock x Knock: The Design and Evaluation of a Unified Authentication Management System

Eiji Hayashi ehayashi@cs.cmu.edu
Jason I. Hong jasonh@cs.cmu.eu
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213 USA

ABSTRACT
We introduce UniAuth, a set of mechanisms for streamlining authentication to devices and web services. With UniAuth, a user first authenticates himself to his UniAuth client, typically his smartphone or wearable device. His client can then authenticate to other services on his behalf. In this paper, we focus on exploring the user experiences with an early iPhone prototype called Knock x Knock. To manage a variety of accounts securely in a usable way, Knock x Knock incorporates features not supported in existing password managers, such as tiered and location-aware lock control, authentication to laptops via knocking, and storing credentials locally while working with laptops seamlessly. In two field studies, 19 participants used Knock x Knock for one to three weeks with their own devices and accounts. Our participants were highly positive about Knock x Knock, demonstrating the desirability of our approach. We also discuss interesting edge cases and design implications.

Author Keywords
Password; Authentication; Usable Security;

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Reliable authentication is an essential requirement for secure systems. Passwords are the most common form of authentication today; however, a great deal of past research has shown that people have difficulties memorizing and managing passwords in a reliable and secure manner (e.g. [19,22,27]). For instance, people tend to choose simple passwords, re-use passwords, and fall for phishing attacks.

UbiComp '15, September 7–11, 2015, Osaka, Japan. Copyright 2015 © ACM 978-1-4503-3574-4/15/09...$15.00.
http://dx.doi.org/10.1145/2750858.2804279

When passwords were first introduced in the 1960s, computers were a scarce resource, and experts only had to manage a few passwords. However, the security landscape has changed dramatically: there are far more users, each managing numerous passwords, using systems with different security requirements, and facing new kinds of attacks. The computing landscape is changing as well. More and more physical objects are equipped with computation, storage, sensing, and networking capabilities. These smart devices will need some sort of user identification or authentication. However, not all smart devices will have input capabilities suitable for password-based authentication (e.g. typing and pointing).

We are at an inflection point for authentication. The increasing burden on end-users, the growing number of security breaches, and the rise of Internet of Things all point to the need for better user authentication.

One approach is to completely replace passwords. However, a survey paper of authentication techniques suggests that passwords have many positive properties over alternatives [9], in particular ease of deployment. Furthermore, passwords are still good enough for many cases if used appropriately, e.g., if they are long, randomly generated, unique among multiple accounts, and updated periodically.

We take an alternative tack: instead of completely replacing passwords, we propose to build machine-readable interfaces to handle users' credentials, and have a smart device manage these credentials appropriately (e.g., using strong passwords, updating them periodically, and authenticating on behalf of users). By using passwords as our backend, our approach maintains backward compatibility with existing services and user practices. Furthermore, it enables partial adoption of the system without waiting for server-side changes. This approach preserves the positive properties of password-based authentication while addressing the aforementioned problems. Furthermore, letting a smart device manage credentials opens up new opportunities, such as using sensors on the device to authenticate a person to the credential manager (instead of always relying on a master password), supporting authentication to physical devices (as opposed to existing password managers that only support authentication to online services), and offering a potential transition path to stronger forms of authentication in a manner that is transparent to end-users.

We believe that this scenario of having a single device manage all of our authentication needs is a very likely scenario in the near future. What might the user experience be like? What kinds of challenges and unexpected uses will there be? Understanding these issues today could help inform the design of better systems tomorrow.

Towards this end, we present UniAuth, a Unified Authentication Framework for facilitating authentication both for online services as well as physical devices. We also present the design, implementation, and evaluation of Knock x Knock, an example UniAuth authenticator for iPhone. Knock x Knock stores account information for online services and Mac laptops in iPhone. When users access their accounts on Mac, the information is seamlessly transmitted to Mac via Bluetooth Low Energy (BLE). In addition to standard password management features such as storing and auto-filling user IDs and passwords, Knock x Knock has many novel aspects that existing password managers and research systems do not currently support, such as seamlessly storing all account information in one mobile device, multiple security tiers for accounts, and location-aware access control to these tiers.

We conducted a one-week preliminary field study and a three-week field study where 19 participants in total used Knock x Knock on their phones and Macs to manage access to actual accounts and laptops. Participants were very positive about the UniAuth concept as well as Knock x Knock. We also obtained many useful insights for refining the UniAuth Framework.

Through our exploration, we make the following contributions. First, we propose the design of the Unified Authentication Framework that streamlines credential management in password-based authentication. Second, we designed and implemented Knock x Knock, an authenticator for iPhones and Macs. Third we explored what the user experience might be like when users have one device managing all of their user authentication needs. Fourth, we report rich data that can help inform researchers and practitioners in developing unified authenticators and password management systems.

RELATED WORK

Password Management Applications

All major web browsers have a built-in password manager for storing and auto-filling credentials. There are also standalone credential manager apps for computers and smartphones (e.g., [1,3]). While these apps help users store credentials, users often still have to do many steps manually, in some cases typing in passwords, in other cases keeping information in password managers consistent with online services. Additionally, these password managers typically offer all-or-nothing access to credentials, which may be a mismatch with users' needs [16].

There are also many research systems that manage passwords. PassPet [29] and PwdHash [25] are browser extensions that dynamically generate site-specific passwords from a single master password and additional information such as user-chosen name for web sites or a domain name. While these systems do not store passwords, it is challenging to accommodate site-specific password composition policies without having a large dictionary of password policies for each website. WebTicket [15] embeds a credential in an encrypted QR code that can be printed or stored on smartphones. Some systems utilize mobile devices in the context of user authentication. Phoolproof Phishing [10] is an authentication scheme designed to prevent phishing attacks, key loggers, and other kinds of malware. Gray [9] is physical access control system using smartphones. Also, there are some commercial systems that mediate authentications to online services using smartphones (e.g., [4,5]). However they do not provide any user benefit without server-side support, making widespread adoption challenging. There are also no published studies about the use of these systems.

Past work has also looked at evaluating password management tools (e.g., [24]). Li et al. conducted security analyses of five web-based password managers and found that four of them had severe vulnerabilities that allowed attackers to obtain a user's credentials for arbitrary websites [23]. Karole et al. investigated users' perceptions of three different password managers: online, phone-based, and USB-key-based ones. They reported that their participants strongly preferred a phone-based password manager [21]. These works lend support to our design choice of storing account information on one local device.

Password Alternatives

Other authentication systems depend on physical devices. RSA SecurID is a device for two-factor authentication. Users have to type their personal identification number as well as a number shown on the SecurID to be authenticated [7]. However, these tokens require server-side modifications in addition to the high deployment cost of these tokens. As a result, these tokens tend to be used only for accounts with very high security requirements such as bank accounts.

Single Sign-On systems also try to address authentication problems. OAuth [4] is one of the most commonly used Single Sign-on standards. With OAuth, a user can log into third party websites using his account at OAuth service providers such as Facebook and Google. However, involving multiple parties makes the system difficult to understand for users [28].

The FIDO alliance is a group of companies seeking to replace password-based authentication with public-private key pair based authentication [2]. To log into an online account, a user first authenticates himself to his FIDO device, typically using some biometric. Then, the device

and a service complete authentication using a pre-shared public-private key pair. Although this approach could be more secure than password-based authentication, it requires server-side modifications and adoption of FIDO devices at the same time. This would make the early deployment of FIDO challenging. Our work in this paper could help inform the design and deployment of systems like FIDO.

Bonneau et al. conducted a comprehensive analysis of password alternatives [8]. They found that most schemes did better than passwords on security; however, every scheme did worse than passwords on deployability, and that deployability was the biggest barrier for adoption. This insight guided the design of UniAuth, and is why we focus on making UniAuth backwards-compatible with existing password-based systems and enabling its partial adoption without server-side modifications, while also offering a path forward for transitioning beyond passwords.

Investigation of Password Usage in the Wild

Many studies have investigated password usage in the wild (e.g., [13]). Inglesant and Sasse reported mismatches between password policies and users' work contexts [19]. Hayashi and Hong conducted a diary study and concluded that their participants had about 11.4 accounts on average and 84.3% of authentications to these accounts happened either at home or workplace [14]. These works investigated password usage in general to understand its challenges and opportunities. In this work, we sought to investigate what the user experience could be like if people used a single smart device to manage their credentials.

UNIFIED AUTHENTICATION FRAMEWORK OVERVIEW

The core idea behind UniAuth is to have one's smart device handle all tasks related to credential management, such as account creation, authentication, password updates, and account termination, with minimum interaction by users. The main strategy is to offer machine-readable interfaces used by the clients. This approach also allows us to offer several features not supported in existing password management systems, such as notifications of logins and authentication to physical devices. For services that do not support these machine-readable interfaces, clients can still provide core functionalities using heuristics, enabling partial adoption of UniAuth by users without having to wait for server-side changes.

With UniAuth, users only need to authenticate themselves to their smart devices a small number of times a day (with the assistance of sensors) to access their accounts for online services as well as physical devices. The focus of this paper is to investigate the user experience of managing credentials with one device. As such, we only briefly present the underlying concepts of UniAuth to save space.

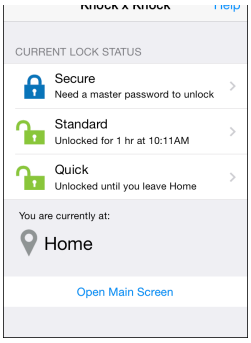
UniAuth consists of three components: a Universal Identity Management Protocol (UIMP), UniAuth clients, and UniAuth proxies. *UIMP* is a set of REST-based APIs that enables UniAuth clients to communicate with services to complete tasks related to credential management. Today, many aspects of authentication are only human-readable or require manual intervention. Examples include password composition policies, password update, account creation, and login pages. UIMP aims to create a protocol that machines can understand and support. If websites and other devices implement UIMP, then any UniAuth client can interact with them with minimum human intervention. UIMP is intentionally designed to mimic the account and authentication functionality that users see in HTML, making it so that only a thin wrapper on top of existing servers is needed to make them compatible with UIMP.

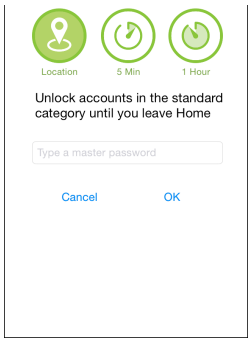
UniAuth clients are authenticators that manage users' credentials using UIMP. The clients can be implemented on many different types of smart devices, e.g., smartphones or wearables. UniAuth clients communicate with services through UIMP on behalf of users. For instance, when a user wants to sign up for a service, her UniAuth client can create an account by generating a strong and unique password that complies with any required password policies, providing requested user information (such as email addresses), and storing the credentials in a secure manner. When she wants to authenticate herself to a service (or a physical device), her UniAuth client automatically provides a credential. To address the bootstrapping problem, UniAuth clients should also work with services that do not support UIMP (e.g. using heuristics to find the password field in a web page). In these cases, functionality is limited to what is possible without UIMP. Nevertheless users can get immediate benefit by using UniAuth clients without waiting for services to support UIMP. This will facilitate adoption of the clients and push service providers to support UIMP.

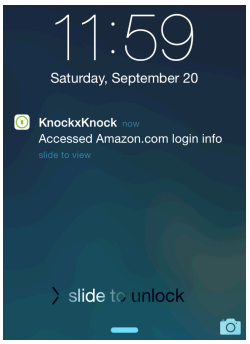
By having one's smart device manage authentication, we can shift the burden of authentication from end-users to their devices. However, a new threat is having one's smart device stolen. We expect UniAuth clients to help protect users' credentials using a set of onboard sensors (e.g., [18]). For example, for places with reasonable physical security (e.g., physical locks) like work and home, UniAuth clients can operate focusing more on convenience, while for places that she rarely or never goes to, or for situations that the system deems risky, it can operate in a high security mode. Similarly, the New York Times web site may only need a low level of assurance, whereas one's bank may want a high level that it is indeed the legitimate user.

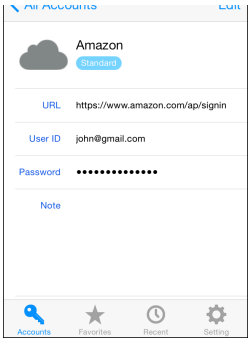
A *UniAuth proxy* is an application typically running on users' computers that mediates the communication between browsers and UniAuth clients (since standard web browsers cannot directly communicate with UniAuth clients).

As noted earlier, UniAuth's underlying user authentication is based on user IDs and passwords. We made this design choice for backward compatibility with existing services, as well as to allow partial adoption of UniAuth. That is, users









(a) Status View **(b) Unlock View** **(c) Notification** **(d) Account Detail View**

gure 1. Knock x Knock displays a status view when launched by a user (a). A user can lock/unlock these tiers based on locations). When accessed from a UniAuth proxy, it shows a notification message (c). A user can see account information including a ssword directly by clicking Open Main Screen and typing his master password (d).

can get immediate benefit by adopting UniAuth clients, without servers having to change anything. However, as servers adopt UniAuth, clients can gain more functionality and security, offering a potentially smoother transition to stronger security, compared to approaches that require end-users and/or servers to completely change (as is the case with FIDO).

In this paper, we developed a UniAuth client and a UniAuth proxy. Then, we evaluated them with existing online services through field studies. This is essentially equal to the early adoption stage, where UniAuth clients and proxies are used with services not supporting UIMP.

KNOCK X KNOCK: A UNIAUTH PROTOTYPE

Knock x Knock is a combined suite comprised of a UniAuth client on iPhone (Figure 1) and a UniAuth proxy on Mac (Figure 2). The proxy consisted of a Mac application and a browser plugin. We named the system Knock x Knock after the well-known American joke intro, and because it lets users physically knock on their Mac twice to login to the Mac. All credentials are stored in the client. The UniAuth proxy consists of browser extensions and a Mac application that mediates communication between the browser extensions and the iPhone app. It also provides an interface that allows users to manage credentials stored in their iPhones using their Mac. Currently, we have browser extensions for Chrome and Safari.

The Mac app connects with the iPhone app via Bluetooth Low Energy (BLE) and connects with the browser extensions using a local web server. Users can also set the iPhone's BLE to have a range of 1 to 15 meters. For both connections, they perform mutual authentication using a pre-shared key to prevent illegitimate connections. Once the BLE connection is established, it is kept until the iPhone goes out of BLE range or the Mac app stops. All data exchanges after the mutual authentication are encrypted using AES-128. Because existing online services do not support UIMP, Knock x Knock currently only supports storing/auto filling credentials for online services and Mac. For the features requiring server-side changes, we showed paper prototypes in our post-study interviews and asked for participants' opinions on these features.

Threat Model

Our primary threats are online/offline attacks against users' accounts such as dictionary attacks and reverse brute force attacks. We also consider attacks targeting users such as phishing attacks. In contrast, we assume that cryptographic primitives protecting the framework are secure and properly used. Thus, communications between entities are secure, and the credential database cannot be compromised without knowing a master password. Also, we assume that there is no malware running on users' computers or smartphones.

Another fundamental assumption in Knock x Knock is that users' iPhones are physically secure. Because users tend to maintain physical proximity to their iPhones, and attackers have to physically come close to the devices, physical attacks (e.g., device thefts) are less likely to happen compared to remote attacks (e.g., dictionary attacks against users' online accounts). However, because the Knock x Knock stores all credentials in one device, an attacker may steal it to obtain credentials. We will discuss the physical attacks later in this paper.

Tiered and Location-Aware Access Control

In terms of access control, most existing password management systems have all-or-nothing access, i.e., allow access to all accounts or to no accounts. However, it can be difficult to satisfy different security and usability requirements for different accounts with this approach [16]. Building on this past work, we propose a tiered access control system for accounts.

In Knock x Knock, each account is stored in one of the three different tiers: Secure, Standard, and Quick (see Figure 1 (a)). Each tier has a lock state and can be locked or unlocked independently. Thus, unlocking the Secure tier does not unlock the Standard or Quick tiers. To access credentials in a tier, the tier should be in unlocked state. For instance, if a user saved his Amazon account in the Standard tier, he has to unlock the Standard tier first to let Knock x Knock provide his credential for Amazon on Mac.

Knock x Knock automatically locks and unlocks the Standard and Quick tier based on whether users are at *trusted locations*, to make account access easier. Users can register trusted locations such as homes and workplaces using their iPhone clients. In the current implementation, a location is a 100 meter radius from a registered geolocation. Our rationale is that past work reported that 84.3% of user authentication for online accounts happened at homes and workplaces [14]. This finding indicated that making user authentication easier at these locations could have a sizeable impact on the usability.

Entering a trusted location unlocks the Quick tier automatically. This lets users log into accounts in the Quick tier on their Mac without touching their phones. Exiting a trusted location locks the Quick tier automatically. The Standard tier can be unlocked by typing a master password on the iPhone (Figure 1 (b)). If this happens at a trusted location, the Standard tier is unlocked until an iPhone exits the trusted location. The Secure tier can be unlocked for one-time access by typing a master password regardless of location. It gets locked after one of the accounts in that tier is accessed. In addition to these options, a user can unlock any tier for a specified period (from five minutes to one day) by typing his master password regardless of locations. It is also technically feasible to use fingerprint or other biometrics as a master password to be more resilient to shoulder surfing attack against a master password. Finally, a user can lock a tier manually at any time.

Storing and Managing Credentials

When a user logs into an account on a web site for the first time, Knock x Knock pops up a dialog asking if he wants to save his credential. The user can edit the name of the entry and its tier. When he clicks OK, account information is sent to his UniAuth client. The account information consists of a user-configured name of the account, a URL, a user ID, a password, and HTML IDs of the user ID field and the password field for logging in. After storing the information, UniAuth automatically fills the credentials when he accesses the same website later. He can also open a main screen by typing his master password to browse and edit stored account information (Figure 1 (d)). This allows users to view their user IDs and passwords to log into accounts manually if needed (e.g., logging into their accounts on foreign computers or sharing an account with others). A user can also browse and edit account information using a UniAuth proxy on one's Mac (Figure 2), but only for categories that are currently unlocked.

One design issue here is whether a system should allow users to see their passwords. If the system would hide the passwords and check URL whenever it sends passwords, the system could prevent phishing attacks. But this may cause some new usability problems (e.g., not being able to log into accounts on other computers). We opted to let users see their passwords. A possible future design is to make it hard to view passwords, or to show a warning message about potential phishing attacks when directly accessing passwords.

Logging into Web Accounts and Mac Laptop

When a user opens a web page with a password field, our browser extension (part of the UniAuth proxy) sends the URL to the UniAuth client over a secure connection. The client looks for a corresponding account for the domain. If an account is stored and its tier is unlocked, it sends the credentials to the browser extension. The browser extension then changes the colors of text fields in the login form to green, indicating that a stored account was found. A user can double click on one of these fields to fill the credential. If the tier is locked, a dialog appears asking the user to unlock the tier.

Users can also log into their Mac by physically knocking on their Mac twice, as if knocking on a door. We implemented this feature to let participants experience logging into physical devices with authenticators. To detect knocking, we use the Mac's microphone and look for sound with amplitude greater than a threshold. Then, the UniAuth proxy extracts acoustic properties and feeds them into a pre-trained SVM classifier. When detecting two successive knockings, the UniAuth proxy requests the appropriate UniAuth client to send a password for the Mac. If the credential for the Mac is stored in the client and its tier is unlocked, the client sends a password back to the proxy; then, the proxy generates fake key type events to fill a password field and log into the Mac automatically. This login will typically happen when a user wakes up a Mac from sleep mode or from a password-protected screensaver. It is also possible for users to press a command button twice instead of knocking their Mac to trigger the login process. Although this knock-to-unlock feature is currently limited to Mac, we foresee that this feature can be expanded to logins for other physical devices.

When a credential is accessed from UniAuth proxy, a UniAuth client shows a notification message (Figure 1 (c)). This notification lets a user know that someone accessed his account if he is still in BLE range but not in front of the computer (e.g., right after he left his desk).

Physical Security of iPhone

As described in our threat model, one potential attack against Knock x Knock is to physically steal a user's iPhone and Mac. After stealing an iPhone, the attackers first have to circumvent the iPhone's lock features (if enabled) and then get out the master password for Knock x Knock to access credentials. If attackers steal both a user's iPhone and Mac, they could access credentials stored in the Quick tier if they can gain access to the Mac web browser and if they know the user's trusted locations. However, without knowing the user's master password, credentials stored in the Standard and Secure tiers cannot be accessed. Considering that credentials stored in the Quick tier would be less critical ones, the expected risk would be acceptable, although further investigation into this type of attack is warranted.

While the attackers were trying to compromise Knock x Knock, the user could either remote wipe the iPhone using the feature provided by Apple, or recover Knock x Knock data from a backup stored in their iCloud on another iPhone and reset all of their passwords. For the services that support UIMP, users can let Knock x Knock reset their passwords using a password-update API provided by UIMP. Thus, we believe that expected risk to Knock x Knock as a result of iPhone theft is reasonably small.

USER STUDY

We conducted a one-week preliminary study and a three-week field study where 19 participants used Knock x Knock on their iPhone and Mac to manage credentials for several real accounts. The overall goal of the studies was to investigate the user experience in using an authenticator, evaluate utility and desirability, and to uncover interesting edge cases. Also, as a credential management tool, Knock x Knock has several novel features not present in existing systems as well as features that have only been investigated individually in lab settings (e.g., [16, 20, 26]). As such, our results provide ecologically valid insights on the effectiveness of these features as a whole.

One-Week Preliminary Field Study

Our first study was a one-week field study with six participants, where we tested our study protocol and looked for possible technical glitches. In general, Knock x Knock worked well. We found one technical issue that caused a problem when iOS killed the Knock x Knock process because of memory pressure. Further analysis showed that this was a bug in iOS. We reported the bug to Apple, and Apple fixed the bug before we started our next study. In terms of the study protocol, we added a few data logging capabilities to better understand participants' usage patterns and to refine our interview questions. Other results found in this study overlapped with ones found in the next study; thus, to save space, we do not report the results here.

Three-Week Field Study

We conducted a three-week-long field study to explore user experiences in managing credentials with Knock x Knock. The study consisted of two in-lab sessions at the start and end, and two interviews in between. In the first session, we installed Knock x Knock on participants' iPhones and Macbooks, and explained how the system worked. We also asked participants to store their user IDs and passwords for their Macbooks and for five of their existing online accounts. Participants chose online accounts ranging from casual accounts such as ones for online bulletin boards to important accounts such as banking accounts. If participants had already saved user IDs and passwords for these accounts in web browsers, we deleted the credentials to let them use Knock x Knock to log into the chosen accounts instead of using browser auto-fill features.

After the first session, we had two semi-structured interviews for each participant at the end of the first and second week. The interviews were done either face-to-face or over the phone and were 15 minutes long. We conducted these interviews to investigate how the participants' perceptions of the system changed over the study period.

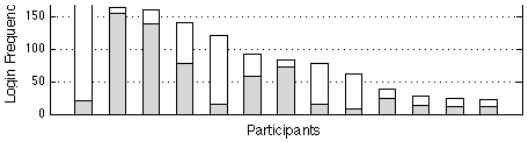
Three weeks later, we had the second in-lab session where we conducted post-study interviews. The interviews were semi-structured and took about 45 minutes. In the interviews, we asked for participants' thoughts on Knock x Knock as well as potential features that Knock x Knock can support with services supporting UIMP. We created paper prototypes showing how these potential features would work and asked for participants' opinions on them. Along with interviews, we collected application logs to analyze participants' usage pattern objectively, totaling 68,032 logs from our participants' Macs and iPhones. We paid $75 USD for completion of the study.

Participants for the Three-Week Field Study

We recruited 13 participants who were using both iPhones and Macbooks, and using either Chrome or Safari as their primary web browsers. The recruitment was done through a university's recruitment website which is meant to public. Seven participants were male and six were female. Their age ranged from 19 to 42 with a mean age of 27. Four of the participants were students, one was a university staff, seven were employed outside of the university, and one was unemployed. All participants, except one participant using KeePass [3], used browser password managers. In addition to browser password managers, four participants used physical memos. Two participants used text files. One participant used 1Password [1] to manage their credentials.

Summary of Knock x Knock Usage

On average, participants stored 7.7 accounts after three weeks of using Knock x Knock (1.1, 3.4, and 3.3 accounts



ure 3. Numbers of logins for each participant in rank ler. White denotes logins to Mac and gray represents logins nline accounts.

in the Secure, Standard, and Quick categories respectively). In the first session, we asked them to register five online accounts and a Mac account. After the first session, six participants added one to nine additional accounts on their own accord. All participants stored their accounts in at least two different tiers. Eight participants used all tiers.

Our participants logged into their Mac 10 to 171 times with a mean of 48.2 (SD=50.2). They also logged into their online accounts 9 to 155 times with a mean of 45.1 (SD=47.8). In total, they logged into their accounts 22 to 192 times with a mean of 93.3 (SD=58.5). Figure 3 shows the distribution of the login frequencies. The white and gray parts of the bars represent logins for Mac and online accounts, respectively. The data showed that the number of logins for online accounts was a few times a day on average. This was because most of our participants did not log out from their accounts. In the post-study interview, most of our participants told us that they usually kept themselves logged into online services and put their computers in sleep mode when they finish. Thus, logins happened only when they restart browsers, typically by restarting their computers. Altogether, the data showed that our participants had a reasonable amount of exposure to Knock x Knock to evaluate it.

In the semi-structured interviews at the end of weeks one and two, we asked about participants' experiences with and perceptions of Knock x Knock. Our participants were generally positive about Knock x Knock and its features. After two weeks, all participants told us that their experiences using Knock x Knock were consistent with those reported in previous interviews. This indicates that three weeks was long enough to mitigate novelty effects.

In the post-interview, we solicited participants' opinions on Knock x Knock using a 5-point Likert scale (5 = very positive) as well as open questions. In the following description, the numbers in parentheses denote the medians/means of our participants' ratings. In general, participants were very positive about Knock x Knock. They agreed (4/4.1) with the sentence "I will use Knock x Knock if it is available." P15 commented, "This is not only convenient. If I were the person with million passwords, this streamlines everything. But then also being someone who is less secure, really really makes it very simple and helps make things more secure." P10 also commented on its security, "It seems more secure since you could use that when the phone is close to a computer you are accessing compared to randomly save passwords in a browser." During the interviews, all participants asked whether there was a plan for public release. Two participants asked whether they could keep using Knock x Knock after the study. These responses clearly illustrated that participants were very positive about Knock x Knock.

We also asked whether they noticed any increase in battery consumption on their iPhones, with only one participant saying that they did. When we asked participants to describe disadvantages of Knock x Knock, they noted that sometimes establishing Bluetooth connection between an iPhone and a Mac was slow. The median connection time was 6.7 seconds. This is because 42% of the connections took longer than 10 seconds due to a bug in the iOS Bluetooth library. This bug has since been fixed in iOS 8.1. Excluding the connections affected by the bug, the median connection time was 2.4 seconds.

Storing All Accounts in a Device Makes Things Simple

Our participants rated storing account information in Knock x Knock on their iPhones as secure (4/3.9) and easy to use (4/4.2). Our participants were very positive about storing account information in one device. P15 commented, "You don't have to invest a lot of time in thinking through what is my password, where did I store it. […] The transition to come to put them all here makes things very simple." Furthermore, many participants believed that mobile phones were more secure than computers because they were more personal devices. P5 noted, "No one else has my iPhone because it's always with me. No one knows my [PIN] code. I let people use my computer, but don't let people use my phone." Two participants were concerned that someone may gain physical access to their phones; however, they also commented that they were not too worried since they always keep their phones with them. These results imply that our participants were aware of the risks of storing their credentials in their iPhones (e.g., device theft); however, they perceived that the benefit of using Knock x Knock outweighed the expected risks.

Knocking Mac to Unlock was Enjoyable

Our participants rated logging into their Mac using Knock x Knock as secure (4/3.7) and easy to use (5/4.3). Participants generally thought it was secure because of proximity. P13 said, "If my phone is nearby, and I'm nearby." P5 also commented on the combination of tiered security. She said, "It's good, because you can control on phone what other accounts they can access. Even if someone logins to my computer, just having my computer doesn't mean they can access everything else."

Three participants reported that knocking their Macs to unlock was enjoyable. While users regard most of the security systems as *burden*, it was very interesting that they reported it enjoyable. In terms of security, P1 commented, "I think it's OK. Even if someone knocks my computer right after I leave my office, I will receive a notification [on

my phone]. Then, I can come back to see what's going on because I'm still in a BT [Bluetooth] range." P5 found an interesting use case that we had not expected, commenting "I share my computer all the time at home. I don't want them to know [my passwords]." This finding suggests that proximity could be useful for certain scenarios, for example for parents and children, or for guests to one's home.

P5 also reported an interesting side effect, noting "One time, my friend asked me what I was doing on my computer, I explained it, and he was really interested." The observability of a security feature has been suggested as one factor that can help facilitate wider adoption [11]. On the other hand, two participants showed concerns over observability. P10 noted, "Someone can just come and click this button especially when this becomes something widely adopted. […] There are cases where I have to leave my stuff [both laptop and phone] like this [on my desk]." For these cases, users can opt out from unlocking one's Mac via Knock x Knock by not storing its credentials. Other possible solutions might include putting Knock x Knock on wearable devices to maintain closer proximity to users.

Not Unlocking Everything at Once

Our participants were very positive about tiered security levels. They rated it as very secure (4/4.6) and very easy to use (5/4.6). P7 reported, "I think it's secure because I have an option of which one to unlock. So, it's not something like you turn on everything at once. There are different security levels for different websites." Interestingly, our participant using 1Password (P1) told us "1Password locks when a screensaver becomes active. So, I have to type my master password again and again throughout a day. I need it to protect important accounts, but, I don't want to type the master password to access a news website." If a password manager only has one tier for accounts, it makes sense to put more weight on security rather than usability so as to protect highly important accounts. However, this approach is suboptimal for people storing many less important accounts. Having multiple tiers offers users a flexible way of balancing security and usability.

One design issue here is, what number of tiers should we have? In the post-study interviews, we asked our participants whether three was appropriate for them. Nine said that three was appropriate. Three answered that they needed only two for now, but having three tiers made sense to them. One said that two tiers would be better in terms of simplicity. None of the participants wanted to have more than three tiers. These results indicated that three was a reasonable number for tiered account management.

Tiered Access Control with Location-Awareness

Knock x Knock locks/unlocks Quick and Standard categories based on whether participants' phones are in trusted locations. Ten of our participants registered two locations and three of them registered three locations. These were mostly their workplaces and homes where physical access was limited. Our participants rated this feature as very secure (5/4.6) and easy to use (4/4.2). P10 commented on its security, saying "I trust people at my home or workplace [enough] to use [location-based lock control]. Or, there is no huge risk of someone hacking into my places. So I think it's secure." P13 commented on its usability, "Especially for the quick category, […] these are accounts that I really don't care. So, having easy access is important." These comments indicate that, combined with tiered access control, location made access to the accounts in Standard and especially Quick tiers very easy while maintaining desirable levels of security for participants.

Our simple definition of a trusted location (100 meters from a user specified location) was sufficient for most people, but we also saw some hard cases. For example, a university staff member (P7) reported that she regularly visited multiple buildings on campus to attend meetings, and that she opted to unlock categories for a day instead of registering all of them as trusted locations. One possible solution is more flexible or ad hoc configurations of trusted locations. Another possibility is to automatically infer trusted locations. However, it is important to note that our participants reported that they chose trusted locations based on their perceptions of security at these places rather than frequency of visits. For example, P12 noted, "Even if I am frequently at a coffee shop, I don't want to add it to [trusted] locations because its security is unclear." We believe that algorithmic techniques for determining the level of physical security of a location could be a compelling area for future research, for the case above, as well as for modulating the level of security needed for Internet of Things.

Potential Features for Future Versions

We also showed participants paper prototypes of three UniAuth features that require server-side modifications, to gauge the potential usefulness of these features.

Unified Account Creation

The Unified Account Creation feature provides users a way to create their accounts easily. When a user visits a UIMP-enabled website for the first time, Knock x Knock can ask if he wants to create an account. If so, it displays a dialog with input fields for all information requested by the website such as a name and an email address. These fields are pre-populated with personal information stored in Knock x Knock along with a randomly generated password. Also, if he prefers, he can overwrite the password with his own manually created password. Then, when he clicks the OK button, Knock x Knock communicates with the website to create an account and store the account information.

Participants were generally positive about this feature, rating it as useful (4/4.2) and agreeing that they would use it if it were available in Knock x Knock (4/4.1). Also, they agreed that they were not concerned about storing personal information in Knock x Knock (4/4.0). Most participants

preferred that they did not have to type their information multiple times to create accounts for different services. Three of them specifically commented that they liked that passwords were randomly generated. Two participants commented that this feature smoothed out the account creation processes. P15 noted, "When you create your account, it signs you in, and you buy something. But, then, you have to log out, revisit a website, and log in to save a password. This makes all in one seamless step."

Managing Password Updates

With this feature, users can receive a reminder after a specified time since one's last password update. Users can select the period (e.g., 3 months), or it can be specified by a service if periodic password updates are enforced. When receiving a reminder, users can manually update a password in Knock x Knock via UIMP, or let Knock x Knock automatically update a password and notify afterward.

Our participants were quite positive about this password update management feature, rating it as useful (4/4.4) and strongly agreeing they would use it if available (5/4.5). Only one participant said he would configure his passwords by himself when receiving password update requests. Other participants preferred to let Knock x Knock generate random passwords. Interestingly, this result goes against Hayashi et al.'s finding that that users did not like using automatically generated passwords [17]. We believe the discrepancy is due to the expected availability of password management tools. While the above study was about password managers in general (software, paper, etc.), our participants evaluated this feature as a part of Knock x Knock on users' iPhones, which are almost always available to them. In our post-study interviews, four participants commented that it was important for them to be able to see their passwords on their phones when using randomly generated passwords. P15 noted, "[With automatic password update] I even don't know what my password is. But, it is so easy to go to my phone, type the master code and look at it." These results suggest a fallback for doing authentication manually is necessary to facilitate adoption of such systems even if this fallback is rarely used.

Server-Side Account Access Notification

In Knock x Knock, we have already implemented a notification feature that shows a message on one's iPhone when account information is accessed by a UniAuth proxy. On the other hand, this feature would be better if implemented on the server-side, sending notifications to users when somebody accesses a user's account.

Our participants rated this feature very useful (5/4.5) and they strongly agreed that they wanted to use this feature if it were available (4/4.5). P8 commented, "I think it's really useful and makes me trust Knock x Knock more. It makes me feel more secure." Participants also requested detailed control on when to receive notifications. P8 said, "I don't want to receive notifications when someone accessed using Knock x Knock. But, something more important ones, like my bank accounts, I'd like to receive notification always." Eight participants mentioned they wanted to receive notifications only for accounts that were accessed without using Knock x Knock. As illustrated by these responses, our participants strongly desired notifications, but did not necessarily want to be overwhelmed by them. One possible design option is to still receive all notifications, but filter out those that Knock x Knock was expecting.

DISCUSSION

Our results also provided interesting implications for how the design of a credential management system affected user acceptances.

Physical Proximity Strongly Affects Perceived Security

There are many possible options for storing credentials: cloud storage, our laptops and desktops, smartphones, and wearable devices. Our results suggest that storing information in a place physically close to users had a very positive effect on our participants' trust in the system. In the post-study interview, many participants commented that they felt safe because their passwords were stored on their phones, which were always nearby.

Researchers rightfully focus more on underlying security mechanisms rather than *perceived* security. However, we argue that users are also influenced by their perceptions of security, which might also include aspects of usability and utility. In the context of an authenticator, from a technical perspective, one's credentials cannot be easily accessed as long as they are encrypted with a master password. However, our participants were still worried that attackers could access the credentials because of insufficient understanding of cryptography. In contrast, physical security is easier to understand. Participants believed that, because no one touches their phones and because credentials are transmitted using short-range wireless, it would be difficult for attackers to access their credentials. This finding suggests that improving perceived security could be as important as improving actual security to facilitate wider adoption of a new security system.

Participants also preferred physical proximity in terms of its availability. They liked the fact that, if they have their phones with batteries charged, they can access their passwords without relying on external infrastructures such as network connectivity and a centralized server. P6 said, "Your passwords are around you all the time. When you need them, they are already there." These suggest that credentials should be stored in smartphones or wearable devices rather than in computers or cloud storages. This would make people feel more comfortable and to facilitate wider adoption.

Guaranteeing Baseline Availability

During the post-study interviews, our participants showed strong preference for directly accessing passwords. For

instance, P7 noted, "it's also important for me to have an option to see my passwords, just in case." However, interestingly, they also admitted that they would probably not do so in practice.

These findings imply that participants understandably want high availability to their credentials. Some aspects of availability, such as battery life, physical closeness, loss, or theft, were clear to participants. However, other parts, such as BLE connection and location awareness, were less clear. To assuage concerns, Knock x Knock provides a fallback case, letting users see their passwords on their phones after entering a master password.

Simple fallback features like this, even if rarely used, may be useful in helping to convince participants of the reliability of new kinds of authenticators. As such, we recommend offering users backup options to guarantee a certain level of availability even in cases where authenticators do not work as expected. One tradeoff, however, is that fallback features would make authenticators less resilient against social engineering attacks. Making it harder to access passwords and presenting reminders of social engineering attacks can help mitigate potential risks.

A Path Towards Better User Authentication

A great deal of research has proposed alternatives to password-based user authentication; however, all of them have their challenges in their deployability [8]. For instance, the current FIDO specifications [2] require modifications to servers *as well as* adoption of FIDO clients by users. Changing one or the other alone for FIDO does not provide immediate benefit, which will likely make initial adoption very challenging.

In contrast, UniAuth clients provide immediate benefit to users by supporting existing password-based authentication without server-side modifications. This should facilitate initial adoption. Once enough users adopt the clients, supporting UIMP provides benefit to service providers because they can streamline account management, such as account creation and password updates. Finally, after users become familiar with letting clients manage authentication and after service providers adopt UniAuth, we can replace passwords with stronger methods (e.g., private-public key pair), with minimum impact on the user experience.

LIMITATIONS

We believe that our work has provided novel insights on credential management tools and frameworks. However, it also has several limitations. For example, our field studies were not long enough to capture long-term effects on participants' credential management practices.

We also focused our evaluation on user acceptance rather than security, because a system with low desirability will be unlikely to be adopted, regardless of its security. As a result, our security analysis in this paper is limited to users' subjective evaluations. More formal evaluation of its security is necessary. However, we still believe that our system addresses a class of security issues in password-based authentication such as weak and reused passwords, and that our work provides insightful design guidelines for practitioners, standards groups, and researchers.

Our system also needs better protection against phone theft. Currently, users can remotely wipe credentials and recover them to new phones from backups. However, for better protection, Knock x Knock could incorporate sensor-based anomaly detection algorithms to throttle access to the client.

CONCLUSION

In the near future, it is very likely that a single smart device manages our authentication needs. Our goal was to understand what this user experience might be like. We evaluated a combination of several features that balance usability with security, looking at common uses as well as edge cases.

Our results demonstrated that our participants were very positive about the concept of UniAuth as well as our prototype implementation, Knock x Knock. After trying Knock x Knock for three weeks with their own devices and accounts, our participants reported that the combination of tiered access control and location-based access control worked very well, and felt that Knock x Knock improved *both* security and usability. We also report some novel uses that we did not expect, including people using proximity to share devices, as well as high mobility users needing more flexible definitions of trusted locations. We also present findings on the importance of physical proximity and the need for baseline availability. We believe that these insights can significantly contribute to our community and to product designers in developing better user authentication frameworks.

REFERENCES

1. 1Password. http://agilebits.com/. Feb. 26, 2015.
2. FIDO. https://fidoalliance.org/. Feb. 26, 2015.
3. KeePass. http://keepass.info/. Feb. 26, 2015.
4. LaunchKey. https://launchkey.com/. Feb. 26, 2015.
5. Nok Nok. https://www.noknok.com/. Feb. 26, 2015.
6. OAuth. http://oauth.net/. Feb. 26, 2015.
7. RSA SecurID, http://www.emc.com/. Feb. 26, 2015.
8. Bonneau, J., Herley, C., Oorschot, P.C.V., and Stajano, F. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. IEEE, 553–567.
9. Bauer, L., Cranor, L.F., Reiter, M.K., and Vaniea, K. Lessons Learned from the Deployment of a Smartphone-Based Access-Control System. In *Proc.* SOUPS (2007), 64–75.

10. Parno ,B., Kuo, C. and Perrig, A. Phoolproof Phishing Prevention. In *Proc. of* the Financial Cryptography and Data Security (2006), 1-19.
11. Das, S., Kim, H.J., Dabbish, L., and Hong, J. The Effect of Social Influence on Security Sensitivity. In *Proc. of* SOUPS (2014).
12. Everitt, K., Bragin, T., Fogarty, J., and Kohno, T. A Comprehensive Study of Frequency, Interference, and Training of Multiple Graphical Passwords. In *Proc. of* SIGCHI (2009), 889-898.
13. Gaw, S. and Felten, E. Password Management Strategies for Online Accounts. In *Proc. of* SOUPS (2006), 44-55.
14. Hayashi, E. and Hong, J. A Diary Study of Password Usage in Daily Life. In *Proc. of* CHI (2011), 2627-2630.
15. Hayashi, E, Pendleton, B, Ozenc, F, and Hong, J. WebTicket: Account Management Using Printable Tokens. In *Proc. of* SIGCHI (2012), 997-1006.
16. Hayashi, E., Riva, O., Strauss, K., Brush, A.J.B., and Schechter, S. Goldilocks and the Two Mobile Devices: Going beyond All-or-Nothing Access to a Device's Applications. In *Proc. of* SOUPS (2012).
17. Hayashi, E. and Hong, J. "It's Hidden in My Computer": Exploring Account Management Tools and Behaviors, CMU-CyLab-13-007 (2013).
18. Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason I. Hong, and Ian Oakley. CASA: Context-Aware Scalable Authentication. In *Proc. of* SOUPS (2013).
19. Inglesant, P.G. and Sasse, M.A. The True Cost of Unusable Password Policies: Password Use in the Wild. In *Proc. of* SIGCHI, (2010), 383-392.
20. Kalamandeen A., Scannell A., Lara E., Sheth A. and LaMarca A. Ensemble: cooperative proximity-based authentication. In *Proc. of* Mobisys (2010), 331-344.
21. Karole, A., Saxena, N. and Christin, N. A Comparative Usability Evaluation of Traditional Password Managers. Lecture Notes in Computer Science. (2010), 233-251.
22. Kuo C., Romanosky S., Cranor L. Human selection of mnemonic phrase-based passwords. In *Proc. of* SOUPS, (2006), 67-78.
23. Li, Z., He, W., Akhawe, D., and Song, D. The emperor's new password manager: Security analysis of web-based password managers. In *Proc. of* USENIX Security, (2014), 465-479.
24. McCarney, D., Barrera, D., Clark, J., and Chiasson, S. Tapas: Design, Implementation, and Usability Evaluation of a Password Manager. In *Proc. of* ACSAC, (2012), 89-98.
25. Ross, B., Jackson, C., Miyake, N., Boneh, D. and Mitchell, J.C. Stronger Password Authentication Using Browser Extensions. In *Proc. of* USENIX Security, (2005), 17-32.
26. Seifert J., De Luca A., Conradi B. and Hussmann H. TreasurePhone: Context-Sensitive User Data Protection on Mobile Phones. In *Proc. of* Pervasive (2010).
27. Shay R., Komanduri S., Kelley P. G., Leon P. G., Mazurek M. L., Bauer L., Christin N., Cranor L. F. Encountering stronger password requirements: user attitudes and behaviors. In *Proc. of* SOUPS (2010).
28. Sun, S.T., Pospisil, E., Muslukhov, I., Dindar, N., Hawkey, K., and Beznosov, K. Investigating Users' Perspectives of Web Single Sign-On. ACM Transactions on Internet Technology 13, 1 (2013), 1–35.
29. Yee, K. and Sitaker, K. Passpet: Convenient Password Management and Phishing Protection. In *Proc. of* SOUPS, (2006), 12–14.