SoK: Web Authentication and Recovery in the Age of End-to-End Encryption

Jenny Blessing
University of Cambridge
jenny.blessing@cl.cam.ac.uk

Daniel Hugenroth
University of Cambridge
daniel.hugenroth@cl.cam.ac.uk

Ross J. Anderson
University of Cambridge
University of Edinburgh

Alastair R. Beresford
University of Cambridge
alastair.beresford@cl.cam.ac.uk

Abstract

The advent of end-to-end encryption (E2EE) has brought new challenges for usable authentication and recovery. Compared to regular web services, the nature of E2EE requires that the provider cannot recover data for users who have forgotten passwords or lost devices. More robust recovery schemes are therefore required, leading to a plethora of solutions ranging from randomly-generated recovery codes to social authentication. These implications have spread to new forms of authentication and legacy web services: passwordless authentication ("passkeys") has become a promising candidate to replace passwords altogether, but is inherently device-bound. However, users expect that they can login from multiple devices and recover their passwords in case of device loss—prompting providers to sync credentials to cloud storage using E2EE and making contemporary authentication for even non-E2EE services dependent on E2EE. Hence, E2EE authentication quickly becomes relevant not only for a niche group of dedicated E2EE enthusiasts but for the general public using the passwordless authentication techniques promoted by their device vendors.

In this paper we systematize existing research literature and industry practice relating to security, privacy, usability, and recoverability of both end-user authentication to E2EE services and the use of E2EE in securing backend credential databases. We investigate authentication and recovery schemes in all widely-used E2EE web services, analyze syncing protocols for E2EE credential managers, and survey passwordless authentication deployment in the top-300 most popular websites. Finally, we present concrete research directions based on observed gaps between industry deployment and academic literature.

Keywords

authentication, E2EE, web, passwords, passkeys, recovery

1 Introduction

Password-based threats have plagued web authentication for as long as the Internet has existed, with Google declaring passwords to be "the single biggest threat to your online security" in 2021 [265]. While industry providers have deployed a wide variety of end-user authentication and recovery schemes in the past decade, developing an authentication scheme that offers a desirable balance of security, privacy, usability, and recoverability for a diverse population of users is an enormous challenge. The inherent difficulty of resolving these trade-offs is the central reason why passwords persist [63, 172, 325]. Even so, industry providers have made significant strides in mitigating password-based threats, such as using browser metadata to detect suspicious logins [138, 284, 416] and offering "single sign-on" options to centralize authentication with a single account.

We are entering a new era: the FIDO2 standard [129] (jointly developed by the World Wide Web Consortium [W3C] and the FIDO Alliance) provides a password-less authentication protocol based on public-key cryptography, commonly referred to as *passkeys*. Passkeys have recently been deployed across all major operating systems and web browsers, and as of May 2024 over 400 million Google accounts have set up a passkey [155]. End-users are able to authenticate to remote web servers using their device authentication (e.g., fingerprint, PIN, etc.) to unlock access to the relevant key material. While passwords as a whole will not disappear any time soon, since they remain the most usable authentication scheme for many, especially at-risk demographics [22, 340], passkeys represent the first genuine contender for password replacement with the backing of major industry players. However, there are still unresolved questions around usability, security, and recoverability, all of which will have a major impact on end-user adoption. To improve usability, users are offered the option to sync device passkeys to either the device's cloud provider or a third-party cloud credential manager, where stored keys are end-to-end encrypted such that they are inaccessible to the provider.

The ability to protect the confidentiality of user data from the third-party service provider hosting the data is arguably the single most significant improvement in end-user privacy in recent years. A widely deployed PET, E2EE guards against a wide range of threats to user privacy ranging from insider employees and Internet Service Providers (ISPs) to governments, and is only becoming more widespread with each year. A small but notable number of consumer-facing cloud storage and email providers have begun protecting user data such as photos, videos and documents with E2EE. This marks an important shift, as E2EE has historically been limited to securing ephemeral communications, or particularly sensitive data such as passwords.

The combination of E2EE credential storage and E2EE messaging, email, and cloud storage services represents a rapid shift towards deploying E2EE within web authentication and mobile apps.

Proceedings on Privacy Enhancing Technologies 2025(3), 560–589

https://doi.org/10.56553/popets-2025-0113

While some E2EE services are only adopted by a niche segment of users, passwordless authentication and accompanying password managers are targeted at the general public. Though an individual website may not adopt E2EE in any capacity, the credentials of a growing percentage of users will be E2EE, raising the prospect of inadvertent loss of access. This development raises concerns around data recoverability since, by definition, service providers cannot access user data in an E2EE service. Therefore users must take an active role in maintaining access to their account, often including the responsibility for storing their decryption keys.

End-user usability has been one of the most significant barriers to large-scale adoption of E2EE for long-term storage, and it is critical to identify specific research gaps and coalesce on standards to increase widespread E2EE adoption. Providers offer a variety of recovery mechanisms to prevent users losing access to their data, some more user-friendly than others, but there is an unavoidable trade-off in user account security: the easier it is for a user to recover their account, the easier it is for an attacker to gain access. At the same time, loss of cloud storage containing years of photo and document storage is a much more concerning prospect than loss of ephemeral communication history for most users, motivating a fresh look at industry deployments. End-user authentication is now at an inflection point, as companies are in the process of deploying novel security and authentication schemes representing a significant departure from existing user experiences.

In this work, we systematize the usage and trade-offs of E2EE in web authentication and recovery. There are two primary requirements for general-purpose authentication credentials: (1) credentials must be syncable across multiple devices (including cloud backup) and (2) credentials must be readily accessible but not guessable. E2EE is critically relevant to both of these goals, but in ways nuanced and complex: for goal (1), E2EE provides vital benefits as it would not be possible to securely sync and back up credentials without E2EE. For goal (2), however, E2EE services undeniably complicate authentication—the risk of account loss has prompted providers to deploy authentication and recovery schemes that are both more diverse and more easily compromised.

We begin with an in-depth discussion of cloud-synced credential managers, which form the foundation of contemporary authentication architectures. All widely used credential managers depend on E2EE to secure a user's authentication credentials even while passing through a third-party server, but there are important security distinctions among the various syncing architectures. We focus particularly on passkeys as a novel authentication credential dependent on E2EE cloud syncing for usability by the general public, and demonstrate that while E2EE cloud sync is essential for usability it nonetheless brings moderate security drawbacks in comparison to non-syncable credentials bound to device hardware. We conclude this section by conducting the first survey of passwordless authentication adoption among the most widely used websites within the U.S. as of May 2024 to better understand which schemes are deployed and where.

We further conduct a comprehensive review of account recovery procedures for E2EE web services (including E2EE cloud storage, email, and messaging), finding that authentication methods in this context vary widely. The majority of service providers rely on asking the user to manually store a recovery key even as more usable variations are feasible, a design choice that arguably makes end users less likely to enable E2EE backups and ultimately undermines the very significant privacy enhancement E2EE provides. Some authentication factors that may be overly cumbersome as part of a daily authentication scheme, such as authentication using trusted contacts, are worth revisiting in the context of E2EE recovery.

Our specific contributions are the following:

- Systematize the security of E2EE cloud credential syncing, including important security distinctions between high-level syncing protocols.
- Provide in-depth discussion of passkey variations (device-bound vs synced) and quantify current passwordless authentication deployment.
- Investigate currently deployed authentication and recovery schemes in E2EE web services.
- Systematize existing research and industry practice relating to privacy, usability, and recoverability of all non-E2EE web authentication and recovery schemes to glean lessons that can be applied when evaluating E2EE recovery processes.
- Discuss key trends in the E2EE authentication landscape and provide concrete research directions based on observed gaps between industry deployment and academic literature.

2 Related Work

2.1 Authentication Frameworks

In 2012 Bonneau et al. [63] introduced a seminal framework for evaluating authentication schemes across security, deployability, and usability, concluding at the time that no scheme met as many desired criteria as conventional password-based authentication. While the desirable properties of any authentication mechanism deployed at scale have generally remained the same, both state-of-the-art industry deployments and academic research understanding have evolved dramatically since their survey—device-based credential protocols such as FIDO2 did not yet exist, and industry had only just begun introducing the concept of multi-factor authentication (MFA) [157]. In 2017, Alomar et al. [29] presented a framework for classifying social authentication schemes and associated attacks, though in practice few of the social authentication schemes are deployed outside of trusted contact recovery. In 2021, Kunke et al. [229] evaluated 12 common account recovery mechanisms within the 2012 framework of Bonneau et al., though at the time the only FIDO2 protocol deployments were hardware token-based and thus they did not consider passwordless authentication.

Prior comparative surveys of authentication schemes were either conducted well before major contemporary shifts in authentication schemes (namely, MFA adoption, device-based authentication, and E2EE authentication) or focused on authentication schemes predating recent trends (as in Lassak et al. [233], a longitudinal study of the usability of email, SMS, recovery questions, and social authentication as fallback mechanisms). To the best of our knowledge no prior work has surveyed authentication and recovery schemes in an end-to-end encrypted context.

2.2 Measuring Industry Deployments

A handful of prior studies have surveyed industry deployments of non-E2EE authentication schemes. In a 2021 survey of the 208 most

widely used websites that offer account creation, Gavazzi et al. [140] found that only 42.3% of accounts support MFA and approximately 22% appear to support some form of risk-based authentication (e.g., blocking suspicious login attempts based on geolocation). In 2019, Ulqinaku et al. [383] found that 23 of the Alexa top 100 websites support the Universal 2nd Factor (U2F) hardware token protocol, but did not find any sites supporting passwordless authentication at the time. More recently, in 2023 Kuchhal et al. [228] surveyed the prevalence of the Web Authentication (WebAuthn) API used to provide public-key authentication (and deployed as part of various MFA schemes and passwordless authentication), finding that while 85 of the 585 domains in the Tranco Top 1K [320] that supported account creation also supported the WebAuthn protocol, the vast majority supported MFA, not passwordless authentication.

In this work, we particularly investigate E2EE web services, where the account that the user is attempting to recover is encrypted such that the provider definitionally can be of no help in recovering the encrypted data, and furthermore the user may no longer have access to their password or other primary authentication mechanism (e.g., mobile device). Holtervennhoff et al. [176] recently conducted a usability survey of users' perceptions and strategies for handling E2EE recovery keys, but no prior work has looked comprehensively at deployed authentication and recovery mechanisms for end-to-end encrypted data, where either the web service or credentials may be E2EE.

3 End-to-End Encryption

We begin by briefly defining end-to-end encryption (E2EE) and summarizing current E2EE deployments.

3.1 Overview

In the client-server architecture pattern, user data frequently passes through third-party servers, such as a messaging application's web server. The core principle of E2EE is that encrypted data can be decrypted only by the ends of the communication (i.e., client devices). E2EE requires that any data is encrypted prior to leaving the end-user's client device using decryption keys that never leave the client. Critically, the service provider is unable to decrypt the data under any circumstances, even under threat of legal mandate, as the decryption keys are stored locally. The goal of E2EE is to combat insider threats, including provider employees, the compromise of provider infrastructure by a malicious third party, or government warrants requesting data access [389].

Traditionally mentioned in the context of secure ephemeral messaging (e.g., WhatsApp), E2EE can be deployed for many scenarios where a user wants to preserve the confidentiality of their data against the service provider storing the data (though the provider will still retain some amount of metadata). Most widely used password managers have deployed E2EE for well over a decade [252], and conference calling platforms such as Zoom and Microsoft Teams began rolling out E2EE in 2020 and 2021 respectively [187, 354], albeit only for premium users in the case of Microsoft Teams [283] and not enabled by default in either Zoom or Teams [187, 283].

Increasingly, E2EE is being deployed to secure long-term data storage through cloud-based services in addition to ephemeral communications. We also include an in-depth discussion of E2EE cloud storage as this is a comparatively recent development within industry and these accounts tend to be of high value for end users.

3.1.1 E2EE Credentials. There are several types of credentials involved in building an E2EE system, the distinctions between which are important when discussing authentication and recovery:

(1) **User-facing account login credentials**: Most authentication today is still done using passwords, passcodes, and other forms of knowledge-based authentication where the user provides a human-memorable string as evidence of their authorization. To the end-user, the day-to-day authentication process in E2EE systems is similar (and often even identical) to non-E2EE systems. The difference between E2EE and non-E2EE authentication lies in the lack of conventional account recovery mechanisms to which users have become accustomed (namely, password reset in case of a forgotten or lost credential). Given that account recovery is inherently an uncommon circumstance, the helplessness of the provider is a distinction users sometimes fail to appreciate until it is too late (discussed in more depth in Section 5).
(2) **Cryptographically-generated account login credentials**: An increasingly common paradigm is to authenticate the user using something they possess, rather than something they know, which translates to cryptographic credentials invisible to the user. Biometric authentication and all forms of device-bound credentials fall under this category, meaning that should the user lose the device or ability to biometrically authenticate, they have no recourse to recover access if that was their only authentication scheme. A key focus of this work is discussing mitigation strategies around device-bound credentials in particular.
(3) **E2EE protocol credentials**: E2EE protocols involve numerous public/private key pairs used to enable secure key exchange and encrypt and decrypt content, including long-term identity key pairs, short-term session keys, and pre-shared keys. In this work, we abstract these details to focus only on the long-term private keys no longer stored with the service provider.

3.2 Move Towards E2EE Storage

Historically, cloud service providers have opted not to encrypt account data storage end-to-end due to political pressure [16, 17] and concerns over usability and account lock-out [188]. Apple in particular has come under pressure in the past from the U.S. government over encrypted cloud storage specifically: around 2017, Apple abandoned internal plans to encrypt cloud storage backups after the FBI objected [197]. From a usability standpoint, academic studies on the usability of multi-factor authentication apps have found account lockout to be a repeated concern for users since app backups are generally encrypted [146, 334]. The risk that users may lose long-term data is likely part of the reason that E2EE storage has been slow to catch on even among providers who deployed E2EE for messaging several years prior. WhatsApp, for instance, rolled out E2EE communications in 2016, but did not offer E2EE backups of these communications until 2021 [281].

In the wake of a changing technical and political landscape, however, Apple has publicly advocated for protecting cloud data

with end-to-end encryption [188] and rolled out an opt-in E2EE backup scheme in late 2022 (see Figure 4 in the Appendix). When it was revealed in February 2025 that the U.K. government had served Apple with a legal notice to provide backdoor access to encrypted iCloud storage, Apple responded by removing the feature altogether rather than comply (and so U.K. users can no longer benefit from comprehensive E2EE backups) [199]. As of March 2025, Apple is locked in a legal battle with the U.K. government over its refusal to provide access to encrypted backups [327] and has received public expressions of support from lawmakers and national security officials in the U.S. government concerned about the security implications of weakening encrypted systems [198].

Apple's public justification for making E2EE backup opt-in for users, rather than the default, is to reduce the risk of permanent data loss as "the feature requires the user to take ultimate responsibility for managing their cryptographic keys" [188]. While Apple's deployment represents a major step forward for cloud data privacy, encrypted data storage for the general public requires us to revisit what security-usability trade-offs are acceptable since a provider by definition cannot restore access. A well-designed user interface can warn the user of the consequences if they lose their recovery credentials, but this could also serve to deter users from opting in. While Apple has declined to report statistics on what percentage of users opted to enable E2EE cloud backups, it is likely to be minimal due in part to concerns over recovery. To encourage widespread adoption of E2EE storage, we need schemes which are suitable for the average user.

Encrypted data storage is similar to non-custodial wallets in the cryptocurrency ecosystem, a space notorious for tales of irreversible data loss. Non-custodial wallets require the user to control their own private keys, instead of having the keys managed by their service provider [115]. The corollary to this setup is that the user has no recourse if they lose their private key, leading to permanent loss of the currency stored in the wallet. Cryptocurrency firms have been grappling with this problem for years, though the problem setup is slightly different in that cryptocurrency storage is often accessed infrequently, while messaging and storage applications can be accessed multiple times per day (which can affect which schemes are considered viable for a general userbase).

3.3 E2EE Storage Deployments

Apple iCloud, Facebook Messenger, and WhatsApp have all deployed opt-in end-to-end encrypted backup services in the past two years. In 2022, Apple's Advanced Data Protection scheme provides users with the option to encrypt their iCloud backups end-to-end, such that the encryption keys are replicated across all user devices. Previously, it had only been possible to encrypt most iCloud data such that Apple retained access. While several other services already offered E2EE storage, these services are largely targeted at a more tech-savvy userbase and have not seen mass adoption.

Specific protocols vary, but cloud storage providers generally achieve E2EE backup by storing decryption keys in hardware security modules (HSMs) such that the user authenticates to the HSM and the provider relays messages between the client device and HSM but has no access itself. The HSM is essential to guard the key material against physical attacks and to enforce rate-limiting.

4 Protecting Authentication Credentials End-to-End

Contemporary authentication credentials can be assigned to one of two high-level categories: (1) credentials bound to the hardware of a single device and (2) credentials that can be synced across multiple devices. Since device-bound credentials pose significant usability drawbacks (elaborated below in §4.1), the most common paradigm is to store syncable, multi-device credentials in a credential manager (historically referred to as *password managers*, though this is increasingly a misnomer as they are used to store both conventional passwords and cryptographic keys). Credential managers, henceforth *CMs*, are databases of authentication credentials that typically allow synchronization across multiple devices. These are in turn protected by either a master password or external hardware (e.g., YubiKey), reducing the cognitive load for the end-user as they will need to remember at most one password that unlocks all other authentication credentials.

All widely used CMs (see Table 1 for a full list of systems considered based on related work and recent rankings of password manager services [221]) depend on E2EE to prevent any party other than the originating user accessing the user's credentials [378]. All data is encrypted on the client device before being synced to the server, typically using keys derived from the master password used to unlock the CM using a key derivation function such as PBKDF2.

In this section, we present several important security distinctions between device-bound and syncable credentials, and discuss variations of CM designs which impact the security of syncable credentials. We conclude with an in-depth discussion of *passkeys*, an emerging passwordless authentication scheme that comes in both credential flavors (device-bound and syncable) and illustrates the security and usability benefits and drawbacks of current architectures.

4.1 Device-Bound Credentials

With the increasing prevalence of hardware tokens and secure hardware chips in smartphones, authentication using only a single strong hardware-backed factor (the user's smartphone) is now viable—allowing providers to remove passwords from daily authentication flows. Most modern devices contain built-in platform authenticators (e.g. Trusted Platform Module [TPM] in Windows, Secure Enclave in macOS/iOS, and StrongBox in Android).

At a high level, device-bound credentials use public-key cryptography to authenticate, where the smartphone's secure hardware component generates a unique keypair for each web account, stores the private key in the hardware module, and shares the public key with the web server. Importantly, once generated the private key *cannot* be exported from device hardware, and hence it is not possible to backup, sync, or otherwise duplicate the credential. To authenticate to the web service, a user need only authenticate to their local device authenticator using their regular device unlock mechanism (e.g., fingerprint, PIN, pattern).

Most common forms of device-bound authentication are based on the FIDO2 specifications [129] which comprise the WebAuthn standard [419] for client-to-server communication and the client-to-authenticator protocol (CTAP) [125]. FIDO2 was intended from the start as a contender for password replacement [132]. The initial

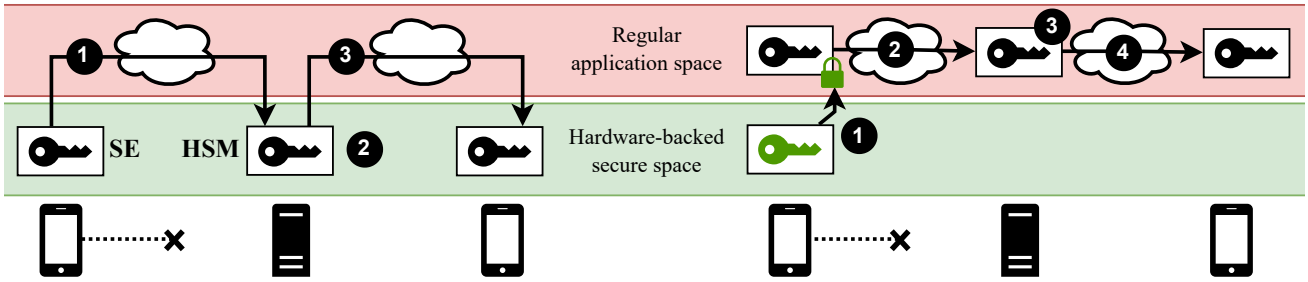

Figure 1: Synchronization flows for credentials. *Left:* in ideal first-party syncing, the credentials always live in secure hardware in either a secure enclaves (SE) or hardware security modules (HSM); neither backup (1) nor recovery (3) expose the key to regular application space; the HSM (2) enforces end-to-end authentication and rate-limits. *Right:* third-party syncing more commonly accesses the key material in regular application space; however, it might be encrypted-at-rest (1) using a key inside the SE; it is typically protected in-transit (2, 4) and uses password-derived keys for encryption-at-rest to ensure E2EE (3).

deployed version of the Fast Identity Online (FIDO) protocol in 2019 ("FIDO U2F") relied on two-factor authentication (2FA) hardware tokens (Section A.2.4), and the passwordless standard was released in 2018 [129]. In FIDO terminology, this means there are two types of hardware authenticators a user can use to authenticate: using a *roaming authenticator*, such as a discrete hardware token, or a *platform authenticator*, such as the built-in smartphone authenticator [132].

The primary benefit provided by the FIDO authentication scheme is mitigating phishing attacks and large-scale compromise, since each account has its own distinct keypair. During login, the device signs a challenge to prove possession of the respective private key (see Figure 2a). Binding the private key to the original web service ensures that authenticators do not sign challenges coming from malicious websites. Additionally, the graphical interfaces of CMs provide no mechanism to view the underlying cryptographic key, preventing users from sharing the key with an adversary [4].

As a result of these enhanced security properties, single-device credentials are often mandated in enterprise use cases or desirable for highly security-conscious individuals who want to ensure maximal resistance to phishing and other common attacks. The primary downside from a security perspective is that credential security now reduces to device security [132], as a compromised device can allow an adversary access to all services. Smartphone providers have deployed several security measures to prevent unauthorized access, such as requiring biometric authentication each time a passkey is used and instituting device unlock rate-limiting, but the precise security protections will depend on platform and user configurations.

Device-bound credentials suffer from serious usability drawbacks: they inherently mean that a user can only authenticate to a service with the particular device (e.g., laptop computer) on which the credential is stored. If the device is lost or damaged, the credential is lost and a user may be locked out of their account permanently. This can be mitigated through the use of multiple independent hardware tokens configured to authenticate to the same account, but a user must go out of their way to set this up and may still lose both hardware devices needed to access the account. Creating and storing credentials in a secure hardware device without any option to export the key material is the gold standard from a security perspective, but it falls short from a usability standpoint due to an inherent lack of recoverability or backup options. We discuss recovery schemes in greater detail in Section 5.

4.2 E2EE-Synced Credentials

To address the tension between security and usability in device-bound credentials, industry has developed a variety of syncable credential schemes in which credentials can be shared with either a separate client device or a cloud service.

Such synchronization can take one of two forms, each with distinct security properties: credentials are backed up and later recovered to a new device; or credentials are exchanged between two known devices. We further separate CMs into two sub-categories: first-party CMs, where the CM is integrated with the specific device OS (e.g., iCloud Keychain and iOS), and third-party CMs, which are designed to be hosted by any OS (e.g., LastPass).

4.2.1 Cloud backups vs. routine device syncing.

Backups and recovery: Storing duplicate copies of credential vaults in the cloud is an essential fail safe for device loss. We discuss E2EE recovery options in greater detail in Section 5, but cloud-based recovery is generally viewed by providers as an exceptional case and providers have varying processes for regaining access ranging from long-term recovery codes to social authentication. Since this scenario assumes loss of any existing, known devices (and sometimes also the loss of the master password), the cloud backup provider authenticates a login attempt from a new client device based only on the specified device-agnostic recovery mechanism, with no direct key agreement between the client and server.

Syncing between known devices: Syncing across active client devices (e.g., propagating an authentication credential newly created on a mobile device to a desktop) is both simpler from a usability standpoint and more secure as it allows for interactive key agreement and authorization that is largely invisible to the user. If a user owns a sufficiently large number of "living devices", such as multiple laptops or a tablet device, each device effectively functions as a backup copy of the credential vault, making it less likely the user would lose *all* devices and need to initiate a recovery process.

The previous backup and recovery process using a cloud-based HSM can be understood as a two-degree exchange between known

devices. In contrast, in this setting the HSM effectively operates as a trusted and attested user device which is being operated remotely. The initial backup step and the subsequent recovery step are therefore just exchanges between trusted devices and the remaining complexity addresses the challenge of remote attestation and authorization.

4.2.2 First-Party vs. Third-Party Credential Syncing.
First-Party Credential Syncing: To enhance authentication usability and security, major industry OS providers have implemented their own in-house CMs (e.g., iCloud Keychain). We refer to this as *first-party credential syncing*, or intra-ecosystem syncing, where the cloud backup service and CM vendors are the same. Operating systems with a built-in CM generally sync credentials to their respective cloud backup service by default [41, 152] (iCloud, for instance, automatically adds credentials to the iOS Keychain provided the third-party service developer has specified the credential is syncable [42]).

Importantly from a security standpoint, Apple's iCloud Keychain syncs credentials from one client device enclave to another client device enclave via Apple-owned HSM clusters such that even the encrypted credentials never leave hardware storage during backup or recovery (as shown in the left diagram in Figure 1) [39]. Google's Android devices handle public key credentials, e.g. passkeys, similarly, relying on cloud-based HSM devices as well [152]. Because these vendors own the device hardware, device OS (including low-level interfaces), and syncing servers, they are able to design an architecture with significantly greater security guarantees than that of third-party CMs which need to account for every type of hardware and OS [116, 295]. While non-vendor-specific CMs may use secure hardware where available, first-party CMs are *guaranteed* to deploy hardware-backed syncing end-to-end.

Third-Party Credential Syncing: The majority of CMs are what we term *third-party* CMs, such as LastPass [239] and BitWarden [54], which can generally be used as either a native application hosted locally or as a browser extension on most major operating systems and web browsers. One of the most attractive features of these CMs is the ability for credentials to be synced and exported across different ecosystems (i.e., inter-ecosystem syncing).

This flexibility comes with security downsides: in order to be exportable, the credentials must necessarily enter the application process at some point, leaving them vulnerable to memory extraction attacks [77, 185, 295] and other offline and online attacks on password managers based on adversarial possession of an encrypted database [74, 147, 302], including brute-force attacks to compute the backup encryption key and decrypt the database [50, 97, 107]. These exports (and corresponding imports on another client) happen frequently in most CMs, with services typically synchronizing state after each new credential is added [117]. Of seven widely used third-party CMs analyzed, only one (Keeper) mentions HSMs as part of their server architecture [212], with the others storing the encrypted database vaults directly on their servers [5, 54, 97, 114, 239, 299]. Several CMs provide only high-level details of their server security architecture publicly, so it is difficult to be certain of their design.

4.2.3 Attacks on E2EE Cloud Credential Storage. While E2EE is essential for providing a threshold level of security for credential vaults synced to the cloud, these services still present several security risks not relevant for device-bound credentials. Syncing across multiple devices inherently increases the attack surface, leaving credentials vulnerable to the compromise of the weakest synced device [353]. Credential cloud sync is critical for satisfying end-user expectations of recoverability, but cloud sync means that the practical security of synced credentials reduces to the security of the cloud account, including cloud provider security [91, 132, 252, 353].

Perhaps most importantly, while the credentials themselves are encrypted in E2EE systems, credential vault metadata is generally unencrypted [175]. The precise categories of metadata stored in plaintext vary by service, but E2EE cloud services have repeatedly been shown to be vulnerable to metadata and other injection attacks exploiting file de-duplication as well as other storage features [117, 175]. LastPass's 2022 compromise was particularly concerning because of the large amount of unencrypted metadata [235, 418], prompting LastPass to begin encrypting more metadata such as URLs [88].

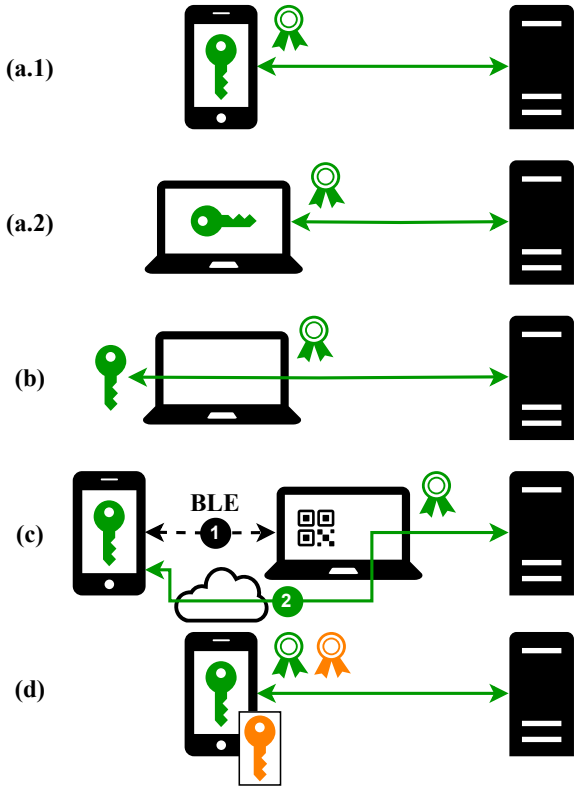

Figure 2: Passkey authentication flows: (a.1) standard authentication using a passkey on a mobile phone and (a.2) computer; (b) authentication with a roaming authenticator (NFC, BLE, or USB); (c) hybrid cross-device authentication initiated via a QR code; (d) authentication with an additional credential using the *device-bound public key extension*.

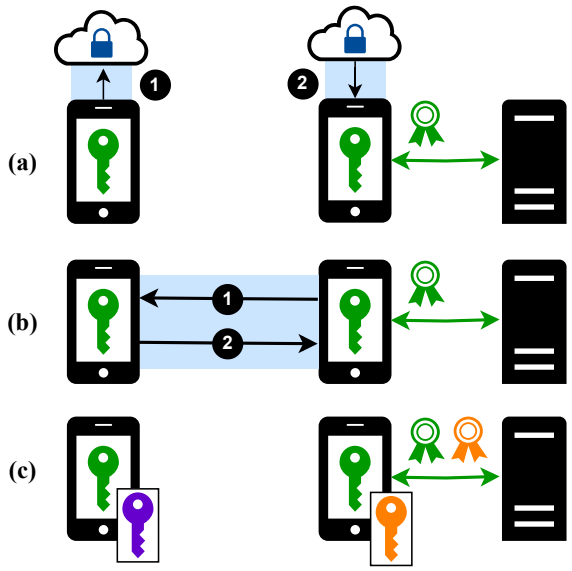

Figure 3: Passkey synchronization: (a) synchronization via the cloud (either E2EE between enclaves or using a third-party app); (b) importing into a new device using CXP; (c) in either approach, device-bound keys are not synchronized.

4.3 Passkeys

Major industry vendors (namely, Apple and Google) refer to the FIDO2 passwordless authentication mechanism as *passkeys*. [132] Google first introduced passkey support for Google Accounts in May 2023 [87] and Apple followed suit in June [409], though both will continue to support passwords as an authentication mechanism for the foreseeable future. On Apple devices, credentials such as passkeys are part of the keychain information that can be backup to iCloud using Apple's in-house E2EE iCloud Keychain [40]. Passkey syncing to iCloud Keychain occurs by default but users can disable this synchronization for individual devices. On Android devices, credentials such as passkeys are managed by the optionally E2EE Google Password Manager which provides E2EE backups that can be restored on new devices.

All major operating systems and web browsers now support passwordless authentication [132], including cross-device authentication (e.g. using a phone with a computer that has no local authenticator) and cross-device sync using a platform-agnostic password manager such as 1Password [8].

4.3.1 Authentication. During registration and sign-in flows a *relying party* (e.g., a webservice) interacts with the device, the *client*, on which the user wishes to authenticate. This is mediated via the WebAuthn protocol [419] such that the relying party generally does not have to be concerned with the authenticator implementation and characteristics. The client-to-authenticator protocol (CTAP) [125] then enables communication between the client and the authenticator. We illustrate common authentication flows in Figure 2. The simplest authentication variant uses a *platform authenticator* where the client device itself has direct access to an internal authenticator, e.g., an iOS device storing passkeys in its Secure Enclave (Figure 2a). Alternatively, the client might use a discrete hardware device, a *roaming authenticator*, connected via NFC, BLE, or USB (Figure 2b).

4.3.2 Cross-Device Authentication. The latest FIDO2 standard draft allows cross-ecosystem *authentication* (CDA), where a passkey on one device can be used to authenticate to a service from a different device (Figure 2c). We consider a typical first-time CDA flow where the user tries to login to a website on a laptop using an existing passkey on their smartphone [126, §11.5.1]. When receiving the WebAuthn request, the client, i.e. the web browser, displays a QR code which is scanned with the smartphone. The smartphone then shares BLE announcements to prove proximity and contribute to the shared secret state between itself and the client. Both now establish a connection via a tunnel service through which the authentication request is forwarded from the client to the authenticator. The client and authenticator may remember this link and subsequent visits can initiate a connection based on the state and without scanning a QR code [126, §11.5.2].

4.3.3 Backup and recovery. Most platforms support first-party backup of passkeys which is typically limited to devices of the same ecosystem (see §4.2.2). Alternatively, the user might manage their passkeys using third-party CMs. In the first-party scenario, passkeys are typically only processed in secure hardware whereas it is common for third-party applications to perform operations on the decrypted credential in the regular application space (Figure 3). The latter setup can expose them while in-use to malware running on the device even if they are encrypted-at-rest with help of a secure element.

The authenticator decides during creation of a passkey whether it allow backups and/or synchronization between devices. For each credential, the authenticator attests to the relying party whether the credential is *backup-eligible*. Credentials which are backup eligible are commonly referred to as a *multi-device key*. Authenticators also report whether the credential is currently backed up, which relying parties can make use of to improve the user experience. For instance, a website might inform the user that their credential is also available on other devices; or it might warn them that their credentials are currently not backed up and that they should set up additional recovery mechanisms.

Since the backup eligibility is chosen by the authenticator, the relying party often cannot determine whether the same device is used across two authentication attempts. Where device binding is important, the relying party can use the *device-bound public key extension* [419, §10.2.2] to ask the authenticator to create an additional public key with the credential (see Figure 2d). The additional public key is never synced or backed up (see Figure 3c) allowing the relying party to identify new devices.

4.3.4 Exchange between known devices. In addition to backups, which allow for future recovery with a fresh device, users may wish to share a passkey between two known devices. While this can be achieved with existing backup-and-recover flows, an interactive protocol between the devices allows for a simpler and more secure exchange. The Credential Exchange Protocol (CXP) [127] describes how an importing device can send an export request to another device which then allows it to wrap the credential using a Hybrid Public Key Encryption (HPKE) scheme [31] and sends it over to

the new device. CXP's flexible authorization design allows to make exchange depend on approval from other parties. For instance, a company can use it to ensure that its employees can forward certain passkeys only to other company-owned devices.

4.3.5 Usability. Academic studies of end-user perceptions of passwordless authentication revealed inaccurate mental models (e.g. believing the fingerprint or other biometric data is sent to the web server [232]) and challenges with inconsistencies in user interface design [306, 308], though passwordless authentication was generally considered simple to set up and use [425]. Users repeatedly express concerns over account loss [261, 425], though cloud sync recovery was not an option at the time the studies were conducted (and so users were presented with a design in which total device loss meant total account loss). After reviewing usability studies of passwordless authentication and current FIDO2 deployments, we identify three residual concerns hampering passkey adoption: sharing, revocation and availability.

Credential Sharing: Credential sharing is a common practice across numerous scenarios, including home and work environments [211, 254, 360, 406, 423, 425]. The latest industry deployments have made passkey sharing simpler with inter-ecosystem syncing, but there are still several cases where existing syncing mechanisms are inadequate [51]. One of the contexts in which credential sharing is most prevalent, the workplace, is particularly challenging given the recent trend towards remote work such that users may not be in physical proximity to each other, making passwords far simpler to share over common written communication channels. While there are a handful of vendor-specific solutions (for instance, passkeys can now be AirDropped via iCloud Keychain in iOS [43]) these are niche and not relevant for many use cases.

Credential Revocation: The FIDO2 standard does not adequately address credential revocation at a global scale. Currently, it assumes web services will implement the ability for a user to revoke access (e.g., remove the public key from the server) for specific authenticators, which a user will need to do for each individual website for which they have registered an authenticator [261, 425].

Credential Use Cases: That passkeys can be bound to the trusted hardware of specific devices (sometimes by default) is a major advantage from a security standpoint but can pose a significant disadvantage for widespread usability and availability, particularly for at-risk demographics. Shared devices (including public computers [289]) are common, with users sharing devices for financial, cultural, and personal reasons [22, 60, 66, 205, 223, 314], including temporary sharing (e.g., showing a friend or relative a photo slideshow [205, 274]). Conversely, this also raises privacy concerns over inadvertent account sharing depending on passkey access duration before requiring reauthentication in a particular implementation.

4.3.6 Passkey deployment and availability. Here, we quantify the deployment of passwordless authentication in practice since the FIDO2 standard was published in 2019 [129]. To measure passkey deployment, we manually inspect each site in the top 300 of the Alexa Top 1M dataset of most widely visited domains [203]. For sites that offer multiple account versions (e.g. a free version and a paid version), we follow the methodology of Gavazzi et al. [140] and select what we expect to be the most common account type. We observed that several sites do not offer passkey generation at the time of account creation, forcing a user to register with a standard username/password combination if not using single sign-on (See Appendix A.1.1), but do allow a user to create a passkey as an additional authentication mechanism when logging in a second time. All experiments were conducted by logging in from Chrome 124.0.6367.203 on macOS Sonoma 14.2.1 in November 2024 from Cambridge, U.K.

Of the top 300 domains, we successfully audited $n = 206$; the remaining sites either did not offer account creation, required service-specific information to set up the account such as a phone number with a particular country code or banking credentials, or did not load. Since 44 of these sites are Google country-specific domains (e.g. *google.ch*) which use the same Google account we eliminate these from our dataset leaving us with $n = 162$ domains. Of these 162 sites, we find 17 sites (10.5%) offer direct passkey support for user authentication. We observe that a further 70 sites (43.2%) do not directly support passkeys but offer single-sign on with a provider which *does* support passkeys (in the vast majority of cases, Google). If we include sites which offer indirect passkey support through SSO, 87 of 162, or 53.7%, of sites in the top 300 directly or indirectly offer passkey support, a significant increase from a 2021 study which found no support for passwordless authentication among 235 popular sites [140].

4.3.7 Passkeys and E2EE systems. Passkeys are not a feasible authentication or recovery mechanism for E2EE systems—the scheme authenticates against a third-party and does not provide key material locally. In fact, it is the other way around: passkeys rely on strong E2EE systems to become usable through cross-device synchronization and backup mechanisms. However, we believe that passkeys hold important lessons for developing and evaluating usable E2EE systems. Researchers, developers, and designers should take a close look at the already identified usability problems that emerge when keys are not directly accessible, but can only be handled within the constraints of existing standards. Unlike a password or recovery code, cryptographic keys are inherently intangible and in case of secure hardware not even accessible to users. Therefore, establishing correct and helpful mental models is essential. In particular, exchanging keys between two known, online devices is much simpler than restoring a backup to a new device at a later time without access to the original device. As a general lesson, passkeys demonstrate that stronger security at reasonable usability cost is often easier to achieve within one homogeneous ecosystem. Requiring inter-ecosystem interoperability can lead to lower overall security where secure hardware requires proprietary access and first-party authorization.

5 Keys Not Under Doormats: Recovery in E2EE Systems

While E2EE represents a major improvement in the security level offered by credential managers and other E2EE web services, its use raises natural follow-up questions around recovery and usability. E2EE comes at a price: if only the user has all the information needed to access the data, the service provider is unable to come to the rescue should a user forget their password or lose their client device. Even loss of access to an E2EE password manager is a

scenario the user can often indirectly recover from through manual service-by-service email- or SMS-based password resets.

The prospect of E2EE for cloud backup services such as Apple iCloud make security and recoverability trade-offs more salient. Users store troves of important photos, documents, and other valuable long-term data in cloud services. Account loss with a cloud service provider can be devastating: stories of users who had their accounts shut down after being unfairly flagged by Google as uploading inappropriate material [206, 208] vividly illustrate the practical consequences of sudden account loss, with one such user describing that it "felt as if her house had burned down" [207].

5.1 E2EE Recovery Mechanisms

Here, we outline and discuss deployed recovery mechanisms in E2EE schemes, systematizing currently deployed protocols across E2EE services. For the purposes of this section we do not distinguish between authentication and recovery schemes since our goal is to summarize pathways towards account access. From an attacker's perspective, there is no distinction between authentication and recovery. Table 1 shows the diverse array of authentication recovery mechanisms used by the 22 most widely used E2EE providers for storage, email, messaging, and CMs. Our discussion focuses on services designed for individual consumer use rather than services targeted at enterprises (Table 1 presents authentication schemes available in the free version of each service and notes schemes available only in premium [non-business] service versions).

If a user has access to a logged-in client device, recovery is simple. A common recovery mechanism invisible to the user is to automatically save a decryption key to the browser or device's local keychain where it can be accessed upon device unlock, enabling the user's device to serve as an authentication mechanism. WhatsApp, for instance, allows a user to reset their recovery code through the WhatsApp app after authenticating to their device using biometrics or entering the device PIN (which allow the WhatsApp client to access the encryption stored on device) [412]. Apple's Advanced Data Protection E2EE cloud backup scheme offers a similar recovery protocol [36], and Messenger's Labyrinth protocol allows users to send a one-time code to their old device [280].

The challenging scenario is the case where a user has lost access to both the password used to unlock the account in question and, where applicable, all relevant client devices. For instance, perhaps someone has lost their phone, and attempts to restore WhatsApp on a new phone, only to discover they cannot find the decryption key to the WhatsApp backup.

5.1.1 Recovery Codes. We find that recovery codes are the primary backup method used to recover access to encrypted data, with 17 of the 22 providers surveyed offering or mandating this backup method. We described subtle variations in recovery code deployments in Appendix A.3.2, but these codes are generally arbitrary alphanumeric strings of 24 to 64 digits intended to be non-human-memorable.

Unfortunately, it is all too easy for users to make mistakes that can lead to accidental account lockout, such as taking a screenshot of the recovery code which is then synced to the cloud storage to which the code restores access [176]. Some services (e.g., Keeper [213]) also make setting up a recovery code optional. Users will have the option of enabling the feature at account setup but can proceed without it. An additional consideration is that non-enterprise cloud services are often used by small businesses and community organizations in addition to personal usage, and access credentials may have high turnover in ways that can make it easier for legitimate users to forget or lose access to credentials.

Recent academic research has suggested that recovery code loss is not uncommon in practice: Holtervennhoff et al. [176] investigated how users perceive and store recovery codes in E2EE services, conducting a user survey of 281 users of Tutanota, an E2EE email service, and qualitatively analyzing Reddit support threads for the same service. They found several support threads in which users had lost their account password, 2FA device, and recovery code, and the user survey revealed a small number of users reported that they had not recorded the recovery code at all, believing there was no chance they would lose their passwords. Approximately 12% of users surveyed believed Tutanota could help them regain access in case of recovery code loss, and only 14.8% of users saved the recovery code in more than one location. These results are derived from the userbase of a privacy-centered email service, which the authors acknowledge "is not representative of email users or privacy-conscious users in general" [176], so we may anticipate user misconceptions to be even higher in a service targeted at a mass audience.

5.1.2 Human-memorable PINs and passcodes. To explore more usable solutions, some providers allow users to enter a shorter or more memorable recovery code, such as a seed phrase, a short PIN, or a user-chosen password. To avoid brute-force attacks, these low-entropy codes are then used to generate a secure encryption key using a key derivation function. Meta's Labyrinth design, for instance, allows users to enter a 4-digit PIN and then uses this short PIN to authenticate to a longer, internal pseudorandom recovery code stored in a rate-limited HSM, limiting the number of attempts to 10. Google Password Manager similarly requires users to set a 6-digit PIN to recover access on a new device [152].

If a hardware exploit is able to overcome an HSM provider's rate limits, however, a short string like a PIN can be brute-forced in as little as a few hours [98, 259, 267]. To address this attack vector, Dauterman et al. [98] proposed distributing the decryption keys among multiple HSMs, and in 2024 Signal deployed a key recovery system that distributes trust among multiple types of HSMs from different vendors since an exploit is unlikely to compromise *all* HSM types [91]. Similarly, Juicebox [297], an open-source key recovery protocol created by Signal cofounder Moxie Marlinspike among others in 2023, enables PIN-based recovery using multiple independent cloud HSM providers but is still in the early stages.

Even with rate-limiting, however, PINs may still be easily guessed as users are also liable to choosing easy-to-guess PINs or PINs based on easily discoverable dates such as a birthday [266]. PINs are also potentially compromised via social attacks such as shoulder-surfing [194] and thus rate limiting mitigates but does not prevent even external attackers from compromising a numeric PIN.

Short, numeric PINs are also not a foolproof strategy to counter the fallibility of human memory as some percentage of users will still invariably lose access to the PIN. In one study of Signal PINs,

		Authentication and Recovery Mechanism								
	E2EE Platform	Device Keychain	User-Chosen Password	Recovery Code	Third-Party Storage	PIN	Recovery File	Recovery Email	Recovery Contact	Recovery Group
Storage	Apple iCloud	●	●	●	○	○	○	○	●	○
	NordLocker	○	●	●	○	○	○	○	○	○
	pCloud	○	●	●	○	○	○	○	○	○
	MEGA	○	●	●	○	○	○	○	○	○
	Tresorit	○	●	○	○	○	○	○	○	○
	Internxt	○	●	●	○	○	○	○	○	○
	Filen	○	●	●	○	○	○	○	○	○
Email	Proton	●	●	●	○	○	●	○	○	○
	PreVeil	●	○	○	○	○	●	○	○	●
	Tutanota	○	●	●	○	○	○	○	○	○
	StartMail	○	●	●	○	○	○	●	○	○
Messaging	FB Messenger	●	●	●	●	●	○	○	○	○
	WhatsApp	●	●	●	●	○	○	○	○	○
	Signal	●	○	○	○	○	○	○	○	○
Credential Managers	LastPass	●	●	●	○	○	○	○	●	○
	Bitwarden	●	●	●	○	○	○	○	●†	○
	Dashlane	●	●	●	○	○	○	○	○	○
	1Password	●	●	●	○	○	●	○	●	○
	NordPass	●	●	●	○	○	●	○	○	○
	Keeper	●	●	●	○	○	○	○	○	○
	Enpass	●	●	○	○	○	○	○	○	○
	Google PM	●	○	○	○	●	○	○	○	○

Table 1: Deployed recovery mechanisms for end-to-end encrypted storage, email, and messaging services. The table displays all recovery options offered by each service, but in practice some of these choices may be mutually exclusive. For instance, WhatsApp allows a user to set either a recovery code or a user-chosen recovery password, but not both. In this context, 'device keychain' scheme is defined as authenticating to the client application using the client device's unlock mechanism. We define and discuss social authentication (recovery contact and recovery group) in depth in § A.3.1. †Only available for premium users.

12% of participants reported "occasionally, frequently, or very frequently" forgetting their PIN [47]. WhatsApp's E2EE backup scheme offers the option of a user-generated password to authenticate to the pseudorandom recovery code instead of directly storing the recovery code, where the low-entropy password is used to generate the proper 64-digit key using an oblivious pseudorandom function. From the user's perspective, this may be simpler to use as they only need to enter the shorter or more human-memorable sequence. Further variations include a "recovery phrase", a long string of between 12 and 24 words that the user can either store or attempt to memorize (commonly used for cryptocurrency wallets and sometimes referred to as a "brain wallet" [76, 115]). Proton, an end-to-end encrypted email service, uses a 12-word recovery phrase instead of a more conventional pseudorandom recovery code [324]. The length of recovery phrases, though, makes memorizing these phrases an unattractive option for the general public, with the net result that recovery phrases offer little to no benefit compared with a standard pseudorandom string.

5.1.3 Manual Reset. As a final form of fallback, we note that StartMail, an E2EE email service, offers users the ability to request an email reset [361]. To preserve E2EE, on the backend their scheme requires the keys of two separate staff members to jointly recover the user's lost decryption key. While this scheme is not properly E2EE in the sense that colluding employees can access user keys, it nonetheless presents an interesting case study.

We further observed that every CM offering a business or enterprise version offer the ability for an administrator of the business account to manually reset a user's master password even when all other recovery options have been lost [6, 53, 96, 113, 189, 236, 238, 298], though there is some variation in whether this feature is enabled by default or if a business needs to explicitly opt in. This is effectively a variation of trusted contact authentication where the trusted contact is another employee within the organization.

5.2 Takeaways from E2EE Recovery

Having observed that recovery codes are the primary backup method used to recover access to encrypted data, we discuss potential design modifications to improve end-user recovery.

5.2.1 Usability Improvements. There are additional feature choices providers can offer to mitigate the risk that a user loses or forgets their recovery passcode in the first place. WhatsApp and Signal both provide regular password reminders asking users to confirm their backup PIN, though in Signal's case a user optionally has to enter a short PIN while WhatsApp requires users to enter either their password or full 64-digit code before they can access the app [393]. Other providers may offer a similar feature as well, though it is not explicitly stated in the documentation. One study of Signal's opt-in PIN reminder, however, found that roughly quarter of participants in one survey reported that they rarely or never confirmed their PIN when prompted [47]. While mandatory reminders may be more effective, they also run the risk of becoming a nuisance to users, potentially leading users to disable E2EE storage to avoid these notifications and achieving the opposite of the desired effect.

We also observe patterns of insecure defaults among E2EE recovery. Namely, some services (e.g. Keeper [213]) make setting up a recovery code optional. Users will have the option of enabling the feature at account setup but can proceed without it, a risky architectural design from a usability standpoint as the user now relies only on their master password.

Facebook Messenger's E2EE protocol, Labyrinth [280]), represents an interesting case study of trade-offs between authentication and data recovery. Since Messenger is commonly used as a web application, Labyrinth is designed to allow users to log in on a new device or browser using only their ordinary Facebook credentials. If a user no longer has access to their cryptographic key material (namely, their private authentication key), they will still be able to log in to their account, but will not have access to their conversation history. This is a reasonable trade-off for E2EE messaging, but does not work for E2EE of long-term storage, where the whole purpose is to access previously generated data.

Cross-Provider Syncing: A promising feature from a usability standpoint is the ability to automatically store the recovery key in a separate cloud service. Meta's Labyrinth protocol gives users the option to store their pseudorandom recovery code in either Google Drive or iCloud Drive (depending on the mobile platform), where it is stored in a hidden folder in the third-party cloud service and does not preclude the user from separately storing the recovery code elsewhere as well. This form of automated storage represents an improvement from a usability standpoint in that a user would now have to lose access to both cloud providers, but there are privacy considerations: in a study of 2FA backups, Gilsenan et al. [146] found that automatically uploading backups to Google Drive requires the user to grant read access to the additional service for their Google account name, email address, and photo.

Proton provides a slight twist on automatic storage by offering a platform-agnostic encrypted "recovery file" that can be stored long-term and provided to Proton at a later date to restore access [324]. The documentation cryptically suggests that both recovery phrases and/or files "may become outdated", at which point a user will be warned that this recovery mechanism is no longer valid to restore access but will have to generate a new recovery phrase (though it is unclear why or how often this might occur).

In general, automatic cross-provider cloud storage represents a real usability improvement over asking the user to store the key on separate physical hardware (e.g. a USB stick) or to write it down on a piece of paper somewhere as both are easy to lose accidentally. In principle, distributing trust among two distinct providers (such that the recovery key for one provider is stored with a separate provider) is the same as the the widely referenced assumption in cryptography of non-colluding servers [392, 403]. Both providers would need to be compromised or collude for data confidentiality to be compromised, and a user would need to lose authentication credentials to both services to become locked out.

6 Key Findings

We identify four key findings (KFs) from our review of the security, usability, and privacy properties of the impact of E2EE on contemporary web authentication and recovery.

KF1: *Syncable passkeys provide lower security guarantees than device-bound passkeys.* Not all passkeys are created equal: the FIDO Alliance uses the term "passkey" to refer to any passwordless FIDO credential [130, 131], but there are important security distinctions [295]. While device-bound credentials by definition never leave the hardware enclave in which they are created, E2EE syncable credentials potentially travel over multiple cloud providers to other client devices. The ability to sync credentials at all is essential for usability [131] and made possible by E2EE, but E2EE is not a panacea: passkeys are vulnerable to all the same issues impacting E2EE cloud storage and credential managers in the past, including metadata, brute-force attacks, and other client-side malware [117, 175, 418]. This is especially true if the user credential vault is protected by a weak password (widespread weak master password use was one of the reasons the 2022 LastPass server compromise was concerning [226, 235]). Only iCloud Keychain syncing provides a similar security level as device-bound credentials as discussed in §4.2.2.

Even so, syncable passkeys provide significant security improvements over even a randomly-generated password. Credential manager GUIs are generally designed such that the passkey's cryptographic key pair cannot even be viewed by the user even if they wanted to provide enhanced phishing-resistance [7]. Passkeys further prevent users from using weak passwords or reusing passwords since the credential is auto-generated for them.

While biometric authentication schemes may provide improved security from a purely technical standpoint, there are important legal precedents in the U.S. governing when law enforcement can compel a user to unlock a device: while passwords and other forms of knowledge-based authentication are generally protected by the Fourth and Fifth Amendments (i.e. law enforcement cannot require an individual to provide their device passcode to guard against unreasonable search and seizure), multiple district-level courts in the U.S. have ruled that law enforcement can forcibly compel fingerprint authentication [196]. A user for whom preventing law enforcement access is the most important aspect of their personal threat model may intentionally avoid biometric authentication even though it provides greater protection against a generic external adversary.

KF2: *We find minimal consensus around recovery schemes in end-to-end encrypted systems.* While a majority (19) of the 22 E2EE service providers studied either used an arbitrary, alphanumeric recovery code as their primary recovery failsafe or did not provide

any recovery mechanism in addition to the master password, beyond recovery codes there is little to no consensus around whether providers *should* offer additional recovery mechanisms and if so, what those mechanisms should be. To improve usability, six of the 22 providers offered the ability to automatically save the recovery code in either third-party storage (e.g. integrated with Google Drive) or as a local recovery file containing the code. Two providers allowed short PINs to be used as a recovery mechanism, while five providers offered some form of social authentication.

Inconsistencies across which authentication and recovery possibilities are provided in the first place is equally as important for accurate user mental models and understanding of recovery possibilities. Prior work has shown that variations in 2FA backup recovery processes make it more likely a user may misconfigure backups [33, 142]. To enable passkey adoption on a large scale, there is a need for industry to standardize available recovery options across E2EE web services, particularly for providers that serve a user base comprised of the general public (e.g., Apple).

KF3: *To prevent lockout, E2EE contexts require a wider set of authentication schemes than general web authentication.* Services should offer a variety of recovery methods to cover the diversity of users' personal situations, and offer the ability to enable more than one recovery path for sufficient redundancy. Given that E2EE inherently lacks provider-assisted recovery as a fallback option, we observe clear differences between authentication schemes deployed in E2EE services compared with schemes deployed in general use cases. In particular, while none of the top-300 websites currently offer recovery via trusted contacts, five of 22 E2EE services offer some form of social authentication recovery.

Above all, the expectation that users will handle retaining their own decryption keys (including recovery keys), a PGP-era approach still the predominant strategy in use today, may be a fair assumption with the risk assessment of E2EE messaging backups but does not pass muster with general cloud storage or authentication credential backups. Failure to provide users with a greater diversity of recovery options (which users have expressed a desire for [176]) may make non-technical users reluctant to opt in to advanced security schemes. Moreover, for most users the prospect of account lockout is itself a security threat, where loss of access to important files, credentials, or other data will have a significant impact on real-world security and well-being.

The net result will be that E2EE services (for data of high importance to users, like cloud storage) are largely confined to a smaller subset of dedicated users, ultimately limiting the security benefits for the general public. Cloud service providers may be reluctant to offer certain recovery options out of concern over increasing the attack surface. But E2EE cloud storage with more usable recovery options is still more secure than non-E2EE cloud storage, where a user is *guaranteed* that at least one third-party, the service provider itself, can access their account data.

KF4: *Distributing trust across multiple providers can mitigate the risks of E2EE account authentication and recovery.* HSM rate-limited PIN authentication is among the most usable of authentication and recovery schemes since it only asks the user to remember a much shorter string. Its low entropy makes it difficult to recommend adoption as a recovery scheme at the moment as a hardware exploit would easily compromise E2EE data, but there are currently multiple promising cryptographic proposals or open-source protocols to divide the recovery process among separate vendor cloud services and/or vendor HSMs [91, 297]. This is one of the most promising avenues for usable E2EE recovery for the general public.

7 Limitations

Literature Survey: Our literature review dataset is thorough but necessarily non-exhaustive, and it is possible our methodology may have missed a small number of papers. For the purposes of this work, our goal is to systematically capture overall trends in academic authentication research, which are reflected in our findings.

Understanding E2EE Passkey Synchronization: Our understanding and discussion of the security and usability of E2EE synchronization of passkeys relies on the publicly available documentation provided by the vendors. Hence, it does not allow us to verify that the deployed system behaves as described and the published documents naturally cannot cover all details. Where necessary we made conservative assumptions about the provided functionalities drawing on knowledge from similar protocols and implementations.

8 Future Research

We find four key areas relevant to E2EE authentication and recovery that are not adequately examined in the current literature:

FR1: *User perceptions and understanding of E2EE recovery.* Prior work in E2EE recovery has shown some users request recovery schemes which are common in everyday web authentication but incompatible with E2EE (e.g., manual reset, security questions, or email/SMS recovery) [176]. Users have also demonstrated poor comprehension of the distinction between account recovery (e.g., the ability to log in) and recovering storage content (e.g., email history, calendar) [176]. To date there has been just one academic study of E2EE recovery [176] and more user interface design research is needed to confirm existing findings and to better understand how to improve user understanding around E2EE recovery. In particular, future work should consider investigating the relative importance users assign to different E2EE services (i.e. how users perceive loss of ephemeral messaging history compared with credential manager loss and loss of cloud storage) and the impact on types of recovery schemes a provider may want to offer.

FR2: *Longitudinal recovery studies:* Account recovery mechanisms are, by definition, more likely to be needed as more time has passed. In 2015, Bonneau et al. [62] found a linear relationship between the time passed since account creation and the proportion of authentication reset requests. Despite this, most usability studies conduct a cross-sectional examination of recovery schemes at a single point in time. A recent longitudinal study from Lassak et al. [233] found that for common non-E2EE recovery schemes (email, SMS, recovery questions, and social authentication) the relative convenience of recovery is consistent over time, but similar studies are critical for E2EE services, where the most common E2EE recovery scheme is to require users to maintain a non-human-memorable recovery code. A survey of E2EE email recovery support threads [176] found strong evidence suggesting users are likely to misplace their recovery code

over time, motivating further work quantifying this possibility and deriving best practices. Evaluating an optimal balance between mandatory and opt-in periodic backup code confirmation (across a spectrum of backup codes from short PINs to arbitrary pseudorandom recovery codes) is another critical area of future research.

FR3: *Comparative analysis of different social authentication schemes:* Despite being deployed as one of Apple's E2EE storage recovery choices, trusted contact recovery has received comparatively little attention from the academic research community as discussed in Section B.1. The need for a greater focus here is particularly critical in light of recent developments in generative AI, leaving end-users highly vulnerable to impersonation scams and account compromise, a reality made all the more dangerous given that deployed authentication schemes are increasingly centralized within a single account.

There are several variations in deployed trusted contact authentication schemes in E2EE services. Some schemes rely on a single trusted contact, some allow users to specify multiple individual trusted contacts, and another (PreVeil) has deployed group recovery where a threshold number of contacts must participate. Providers also sometimes also enforce time delays after trusted contact authentication has taken place. To date there have not been any studies investigating user *preferences* within these variations to understand what the general public would find most usable.

FR4: *Improve security of E2EE credential syncing:* A broad category of future work is to improve the security properties provided by E2EE credential managers. This includes metadata-hiding protocols for E2EE credential stores and cloud storage more generally, further analysis of backend third-party CM architectures, and improving interoperability of cross-vendor secure enclaves to enhance end-device security.

9 Conclusion

In this work, we systematize E2EE authentication and recovery mechanisms, identifying numerous unresolved challenges and open areas of research. The dual trends of E2EE credential management and E2EE sevices are deeply interconnected and thus we survey them together to arrive at a complete picture of contemporary web authentication and recovery. Given that 53.7% of top 300 domains now support authentication via passkeys either directly or via single sign-on, and the vast majority of passkey implementations require E2EE credential backups, E2EE recovery is a critical area of future research. Each of the various E2EE recovery mechanisms currently offered by cloud services exist somewhere on a spectrum of prioritizing security to prioritizing recoverability. Prior work has suggested the possibility of resolving E2EE-recoverability tradeoffs using biometric authentication in the distant future (in a scenario where the user needs to provide only biometric data to recover account access) [301, 443], but the perennial challenge with biometric and social authentication is keeping pace with evermore-sophisticated scams that convincingly fake some aspect of human interaction (e.g. voice, live video) [71, 247, 371]. Most importantly, E2EE providers should offer users a greater choice of recovery options to reflect the diversity of users' situations and perceived account value now that E2EE is increasingly targeted at the general public.

Acknowledgments

Jenny Blessing is funded by Entrust and Daniel Hugenroth is supported by Nokia Bell Labs. Ross Anderson made important contributions to the ideas contained in this paper. Unfortunately he died on 28th March 2024 before the final version was written; any errors remain our own.

References

[1] 2016. Towards Improving the Memorability of System-assigned Random Passwords. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. USENIX Association, Denver, CO. https://www.usenix.org/conference/soups2016/workshop-program/way2016/presentation/al-ameen

[2] 2024. Have I Been Pwned. https://haveibeenpwned.com/. Last accessed 30th May 2024.

[3] 2024. Mozilla Monitor. https://monitor.mozilla.org/. Last accessed 24th Nov 2024.

[4] 1Password. 2023. Viewing passkeys. https://1password.community/discussion/142844/viewing-passkeys. October 2023.

[5] 1Password. 2024. 1Password Security Design. https://1passwordstatic.com/files/security/1password-white-paper.pdf. June 25, 2024.

[6] 1Password. 2024. Best practices for securing your 1Password Business account. https://support.1password.com/business-security-practices/. Last accessed November 29th, 2024.

[7] 1Password. 2024. How can I see my actual passkey? https://1password.community/discussion/148371/how-can-i-see-my-actual-passkey. September 20th, 2024.

[8] 1Password. September 2023. Now available: Save and sign in with passkeys using 1Password in the browser and on iOS. https://blog.1password.com/save-use-passkeys-web-ios/.

[9] Jacob Abbott, Daniel Calarco, and L Jean Camp. 2018. Factors influencing password reuse: A case study. TPRC.

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

[11] Yomna Abdelrahman, Mohamed Khamis, Stefan Schneegass, and Florian Alt. 2017. Stay cool! understanding thermal attacks on mobile-based user authentication. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3751–3763.

[12] Yasmeen Abdrabou, Yomna Abdelrahman, Mohamed Khamis, and Florian Alt. 2021. Think harder! Investigating the effect of password strength on cognitive load during password creation. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–7.

[13] Yasmeen Abdrabou, Johannes Schütte, Ahmed Shams, Ken Pfeuffer, Daniel Buschek, Mohamed Khamis, and Florian Alt. 2022. " Your Eyes Tell You Have Used This Password Before": Identifying Password Reuse from Gaze and Keystroke Dynamics. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[14] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. 2021. Hear "no evil", see "kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems. In *2021 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 712–729.

[15] Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. 2021. Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems. In *2021 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 730–747.

[16] Hal Abelson, Ross Anderson, Steven M Bellovin, Josh Benaloh, Matt Blaze, Jon Callas, Whitfield Diffie, Susan Landau, Peter G Neumann, Ronald L Rivest, et al. 2021. Bugs in our pockets: The risks of client-side scanning. *arXiv preprint arXiv:2110.07450* (2021).

[17] Harold Abelson, Ross Anderson, Steven M Bellovin, Josh Benaloh, Matt Blaze, Whitfield "Whit" Diffie, John Gilmore, Matthew Green, Susan Landau, Peter G Neumann, et al. 2015. Keys under doormats. *Commun. ACM* 58, 10 (2015), 24–26.

[18] AccessNow. 2017. *PUBLIC SECURITY ALERT: New Facebook attack – watch out for phishy messages that say you're a "Trusted Contact".*

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

[20] Alaadin Addas, Amirali Salehi-Abari, and Julie Thorpe. 2019. Geographical security questions for fallback authentication. In *2019 17th international conference on privacy, security and trust (PST)*. IEEE, 1–6.

[21] Age Verification Providers Association. 2024. US State age verification laws for adult content. https://avpassociation.com/4271-2/. Last accessed November 20th, 2024.

[22] Syed Ishtiaque Ahmed, Md Romael Haque, Irtaza Haider, Jay Chen, and Nicola Dell. 2019. "Everyone Has Some Personal Stuff" Designing to Support Digital Privacy with Shared Mobile Phone Use in Bangladesh. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[23] Mahdi Nasrullah Al-Ameen, Kanis Fatema, Matthew Wright, and Shannon Scielzo. 2015. The Impact of Cues and User Interaction on the Memorability of System-AssignedRecognition-Based Graphical Passwords. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. 185–196.

[24] Mahdi Nasrullah Al-Ameen, Matthew Wright, and Shannon Scielzo. 2015. Towards making random passwords memorable: Leveraging users' cognitive ability through multiple cues. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2315–2324.

[25] Fatma Al Maqbali and Chris J Mitchell. 2018. Email-based password recovery-risking or rescuing users?. In *2018 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 1–5.

[26] Yusuf Albayram and Mohammad Maifi Hasan Khan. 2015. Evaluating the effectiveness of using hints for autobiographical authentication: A field study. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. 211–224.

[27] Yusuf Albayram and Mohammad Maifi Hasan Khan. 2016. Evaluating smartphone-based dynamic security questions for fallback authentication: a field study. *Human-Centric Computing and Information Sciences* 6 (2016), 1–35.

[28] Reem AlHusain and Ali Alkhalifah. 2021. Evaluating fallback authentication research: A systematic literature review. *Computers & Security* 111 (2021), 102487.

[29] Noura Alomar, Mansour Alsaleh, and Abdulrahman Alarifi. 2017. Social authentication applications, attacks, defense strategies and future research directions: a systematic review. *IEEE Communications Surveys & Tutorials* 19, 2 (2017), 1080–1111.

[30] Suood Alroomi and Frank Li. 2023. Measuring Website Password Creation Policies At Scale. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 3108–3122.

[31] Joël Alwen, Bruno Blanchet, Eduard Hauck, Eike Kiltz, Benjamin Lipp, and Doreen Riepel. 2021. Analysing the HPKE standard. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 87–116.

[32] Sabrina Amft, Sandra Höltervennhoff, Nicolas Huaman, Yasemin Acar, and Sascha Fahl. 2023. "Would You Give the Same Priority to the Bank and a Game? I Do Not!" Exploring Credential Management Strategies and Obstacles during Password Manager Setup. In *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. 171–190.

[33] Sabrina Amft, Sandra Höltervennhoff, Nicolas Huaman, Alexander Krause, Lucy Simko, Yasemin Acar, and Sascha Fahl. 2023. "We've Disabled MFA for You": An Evaluation of the Security and Usability of Multi-Factor Authentication Recovery Deployments. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 3138–3152.

[34] Nampoina Andriamilanto, Tristan Allard, Gaëtan Le Guelvouit, and Alexandre Garel. 2021. A large-scale empirical analysis of browser fingerprints properties for web authentication. *ACM Transactions on the Web (TWEB)* 16, 1 (2021), 1–62.

[35] Apple. 2023. *Help a friend or family member as their account recovery contact*. https://support.apple.com/en-us/102608.

[36] Apple. 2023. *If you can't remember the password for your encrypted backup*.

[37] Apple. 2023. *Set up a recovery key for your Apple ID*.

[38] Apple. 2024. How to use account recovery when you can't reset your Apple ID password. https://support.apple.com/en-us/118574. Last accessed 30th May 2024.

[39] Apple. 2024. iCloud Keychain security overview. https://support.apple.com/en-gb/guide/security/sec1c89c6f3b/1/web/1. Last accessed 28th November 2024.

[40] Apple. 2024. Passkeys Overview. https://developer.apple.com/passkeys/. Last accessed 28th May 2024.

[41] Apple. 2024. Secure iCloud Keychain recovery. https://support.apple.com/en-gb/guide/security/secdeb202947/1/web/1. Last accessed 19th June 2024.

[42] Apple. 2024. Secure keychain syncing. https://support.apple.com/en-gb/guide/security/sec0a319b35f/1/web/1. Last accessed 28th November 2024.

[43] Apple. 2024. Share passkeys and passwords securely with AirDrop on iPhone. https://support.apple.com/en-gb/guide/iphone/iph0dd1796bb/ios. Last accessed 29th November 2024.

[44] Zhongjie Ba, Sixu Piao, Xinwen Fu, Dimitrios Koutsonikolas, Aziz Mohaisen, and Kui Ren. 2018. ABC: Enabling smartphone authentication with built-in camera. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018*.

[45] Mihai Bâce, Alia Saad, Mohamed Khamis, Stefan Schneegass, and Andreas Bulling. 2022. PrivacyScout: Assessing vulnerability to shoulder surfing on mobile devices. *Proceedings on Privacy Enhancing Technologies* (2022).

[46] Guangdong Bai, Jike Lei, Guozhu Meng, Sai Sathyanarayan Venkatraman, Prateek Saxena, Jun Sun, Yang Liu, and Jin Song Dong. 2013. Authscan: Automatic extraction of web authentication protocols from implementations. (2013).

[47] Daniel V Bailey, Philipp Markert, and Adam J Aviv. 2021. "I have no idea what they're trying to accomplish:" Enthusiastic and Casual Signal Users' Understanding of Signal PINs. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. 417–436.

[48] David G Balash, Xiaoyuan Wu, Miles Grant, Irwin Reyes, and Adam J Aviv. 2022. Security and privacy perceptions of Third-Party application access for google accounts. In *31st USENIX Security Symposium (USENIX Security 22)*. 3397–3414.

[49] Ingolf Becker, Simon Parkin, and M Angela Sasse. 2018. The rewards and costs of stronger passwords in a university: linking password lifetime to strength. In *27th USENIX Security Symposium (USENIX Security 18)*. 239–253.

[50] Andrey Belenko and Dmitry Sklyarov. 2012. "Secure Password Managers" and "Military-Grade Encryption" on Smartphones: Oh, Really? *Blackhat Europe* (2012), 56.

[51] Kemal Bicakci and Yusuf Uzunay. 2022. Is FIDO2 passwordless authentication a hype or for real?: A position paper. In *2022 15th International Conference on Information Security and Cryptography (ISCTURKEY)*. IEEE, 68–73.

[52] Bitkey. 2022. *Losing your keys without losing your coins*.

[53] Bitwarden. 2024. Account Recovery. https://bitwarden.com/help/account-recovery/. Last accessed November 29th, 2024.

[54] Bitwarden. 2024. Bitwarden Security Whitepaper. https://bitwarden.com/help/bitwarden-security-white-paper/. Last accessed 29th November 2024.

[55] Sam Blackshear, Konstantinos Chalkias, Panagiotis Chatzigiannis, Riyaz Faizullabhoy, Irakliy Khaburzaniya, Eleftherios Kokoris Kogias, Joshua Lind, David Wong, and Tim Zakian. 2021. Reactive key-loss protection in blockchains. In *Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers 25*. Springer, 431–450.

[56] Jeremiah Blocki, Benjamin Harsha, and Samson Zhou. 2018. On the economics of offline password cracking. In *2018 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 853–871.

[57] Jeremiah Blocki, Saranga Komanduri, Lorrie Cranor, and Anupam Datta. 2014. Spaced repetition and mnemonics enable recall of multiple strong passwords. *arXiv preprint arXiv:1410.1490* (2014).

[58] Jeremiah Blocki and Peiyuan Liu. 2023. Towards a rigorous statistical analysis of empirical password datasets. In *2023 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 606–625.

[59] Jeremiah Blocki and Wuwei Zhang. 2022. DALock: Password Distribution-Aware Throttling. *Proceedings on Privacy Enhancing Technologies* (2022).

[60] Joshua Blumenstock and Nathan Eagle. 2010. Mobile divides: gender, socioeconomic status, and mobile phone use in Rwanda. In *Proceedings of the 4th ACM/IEEE international conference on information and communication technologies and development*. 1–10.

[61] Joseph Bonneau. 2012. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 538–552.

[62] Joseph Bonneau, Elie Bursztein, Ilan Caron, Rob Jackson, and Mike Williamson. 2015. Secrets, lies, and account recovery: Lessons from the use of personal knowledge questions at google. In *Proceedings of the 24th international conference on world wide web*. 141–150.

[63] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 553–567.

[64] Joseph Bonneau and Sören Preibusch. 2010. The Password Thicket: Technical and Market Failures in Human Authentication on the Web.. In *WEIS*.

[65] John Brainard, Ari Juels, Ronald L Rivest, Michael Szydlo, and Moti Yung. 2006. Fourth-factor authentication: somebody you know. In *Proceedings of the 13th ACM conference on Computer and communications security*. 168–178.

[66] Jenna Burrell. 2010. Evaluating Shared Access: social equality and the circulation of mobile phones in rural Uganda. *Journal of computer-mediated communication* 15, 2 (2010), 230–250.

[67] Daniel Buschek, Fabian Hartmann, Emanuel Von Zezschwitz, Alexander De Luca, and Florian Alt. 2016. Snapapp: Reducing authentication overhead with a time-constrained fast unlock option. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3736–3747.

[68] Andre Büttner, Andreas Thue Pedersen, Stephan Wiefling, Nils Gruschka, and Luigi Lo Iacono. 2023. Is It Really You Who Forgot the Password? When Account Recovery Meets Risk-Based Authentication. In *International Conference on Ubiquitous Security*. Springer, 401–419.

[69] Michele Campobasso and Luca Allodi. 2020. Impersonation-as-a-service: Characterizing the emerging criminal infrastructure for user impersonation at scale. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1665–1680.

[70] Claude Castelluccia, Markus Dürmuth, and Daniele Perito. 2012. Adaptive password-strength meters from markov models.. In *Proceedings of the Symposium on Network and Distributed System Security*.

[71] Charles Bethea). 2023. The Terrifying A.I. Scam That Uses Your Loved One's Voice. https://www.newyorker.com/science/annals-of-artificial-intelligence/the-terrifying-ai-scam-that-uses-your-loved-ones-voice. March 2024.

[72] Melissa Chase, Hannah Davis, Esha Ghosh, and Kim Laine. 2022. Acsesor: A new framework for auditable custodial secret storage and recovery. *Cryptology ePrint Archive* (2022).

[73] Rahul Chatterjee, Anish Athayle, Devdatta Akhawe, Ari Juels, and Thomas Ristenpart. 2016. pASSWORD tYPOS and how to correct them securely. In *2016 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 799–818.

[74] Rahul Chatterjee, Joseph Bonneau, Ari Juels, and Thomas Ristenpart. 2015. Cracking-resistant password vaults using natural language encoders. In *2015 IEEE Symposium on Security and Privacy*. IEEE, 481–498.

[75] Rahul Chatterjee, Joanne Woodage, Yuval Pnueli, Anusha Chowdhury, and Thomas Ristenpart. 2017. The typtop system: Personalized typo-tolerant password checking. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 329–346.

[76] Panagiotis Chatzigiannis, Konstantinos Chalkias, Aniket Kate, Easwar Vivek Mangipudi, Mohsen Minaei, and Mainack Mondal. 2023. SoK: Web3 Recovery Mechanisms. *Cryptology ePrint Archive* (2023).

[77] Efstratios Chatzoglou, Vyron Kampourakis, Zisis Tsiatsikas, Georgios Karopoulos, and Georgios Kambourakis. 2024. Keep your memory dump shut: Unveiling data leaks in password managers. In *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 61–75.

[78] Guangke Chen, Sen Chenb, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. 2021. Who is real bob? adversarial attacks on speaker recognition systems. In *2021 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 694–711.

[79] Yuxin Chen, Zhuolin Yang, Ruben Abbou, Pedro Lopes, Ben Y Zhao, and Haitao Zheng. 2021. User authentication via electrical muscle stimulation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[80] Yu Chen, Yang Yu, and Lidong Zhai. 2023. InfinityGauntlet: Expose Smartphone Fingerprint Authentication to Brute-force Attack. In *32nd USENIX Security Symposium (USENIX Security 23)*. 2027–2041.

[81] Eunyong Cheon, Jun Ho Huh, and Ian Oakley. 2023. GestureMeter: Design and Evaluation of a Gesture Password Strength Meter. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[82] Eunyong Cheon, Yonghwan Shin, Jun Ho Huh, Hyoungshick Kim, and Ian Oakley. 2020. Gesture authentication for smartphones: Evaluation of gesture password selection policies. In *2020 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 249–267.

[83] Ivan Cherapau, Ildar Muslukhov, Nalin Asanka, and Konstantin Beznosov. 2015. On the Impact of Touch ID on iPhone Passcodes. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. 257–276.

[84] Eugene Cho, Jinyoung Kim, and S Shyam Sundar. 2020. Will you log into tinder using your facebook account? adoption of single sign-on for privacy-sensitive apps. In *Extended abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–7.

[85] Geumhwan Cho, Jun Ho Huh, Junsung Cho, Seongyeol Oh, Youngbae Song, and Hyoungshick Kim. 2017. Syspal: System-guided pattern locks for android. In *2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 338–356.

[86] Soumyadeb Chowdhury, Ron Poet, and Lewis Mackenzie. 2014. Passhint: Memorable and secure authentication. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2917–2926.

[87] Christiaan Brand and Sriram Karra. May 2023. The beginning of the end of the password. https://blog.google/technology/safety-security/the-beginning-of-the-end-of-the-password/. (May 2023).

[88] Christofer Hoff. 2024. LastPass is Encrypting URLs. Here's What's Happening. https://blog.lastpass.com/posts/lastpass-is-encrypting-urls-heres-whats-happening. May 22, 2024.

[89] Stéphane Ciolino, Simon Parkin, and Paul Dunphy. 2019. Of two minds about Two-Factor: Understanding everyday FIDOU2F usability through device comparison and experience sampling. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 339–356.

[90] Jessica Colnago, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Lorrie Cranor, and Nicolas Christin. 2018. "It's not actually that horrible" Exploring Adoption of Two-Factor Authentication at a University. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–11.

[91] Graeme Connell, Vivian Fang, Rolfe Schmidt, Emma Dauterman, and Raluca Ada Popa. 2024. Secret Key Recovery in a Global-Scale End-to-End Encryption System. *Cryptology ePrint Archive* (2024).

[92] Sourav Kumar Dandapat, Swadhin Pradhan, Bivas Mitra, Romit Roy Choudhury, and Niloy Ganguly. 2015. Activpass: your daily activity is your password. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2325–2334.

[93] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The tangled web of password reuse.. In *Proceedings of the Symposium on Network and Distributed System Security*, Vol. 14. 23–26.

[94] Sanchari Das, Andrew Dingman, and L Jean Camp. 2018. Why Johnny doesn't use two factor a two-phase usability study of the FIDO U2F security key. In *Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26–March 2, 2018, Revised Selected Papers 22*. Springer, 160–179.

[95] Sanchari Das, Bingxing Wang, Andrew Kim, and L Jean Camp. 2020. MFA is A Necessary Chore!: Exploring User Mental Models of Multi-Factor Authentication Technologies.. In *HICSS*. 1–10.

[96] Dashlane. 2024. Admin-assisted recovery for members of professional plans. https://support.dashlane.com/hc/en-us/articles/115005111905-Admin-assisted-recovery-for-members-of-professional-plans. Last accessed November 29th, 2024.

[97] Dashlane. 2024. Dashlane's Security Principles & Architecture. https://www.dashlane.com/download/whitepaper-en.pdf. June 10, 2024.

[98] Emma Dauterman, Henry Corrigan-Gibbs, and David Mazières. 2020. SafetyPin: Encrypted backups with Human-Memorable secrets. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. 1121–1138.

[99] Xavier de Carné de Carnavalet and Mohammad Mannan. 2014. From very weak to very strong: Analyzing password-strength meters. In *Network and Distributed System Security Symposium (NDSS 2014)*. Internet Society.

[100] Emiliano De Cristofaro, Honglu Du, Julien Freudiger, and Greg Norcie. 2013. A comparative usability study of two-factor authentication. *arXiv preprint arXiv:1309.5344* (2013).

[101] Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. 2012. Touch Me Once and I Know it's you! Implicit Authentication Based on Touch Screen Patterns. In *proceedings of the CHI Conference on Human Factors in Computing Systems*. 987–996.

[102] Alexander De Luca, Alina Hang, Emanuel Von Zezschwitz, and Heinrich Hussmann. 2015. I feel like I'm taking selfies all day! Towards understanding biometric authentication on smartphones. In *Proceedings of the 33rd annual ACM Conference on Human Factors in Computing Systems*. 1411–1414.

[103] Alexander De Luca, Marian Harbach, Emanuel von Zezschwitz, Max-Emanuel Maurer, Bernhard Ewald Slawik, Heinrich Hussmann, and Matthew Smith. 2014. Now you see me, now you don't: protecting smartphone authentication from shoulder surfers. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2937–2946.

[104] Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. 2016. An Empirical Study of Textual Key-Fingerprint Representations. In *25th USENIX Security Symposium (USENIX Security 16)*. 193–208.

[105] Michael Dietz and Dan S Wallach. 2014. Hardening Persona-Improving Federated Web Login.. In *Proceedings of the Symposium on Network and Distributed System Security*.

[106] Yana Dimova, Tom Van Goethem, and Wouter Joosen. 2023. Everybody's Looking for SSOmething: A large-scale evaluation on the privacy of OAuth authentication on the web. *Proceedings on Privacy Enhancing Technologies* (2023).

[107] Dominik Schurmann. 2023. Why 2FA was useless with LastPass. https://www.heylogin.com/en/post/lastpass-2fa-useless. March 2, 2023.

[108] Paul Dunphy, Vasilis Vlachokyriakos, Anja Thieme, James Nicholson, John McCarthy, and Patrick Olivier. 2015. Social Media As a Resource for Understanding Security Experiences: A Qualitative Analysis of # Password Tweets. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. 141–150.

[109] Antonin Durey, Pierre Laperdrix, Walter Rudametkin, and Romain Rouvoy. 2021. FP-Redemption: Studying browser fingerprinting adoption for the sake of web security. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 18th International Conference, DIMVA 2021, Virtual Event, July 14–16, 2021, Proceedings 18*. Springer, 237–257.

[110] Jonathan Dutson, Danny Allen, Dennis Eggett, and Kent Seamons. 2019. 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)*. IEEE, 119–128.

[111] Ed Felten. 2024. How Yahoo could have protected Palin's email. https://freedom-to-tinker.com/2008/09/21/how-yahoo-could-have-protected-palins-email/. September 2008.

[112] Serge Egelman, Andreas Sotirakopoulos, Ildar Muslukhov, Konstantin Beznosov, and Cormac Herley. 2013. Does my password go up to eleven? The impact of password meters on password selection. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2379–2388.

[113] Enpass. 2024. Access Recovery in Enpass Business. https://support.enpass.io/business/console/hub/access_recovery_in_enpass_business.htm. Last accessed November 29th, 2024.

[114] Enpass. 2024. Enpass Security Whitepaper. https://support.enpass.io/docs/security-whitepaper-enpass/index.html. Last accessed November 30th, 2024.

[115] Yimika Erinle, Yathin Kethepalli, Yebo Feng, and Jiahua Xu. 2023. SoK: Design, Vulnerabilities and Defense of Cryptocurrency Wallets. *arXiv preprint arXiv:2307.12874* (2023).

[116] evmbrahmin.eth. 2024. Passkeys in Apple's iCloud Keychain. https://evmbrahmin.com/blog/iCloud-keychain-passkeys-guide.html. Last accessed November 30th, 2024.

[117] Andrés Fábrega, Armin Namavari, Rachit Agarwal, Ben Nassi, and Thomas Ristenpart. 2024. Exploiting Leakage in Password Managers via Injection Attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*. 4337–4354.

[118] Facebook. 2016. *Messenger Secret Conversations: Technical Whitepaper*.

[119] Facebook. 2023. Choose friends to help you log in if you ever get locked out of your account. https://m.facebook.com/help/119897751441086/?wtsid=rdr_0t7GXNGeR7wZU8ogO. (2023).

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

[121] Shehroze Farooqi, Fareed Zaffar, Nektarios Leontiadis, and Zubair Shafiq. 2017. Measuring and mitigating oauth access token abuse by collusion networks. In *Proceedings of the 2017 Internet Measurement Conference*. 355–368.

[122] Haonan Feng, Hui Li, Xuesong Pan, Ziming Zhao, and T Cactilab. 2021. A Formal Analysis of the FIDO UAF Protocol.. In *Proceedings of the Symposium on Network and Distributed System Security*.

[123] Hossein Fereidooni, Jan König, Phillip Rieger, Marco Chilese, Bora Gökbakan, Moritz Finke, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. 2023. AuthentiSense: A Scalable Behavioral Biometrics Authentication Scheme using Few-Shot Learning for Mobile Platforms. *arXiv preprint arXiv:2302.02740* (2023).

[124] Daniel Fett, Ralf Küsters, and Guido Schmitz. 2015. Spresso: A secure, privacy-respecting single sign-on system for the web. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1358–1369.

[125] FIDO Alliance. 2021. Client to Authenticator Protocol (CTAP). https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-20210615.html. Last accessed 25th June 2024.

[126] FIDO Alliance. 2021. Client to Authenticator Protocol (CTAP). http://fidoalliance.org/specs/fido-v2.2-rd-20230321/fido-client-to-authenticator-protocol-v2.2-rd-20230321.html. Last accessed 28th November 2024.

[127] FIDO Alliance. 2024. Credential Exchange Protocol. https://fidoalliance.org/specs/cx/cxp-v1.0-wd-20241003.html. Last accessed 28th November 2024.

[128] FIDO Alliance. 2024. FIDO Universal 2nd Factor (U2F) Overview. https://fidoalliance.org/specs/u2f-specs-master/fido-u2f-overview.html. Last accessed 28th May 2024.

[129] FIDO Alliance. 2024. FIDO2. https://fidoalliance.org/fido2/. Last accessed 28th May 2024.

[130] FIDO Alliance. 2024. Passkeys. https://fidoalliance.org/passkeys/#faq. Last accessed November 30th, 2024.

[131] FIDO Alliance. 2024. White Paper: Synced Passkey Deployment: Emerging Practices for Consumer Use Cases. https://fidoalliance.org/white-paper-synced-passkey-deployment-emerging-practices-for-consumer-use-cases/. January 2024.

[132] FIDO Alliance. March 2022. *How FIDO Addresses a Full Range of Use Cases*.

[133] Dinei Florencio and Cormac Herley. 2007. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*. 657–666.

[134] Dinei Florêncio, Cormac Herley, and Baris Coskun. 2007. Do strong web passwords accomplish anything? *HotSec* 7, 6 (2007), 159.

[135] Dinei Florêncio, Cormac Herley, and Paul C Van Oorschot. 2014. Password Portfolios and the Finite-Effort User: Sustainably Managing Large Numbers of Accounts. In *23rd USENIX Security Symposium (USENIX Security 14)*. 575–590.

[136] Schubert Foo, Siu Cheung Hui, Peng Chor Leong, and Shigong Liu. 2000. An integrated help desk support for customer services over the World Wide Web—a case study. *Computers in Industry* 41, 2 (2000), 129–145.

[137] Tore Kasper Frederiksen, Julia Hesse, Bertram Poettering, and Patrick Towa. 2023. Attribute-based Single Sign-On: Secure, Private, and Efficient. *Cryptology ePrint Archive* (2023).

[138] David Freeman, Sakshi Jain, Markus Dürmuth, Battista Biggio, and Giorgio Giacinto. 2016. Who Are You? A Statistical Approach to Measuring User Authenticity.. In *Proceedings of the Symposium on Network and Distributed System Security*, Vol. 16. 21–24.

[139] Xianyi Gao, Yulong Yang, Can Liu, Christos Mitropoulos, Janne Lindqvist, and Antti Oulasvirta. 2018. Forgetting of passwords: ecological theory and data. In *27th USENIX Security Symposium (USENIX Security 18)*. 221–238.

[140] Anthony Gavazzi, Ryan Williams, Engin Kirda, Long Lu, Andre King, Andy Davis, and Tim Leek. 2023. A Study of Multi-Factor and Risk-Based Authentication Availability. In *32nd USENIX Security Symposium (USENIX Security 23)*. 2043–2060.

[141] Nethanel Gelernter, Senia Kalma, Bar Magnezi, and Hen Porcilan. 2017. The password reset MitM attack. In *2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 251–267.

[142] Eva Gerlitz, Maximilian Häring, Charlotte Theresa Mädler, Matthew Smith, and Christian Tiefenau. 2023. Adventures in recovery land: Testing the account recovery of popular websites when the second factor is lost. In *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. 227–243.

[143] Mohammad Ghasemisharif, Chris Kanich, and Jason Polakis. 2022. Towards automated auditing for account and session management flaws in single sign-on deployments. In *2022 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 1774–1790.

[144] Mohammad Ghasemisharif, Amrutha Ramesh, Stephen Checkoway, Chris Kanich, and Jason Polakis. 2018. O single Sign-Off, where art thou? An empirical analysis of single Sign-On account hijacking and session management on the web. In *27th USENIX Security Symposium (USENIX Security 18)*. 1475–1492.

[145] Sanam Ghorbani Lyastani, Sven Bugiel, and Michael Backes. 2023. A Systematic Study of the Consistency of Two-Factor Authentication User Journeys on Top-Ranked Websites. In *Network and Distributed System Security (NDSS) Symposium 2023*.

[146] Conor Gilsenan, Fuzail Shakir, Noura Alomar, and Serge Egelman. 2023. Security and Privacy Failures in Popular 2FA Apps. In *32nd USENIX Security Symposium (USENIX Security 23)*.

[147] Maximilian Golla, Benedict Beuscher, and Markus Dürmuth. 2016. On the security of cracking-resistant password vaults. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1230–1241.

[148] Maximilian Golla and Markus Dürmuth. 2016. Analyzing 4 million real-world personal knowledge questions (short paper). In *Technology and Practice of Passwords: 9th International Conference, PASSWORDS 2015, Cambridge, UK, December 7–9, 2015, Proceedings 9*. Springer, 39–44.

[149] Maximilian Golla and Markus Dürmuth. 2018. On the accuracy of password strength meters. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1567–1582.

[150] Maximilian Golla, Grant Ho, Marika Lohmus, Monica Pulluri, and Elissa M Redmiles. 2021. Driving 2FA Adoption at Scale: Optimizing Two-Factor Authentication Notification Design Patterns. In *30th USENIX Security Symposium (USENIX Security 21)*. 109–126.

[151] Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa Redmiles, and Blase Ur. 2018. "What was that site doing with my Facebook password?" Designing Password-Reuse Notifications. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1549–1566.

[152] Google. 2022. Security of Passkeys in the Google Password Manager. https://security.googleblog.com/2022/10/SecurityofPasskeysintheGooglePasswordManager.html. Last accessed 28th May 2024.

[153] Google. 2024. How to recover your Google Account or Gmail. https://support.google.com/accounts/answer/7682439?hl=en&sjid=6761530502411005413-EU. Last accessed 30th May 2024.

[154] Google. 2024. Why your account recovery request is delayed. https://support.google.com/accounts/answer/9412469?hl=en&sjid=6761530502411005413-EU. Last accessed 30th May 2024.

[155] Google. May 2024. Passkeys, Cross-Account Protection and new ways we're protecting your accounts. https://blog.google/technology/safety-security/google-passkeys-update-april-2024/.

[156] Paul A Grassi, James L Fenton, Elaine M Newton, Ray Perlner, Andrew Regenscheid, William E Burr, Justin P Richer, Naomi Lefkovitz, Jamie M Danker, Yee-Yin Choong, et al. 2020. Digital identity guidelines: Authentication and lifecycle management [includes updates as of 03-02-2020]. (2020).

[157] Eric Grosse and Mayank Upadhyay. 2012. Authentication at scale. *IEEE Security & Privacy* 11, 1 (2012), 15–22.

[158] Cheng Guo, Brianne Campbell, Apu Kapadia, Michael K Reiter, and Kelly Caine. 2021. Effect of Mood, Location, Trust, and Presence of Others on Video-Based Social Authentication. In *30th USENIX Security Symposium (USENIX Security 21)*. 1–18.

[159] Chengqian Guo, Jingqiang Lin, Quanwei Cai, Wei Wang, Fengjun Li, Qiongxiao Wang, Jiwu Jing, and Bin Zhao. 2021. Uppresso: Untraceable and unlinkable privacy-preserving single sign-on services. *arXiv preprint arXiv:2110.10396* (2021).

[160] Hana Habib, Pardis Emami Naeini, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2018. User behaviors and attitudes under password expiration policies. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. 13–30.

[161] Ameya Hanamsagar, Simon S Woo, Chris Kanich, and Jelena Mirkovic. 2018. Leveraging semantic transformation to investigate password habits and their causes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[162] Alina Hang, Alexander De Luca, and Heinrich Hussmann. 2015. I know what you did last week! do you? dynamic security questions for fallback authentication on smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1383–1392.

[163] Alina Hang, Alexander De Luca, Matthew Smith, Michael Richter, and Heinrich Hussmann. 2015. Where have you been? Using Location-Based security questions for fallback authentication. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. 169–183.

[164] Alina Hang, Alexander De Luca, Emanuel Von Zezschwitz, Manuel Demmler, and Heinrich Hussmann. 2015. Locked your phone? buy a new one? from tales

of fallback authentication on smartphones to actual concepts. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 295–305.

[165] Hao Chen, Zhiyi Zhang, Haozhi Xiong, Ananth Raghunathan). 2023. How Meta is improving password security and preserving privacy. https://engineering.fb.com/2023/08/08/security/how-meta-is-improving-password-security-and-preserving-privacy/. August 2023.

[166] Hao Chen, Zhiyi Zhang, Haozhi Xiong, Ananth Raghunathan). 2024. Passwordless login with passkeys. https://developers.google.com/identity/passkeys. Last accessed 29th May 2024.

[167] SM Taiabul Haque, Shannon Scielzo, and Matthew Wright. 2014. Applying psychometrics to measure user comfort when constructing a strong password. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. 231–242.

[168] Marian Harbach, Alexander De Luca, and Serge Egelman. 2016. The anatomy of smartphone unlocking: A field study of android lock screens. In *Proceedings of the 2016 chi Conference on Human Factors in Computing Systems*. 4806–4817.

[169] Marian Harbach, Sascha Fahl, Matthias Rieger, and Matthew Smith. 2013. On the acceptance of privacy-preserving authentication technology: the curious case of national identity cards. In *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings 13*. Springer, 245–264.

[170] Ruiwen He, Xiaoyu Ji, Xinfeng Li, Yushi Cheng, and Wenyuan Xu. 2022. "OK, Siri" or "Hey, Google": Evaluating Voiceprint Distinctiveness via Content-based PROLE Score. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 1131–1148. https://www.usenix.org/conference/usenixsecurity22/presentation/he-ruiwen

[171] Heather Chen and Kathleen Magramo. 2024. Finance worker pays out $25 million after video call with deepfake 'chief financial officer'. https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk/index.html. February 4, 2024.

[172] Cormac Herley and Paul Van Oorschot. 2011. A research agenda acknowledging the persistence of passwords. *IEEE Security & privacy* 10, 1 (2011), 28–36.

[173] Cormac Herley, Paul C Van Oorschot, and Andrew S Patrick. 2009. Passwords: If we're so smart, why are we still using them?. In *Financial Cryptography and Data Security: 13th International Conference, FC 2009, Accra Beach, Barbados, February 23-26, 2009. Revised Selected Papers 13*. Springer, 230–237.

[174] Brad Hill. 2017. Moving Account Recovery beyond Email and the "Secret" Question. (2017).

[175] Jonas Hofmann and Kien Tuong Truong. 2024. End-to-End Encrypted Cloud Storage in the Wild: A Broken Ecosystem. *Cryptology ePrint Archive* (2024).

[176] Sandra Hölltervennhoff, Noah Wöhler, Arne Möhle, Marten Oltrogge, Yasemin Acar, Oliver Wiese, and Sascha Fahl. 2024. A Mixed-Methods Study on User Experiences and Challenges of Recovery Codes for an End-to-End Encrypted Service. In *In 33rd USENIX Security Symposium*.

[177] Christian Holz and Frank R Bentley. 2016. On-demand biometrics: Fast cross-device authentication. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3761–3766.

[178] Pili Hu, Ronghai Yang, Yue Li, and Wing Cheong Lau. 2014. Application impersonation: problems of OAuth and API design in online social networks. In *Proceedings of the second ACM conference on Online social networks*. 271–278.

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

[180] Jun Ho Huh, Hyoungshick Kim, Swathi SVP Rayala, Rakesh B Bobba, and Konstantin Beznosov. 2017. I'm too busy to reset my LinkedIn password: On the effectiveness of password reset emails. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 387–391.

[181] Jun Ho Huh, Seongyeol Oh, Hyoungshick Kim, Konstantin Beznosov, Apurva Mohan, and S Raj Rajagopalan. 2015. Surpass: System-initiated user-replaceable passwords. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 170–181.

[182] Adryana Hutchinson, Jinwei Tang, Adam J Aviv, and Peter Story. 2024. Measuring the Prevalence of Password Manager Issues Using In-Situ Experiments. (2024).

[183] Information Commissioner's Office. 2024. What is the eIDAS Regulation? https://ico.org.uk/for-organisations/guide-to-eidas/what-is-the-eidas-regulation/. Last accessed November 20th, 2024.

[184] Philip G Inglesant and M Angela Sasse. 2010. The true cost of unusable password policies: password use in the wild. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 383–392.

[185] ISE. 2019. Password Managers: Under the Hood of Secrets Management. https://www.ise.io/casestudies/password-manager-hacking/. February 19, 2019.

[186] Mazharul Islam, Marina Sanusi Bohuk, Paul Chung, Thomas Ristenpart, and Rahul Chatterjee. 2023. Araña: Discovering and Characterizing Password Guessing Attacks in Practice. In *32nd USENIX Security Symposium (USENIX Security 23)*. 1019–1036.

[187] Takanori Isobe and Ryoma Ito. 2021. Security analysis of end-to-end encryption for zoom meetings. *IEEE access* 9 (2021), 90677–90689.

[188] Ivan Krstić. 2023. *Personal Data in the Cloud Is Under Siege. End-to-End Encryption Is Our Most Powerful Defense.*

[189] James Humphreys. 2023. LATEST: Keeper Security's recovery feature. https://securityjournaluk.com/latest-keeper-securitys-recovery-feature/. April 28, 2023.

[190] Abhishek Jana, Md Kamruzzaman Sarker, Monireh Ebrahimi, Pascal Hitzler, and George T Amariucai. 2020. Neural fuzzy extractors: A secure way to use artificial neural networks for biometric user authentication. *arXiv preprint arXiv:2003.08433* (2020).

[191] Mohit Kumar Jangid, Yue Zhang, and Zhiqiang Lin. 2023. Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing.. In *Proceedings of the Symposium on Network and Distributed System Security*.

[192] Ashar Javed, David Bletgen, Florian Kohlar, Markus Dürmuth, and Jörg Schwenk. 2014. Secure fallback authentication and the trusted friend attack. In *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 22–28.

[193] Gokul Chettoor Jayakrishnan, Gangadhara Reddy Sirigireddy, Sukanya Vaddepalli, Vijayanand Banahatti, Sachin Premsukh Lodha, and Sankalp Suneel Pandit. 2020. Password: A serious game to promote password awareness and diversity in an enterprise. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. 1–18.

[194] Joanna Stern and Nicole Nguyen. February 2023. A Basic iPhone Feature Helps Criminals Steal Your Entire Digital Life. https://www.wsj.com/tech/personal-tech/apple-iphone-security-theft-passcode-data-privacya-basic-iphone-feature-helps-criminals-steal-your-digital-life-cbf14b1a.

[195] John Swanson. December 2022. *Raising the bar for software security: next steps for GitHub.com 2FA*.

[196] Jon Brodkin. 2024. Cops can force suspect to unlock phone with thumbprint, US court rules. https://arstechnica.com/tech-policy/2024/04/cops-can-force-suspect-to-unlock-phone-with-thumbprint-us-court-rules/. April 18th, 2024.

[197] Joseph Menn. 2024. Exclusive: Apple dropped plan for encrypting backups after FBI complained. https://www.reuters.com/article/world/exclusive-apple-dropped-plan-for-encrypting-backups-after-fbi-complained-sour-idUSKBN1ZK1CO/. January 22, 2020.

[198] Joseph Menn. 2025. U.K. demand for a back door to Apple data threatens Americans, lawmakers say. https://www.washingtonpost.com/technology/2025/02/13/apple-uk-security-back-door-adp/. February 13, 2025.

[199] Joseph Menn. 2025. U.K. orders Apple to let it spy on users' encrypted accounts. https://www.washingtonpost.com/technology/2025/02/07/apple-encryption-backdoor-uk/. February 7, 2025.

[200] Zeinab Joudaki, Julie Thorpe, and Miguel Vargas Martin. 2018. Reinforcing system-assigned passphrases through implicit learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1533–1548.

[201] Roger Piqueras Jover. 2020. Security analysis of SMS as a second factor of authentication. *Commun. ACM* 63, 12 (2020), 46–52.

[202] Mike Just and David Aspinall. 2009. Personal choice and challenge questions: a security and usability assessment. In *Proceedings of the 5th Symposium on Usable Privacy and Security*. 1–11.

[203] Kaggle. 2024. Alexa Top 1 Million Sites. https://www.kaggle.com/datasets/cheedcheed/top1m. Last accessed November 29th, 2024.

[204] Shirin Kalantari, Pieter Philippaerts, Yana Dimova, Danny Hughes, Wouter Joosen, and Bart De Decker. 2023. A User-Centric Approach to API Delegations: Enforcing Privacy Policies on OAuth Delegations. In *European Symposium on Research in Computer Security*. Springer, 318–337.

[205] Amy K Karlson, AJ Bernheim Brush, and Stuart Schechter. 2009. Can I borrow your phone? Understanding concerns when sharing mobile phones. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1647–1650.

[206] Kashmir Hill. 2022. *A Dad Took Photos of His Naked Toddler for the Doctor. Google Flagged Him as a Criminal.*

[207] Kashmir Hill. 2022. *Her Child's Naked Dance Killed Her Google Account. New Appeals Path Restored It.*

[208] Kashmir Hill. 2023. *How Your Child's Online Mistake Can Ruin Your Digital Life.*

[209] Andre Kassis and Urs Hengartner. 2023. Breaking Security-Critical Voice Authentication. In *2023 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 951–968.

[210] Christina Katsini, Christos Fidas, George E Raptis, Marios Belk, George Samaras, and Nikolaos Avouris. 2018. Influences of human cognition and visual behavior on password strength during picture password composition. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.

[211] Joseph'Jofish' Kaye. 2011. Self-reported password sharing strategies. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2619–2622.

[212] Keeper. 2024. Keeper Encryption and Security Model Details. https://docs.keeper.io/en/enterprise-guide/keeper-encryption-model. Last accessed November 30th, 2024.

[213] Keeper. 2024. Master Password Reset & Account Recovery. https://docs.keeper.io/en/user-guides/troubleshooting/reset-your-master-password. Last accessed November 29th, 2024.

[214] Markus Keil, Philipp Markert, and Markus Dürmuth. 2022. "It's Just a Lot of Prerequisites": A User Perception and Usability Analysis of the German ID Card as a FIDO2 Authenticator. In *Proceedings of the 2022 European Symposium on Usable Security*. 172–188.

[215] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. 2012. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 523–537.

[216] Michal Kepkowski, Lucjan Hanzlik, Ian Wood, and Mohamed Ali Kaafar. 2022. How not to handle keys: Timing attacks on FIDO authenticator privacy. *arXiv preprint arXiv:2205.08071* (2022).

[217] Michal Kepkowski, Maciej Machulak, Ian Wood, and Dali Kaafar. 2023. Challenges with Passwordless FIDO2 in an Enterprise Setting: A Usability Study. In *2023 IEEE Secure Development Conference (SecDev)*. IEEE, 37–48.

[218] Mohamed Khamis, Tobias Seitz, Leonhard Mertl, Alice Nguyen, Mario Schneller, and Zhe Li. 2019. Passquerade: Improving error correction of text passwords on mobile devices by using graphic filters for password masking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–8.

[219] Johannes Kiesel, Benno Stein, and Stefan Lucks. 2017. A Large-scale Analysis of the Mnemonic Password Advice.. In *Proceedings of the Symposium on Network and Distributed System Security*.

[220] Hyoungshick Kim, John Tang, and Ross Anderson. 2012. Social authentication: harder than it looks. In *International conference on financial cryptography and data security*. Springer, 1–15.

[221] Kim Key. 2024. The Best Password Managers for 2024. https://uk.pcmag.com/password-managers/4296/the-best-password-managers. November 19, 2024.

[222] Kim Zetter. 2008. *Palin E-Mail hacker Says It Was Easy.*

[223] Leah Komen. 2016. "Here you can use it": Understanding mobile phone sharing and the concerns it elicits in rural Kenya. *for (e) dialogue* 1, 1 (2016), 52–65.

[224] Ross Koppel, Jim Blythe, Vijay Kothari, and Sean Smith. 2016. Beliefs about cybersecurity rules and passwords: A comparison of two survey samples of cybersecurity professionals versus regular users. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*.

[225] Masoud Mehrabi Koushki, Borke Obada-Obieh, Jun Ho Huh, and Konstantin Beznosov. 2021. On smartphone users' difficulty with understanding implicit authentication. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[226] Krebs on Security. 2023. LastPass: 'Horse Gone Barn Bolted' is Strong Password. https://krebsonsecurity.com/2023/09/lastpass-horse-gone-barn-bolted-is-strong-password/. September 22, 2023.

[227] Maximilian Kroschewski and Anja Lehmann. 2023. Save The Implicit Flow? Enabling Privacy-Preserving RP Authentication in OpenID Connect. *Proceedings on Privacy Enhancing Technologies* (2023).

[228] Dhruv Kuchhal, Muhammad Saad, Adam Oest, and Frank Li. 2023. Evaluating the Security Posture of Real-World FIDO2 Deployments. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2381–2395.

[229] Johannes Kunke, Stephan Wiefling, Markus Ullmann, and Luigi Lo Iacono. 2021. Evaluation of account recovery strategies with FIDO2-based passwordless authentication. *arXiv preprint arXiv:2105.12477* (2021).

[230] Russell WF Lai, Christoph Egger, Dominique Schröder, and Sherman SM Chow. 2017. Phoenix: Rebirth of a Cryptographic Password-Hardening Service. In *26th USENIX Security Symposium (USENIX Security 17)*. 899–916.

[231] Daniele Lain, Kari Kostiainen, and Srdjan Čapkun. 2022. Phishing in organizations: Findings from a large-scale and long-term study. In *2022 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 842–859.

[232] Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. 2021. "It's Stored, Hopefully, on an Encrypted Server": Mitigating Users' Misconceptions About FIDO2 Biometric WebAuthn. In *30th USENIX Security Symposium (USENIX Security 21)*. 91–108.

[233] Leona Lassak, Philipp Markert, Maximilian Golla, Elizabeth Stobert, and Markus Dürmuth. 2024. A Comparative Long-Term Study of Fallback Authentication Schemes. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–19.

[234] Leona Lassak, Elleen Pan, Blase Ur, and Maximilian Golla. 2024. Why Aren't We Using Passkeys? Obstacles Companies Face Deploying FIDO2 Passwordless Authentication. (2024).

[235] LastPass. 2022. 12-22-2022: Notice of Security Incident. https://blog.lastpass.com/posts/notice-of-recent-security-incident. December 22, 2022.

[236] LastPass. 2024. About the encryption process when a super admin resets a master password. https://support.lastpass.com/s/document-item?language=en_US&bundleId=lastpass&topicId=LastPass/reset-master-password-encryption.html&_LANG=enus. May 22, 2024.

[237] LastPass. 2024. Emergency Access. https://www.lastpass.com/features/emergency-access. Last accessed 30th November 2024.

[238] LastPass. 2024. Enable the "Permit super admins to reset master passwords" policy. https://support.lastpass.com/s/document-item?language=en_US&bundleId=lastpass&topicId=LastPass/t_lastpass_enable_super_admin_reset_masterpsw.html&_LANG=enus. July 11, 2024.

[239] LastPass. 2024. LastPass Technical Whitepaper. https://support.lastpass.com/s/document-item?language=en_US&bundleId=lastpass&topicId=LastPass/lastpass_technical_whitepaper.html&_LANG=enus. October 29, 2024.

[240] Jaeho Lee, Ang Chen, and Dan S Wallach. 2019. Total Recall: Persistence of Passwords in Android.. In *Proceedings of the Symposium on Network and Distributed System Security*.

[241] Kevin Lee, Benjamin Kaiser, Jonathan Mayer, and Arvind Narayanan. 2020. An empirical study of wireless carrier authentication for SIM swaps. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. 61–79.

[242] Kevin Lee and Arvind Narayanan. 2021. Security and privacy risks of number recycling at mobile carriers in the United States. In *2021 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 1–17.

[243] Kevin Lee, Sten Sjöberg, and Arvind Narayanan. 2022. Password policies of most top websites fail to follow best practices. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*. 561–580.

[244] Chongqing Lei, Zhen Ling, Yue Zhang, Kai Dong, Kaizheng Liu, Junzhou Luo, and Xinwen Fu. 2023. Do Not Give a Dog Bread Every Time He Wags His Tail: Stealing Passwords through Content Queries (CONQUER) Attacks. In *The Network and Distributed System Security Symposium (NDSS)*. Internet Society.

[245] Zeyu Lei, Yuhong Nan, Yanick Fratantonio, and Antonio Bianchi. 2021. On the insecurity of SMS one-time password messages against local attackers in modern mobile devices. In *Network and Distributed Systems Security (NDSS) Symposium 2021*.

[246] Brittany Lewis and Krishna Venkatasubramanian. 2021. "I... Got my NosePrint. But it Wasn't Accurate": How People with Upper Extremity Impairment Authenticate on their Personal Computing Devices. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[247] Changjiang Li, Li Wang, Shouling Ji, Xuhong Zhang, Zhaohan Xi, Shanqing Guo, and Ting Wang. 2022. Seeing is living? rethinking the security of facial liveness verification in the deepfake era. In *31st USENIX Security Symposium (USENIX Security 22)*. 2673–2690.

[248] Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable re-authentication for smartphones.. In *Proceedings of the Symposium on Network and Distributed System Security*, Vol. 56. Citeseer, 57–59.

[249] Yan Li, Yingjiu Li, Qiang Yan, Hancong Kong, and Robert H Deng. 2015. Seeing your face is not enough: An inertial sensor-based liveness detection for face authentication. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1558–1569.

[250] Yue Li, Haining Wang, and Kun Sun. 2018. Email as a master key: Analyzing account recovery in the wild. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1646–1654.

[251] Zhigong Li, Weili Han, and Wenyuan Xu. 2014. A Large-Scale Empirical Analysis of Chinese Web Passwords. In *23rd USENIX Security Symposium (USENIX Security 14)*. 559–574.

[252] Zhiwei Li, Warren He, Devdatta Akhawe, and Dawn Song. 2014. The Emperor's new password manager: Security analysis of web-based password managers. In *23rd USENIX Security Symposium (USENIX Security 14)*. 465–479.

[253] Chen Liang, Chun Yu, Xiaoying Wei, Xuhai Xu, Yongquan Hu, Yuntao Wang, and Yuanchun Shi. 2021. Auth+ track: Enabling authentication free interaction on smartphone by continuous user tracking. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.

[254] Junchao Lin, Jason I Hong, and Laura Dabbish. 2021. "It's our mutual responsibility to share" The Evolution of Account Sharing in Romantic Couples. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–27.

[255] Xu Lin, Panagiotis Ilia, Saumya Solanki, and Jason Polakis. 2022. Phish in sheep's clothing: Exploring the authentication pitfalls of browser fingerprinting. In *31st USENIX Security Symposium (USENIX Security 22)*. 1651–1668.

[256] Enze Liu, Amanda Nakanishi, Maximilian Golla, David Cash, and Blase Ur. 2019. Reasoning analytically about password-cracking software. In *2019 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 380–397.

[257] Jian Liu, Chen Wang, Yingying Chen, and Nitesh Saxena. 2017. VibWrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 73–87.

[258] Peiyuan Liu, Jeremiah Blocki, and Wenjie Bai. 2023. Confident monte carlo: Rigorous analysis of guessing curves for probabilistic password models. In *2023 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 626–644.

[259] Lorenzo Franceschi-Biccherai. 2018. Stop Using 6-Digit iPhone Passcodes. https://www.vice.com/en/article/how-to-make-a-secure-iphone-passcode-6-digits/. April 16, 2018.

[260] Sanam Ghorbani Lyastani, Michael Schilling, Sascha Fahl, Michael Backes, and Sven Bugiel. 2018. Better managed than memorized? Studying the Impact of

Managers on Password Strength and Reuse. In *27th USENIX Security Symposium (USENIX Security 18)*. 203–220.
[261] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. 2020. Is fido2 the kingslayer of user authentication? a comparative usability study of fido2 passwordless authentication. In *2020 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 268–285.
[262] Jerry Ma, Weining Yang, Min Luo, and Ninghui Li. 2014. A study of probabilistic password models. In *2014 IEEE Symposium on Security and Privacy*. IEEE, 689–704.
[263] Christian Mainka, Vladislav Mladenov, and Jörg Schwenk. 2016. Do not trust me: Using malicious IdPs for analyzing and attacking single sign-on. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 321–336.
[264] Christian Mainka, Vladislav Mladenov, Jörg Schwenk, and Tobias Wich. 2017. SoK: single sign-on security—an evaluation of openID connect. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 251–266.
[265] Mark Risher. May 2021. A simpler and safer future — without passwords. https://blog.google/technology/safety-security/a-simpler-and-safer-future-without-passwords/. (May 2021).
[266] Philipp Markert, Daniel V Bailey, Maximilian Golla, Markus Dürmuth, and Adam J Aviv. 2020. This pin can be easily guessed: Analyzing the security of smartphone unlock pins. In *2020 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 286–303.
[267] Philipp Markert, Daniel V Bailey, Maximilian Golla, Markus Dürmuth, and Adam J Aviv. 2021. On the security of smartphone unlock PINs. *ACM Transactions on Privacy and Security (TOPS)* 24, 4 (2021), 1–36.
[268] Philipp Markert, Florian Farke, and Markus Dürmuth. 2019. View the email to get hacked: Attacking SMS-based two-factor authentication. *Who Are You* (2019), 1–6.
[269] Philipp Markert, Leona Lassak, Maximilian Golla, and Markus Dürmuth. 2024. Understanding Users' Interaction with Login Notifications. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–17.
[270] Philipp Markert, Theodor Schnitzler, Maximilian Golla, and Markus Dürmuth. 2022. "As soon as it's a risk, I want to require MFA": How Administrators Configure Risk-based Authentication. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*. 483–501.
[271] Karola Marky, Kirill Ragozin, George Chernyshov, Andrii Matviienko, Martin Schmitz, Max Mühlhäuser, Chloe Eghtebas, and Kai Kunze. 2022. "Nah, it's just annoying!" A Deep Dive into User Perceptions of Two-Factor Authentication. *ACM transactions on computer-human interaction* 29, 5 (2022), 1–32.
[272] Karola Marky, Martin Schmitz, Verena Zimmermann, Martin Herbers, Kai Kunze, and Max Mühlhäuser. 2020. 3d-auth: Two-factor authentication with personalized 3d-printed items. In *Proceedings of the 2020 chi Conference on Human Factors in Computing Systems*. 1–12.
[273] Sonali Tukaram Marne, Mahdi Nasrullah Al-Ameen, and Matthew K Wright. 2017. Learning System-assigned Passwords: A Preliminary Study on the People with Learning Disabilities.. In *SOUPS*.
[274] Tara Matthews, Kerwell Liao, Anna Turner, Marianne Berkovich, Robert Reeder, and Sunny Consolvo. 2016. "She'll just grab any device that's closer" A Study of Everyday Device & Account Sharing in Households. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5921–5932.
[275] Max Eddy). 2024. So You're Locked Out of Your Two-Factor Authentication App. Don't Panic. https://www.nytimes.com/wirecutter/guides/locked-out-two-factor-authentication-app-recovery/. April 2024.
[276] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. 2013. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 173–186.
[277] William Melicher, Darya Kurilova, Sean M Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. 2016. Usability and security of text passwords on mobile devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 527–539.
[278] William Melicher, Blase Ur, Sean M Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Fast, lean, and accurate: Modeling password guessability using neural networks. In *25th USENIX Security Symposium (USENIX Security 16)*. 175–191.
[279] Weizhi Meng, Wenjuan Li, Lijun Jiang, and Liying Meng. 2016. On multiple password interference of touch screen patterns and text passwords. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 4818–4822.
[280] Meta. 2023. *The Labyrinth Encrypted Message Storage Protocol*.
[281] Meta. September 2021. *How WhatsApp is enabling end-to-end encrypted backups*.
[282] Nicholas Micallef and Nalin Asanka Gamagedara Arachchilage. 2017. A Gamified Approach to Improve Users' Memorability of Fall-back Authentication.. In *SOUPS*.
[283] Microsoft. 2024. Require end-to-end encryption for sensitive Teams meetings. https://learn.microsoft.com/en-us/microsoftteams/end-to-end-encrypted-meetings. September 18, 2024.
[284] Grzergor Milka. 2018. Anatomy of account takeover. In *Enigma 2018 (Enigma 2018)*.
[285] Ariana Mirian, Joe DeBlasio, Stefan Savage, Geoffrey M Voelker, and Kurt Thomas. 2019. Hack for hire: Exploring the emerging market for account hijacking. In *The World Wide Web Conference*. 1279–1289.
[286] Srivathsan G Morkonda, Sonia Chiasson, and Paul C van Oorschot. 2021. Empirical analysis and privacy implications in OAuth-based single sign-on systems. In *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*. 195–208.
[287] Srivathsan G Morkonda, Sonia Chiasson, and Paul C van Oorschot. 2022. "Sign in with…Privacy": Timely Disclosure of Privacy Differences among Web SSO Login Options. *arXiv preprint arXiv:2209.04490* (2022).
[288] Collin Mulliner, Ravishankar Borgaonkar, Patrick Stewin, and Jean-Pierre Seifert. 2013. SMS-Based One-Time Passwords: Attacks and Defense: (Short Paper). In *Detection of Intrusions and Malware, and Vulnerability Assessment: 10th International Conference, DIMVA 2013, Berlin, Germany, July 18-19, 2013. Proceedings 10*. Springer, 150–159.
[289] Collins W Munyendo, Yasemin Acar, and Adam J Aviv. 2023. "In Eighty Percent of the Cases, I Select the Password for Them": Security and Privacy Challenges, Advice, and Opportunities at Cybercafes in Kenya. In *2023 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 570–587.
[290] Collins W Munyendo, Philipp Markert, Alexandra Nisenoff, Miles Grant, Elena Korkes, Blase Ur, and Adam J Aviv. 2022. "The Same PIN, Just Longer": On the (In) Security of Upgrading PINs from 4 to 6 Digits. In *31st USENIX Security Symposium (USENIX Security 22)*. 4023–4040.
[291] Collins W Munyendo, Peter Mayer, and Adam J Aviv. 2023. "I just stopped using one and started using the other": Motivations, Techniques, and Challenges When Switching Password Managers. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 3123–3137.
[292] Alena Naiakshina, Anastasia Danilova, Eva Gerlitz, Emanuel Von Zezschwitz, and Matthew Smith. 2019. "If you want, I can store the encrypted password" A Password-Storage Field Study with Freelance Developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[293] Alena Naiakshina, Anastasia Danilova, Christian Tiefenau, Marco Herzog, Sergej Dechand, and Matthew Smith. 2017. Why do developers get password storage wrong? A qualitative usability study. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 311–328.
[294] James Nicholson, Lynne Coventry, and Pam Briggs. 2013. Age-related performance issues for PIN and face-based authentication systems. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 323–332.
[295] Nicolas Bacca. 2024. Passkeys: The Good, The Bad, The Ugly. https://www.youtube.com/watch?v=TEjNSr8jjUI&ab_channel=EthereumFoundation. November 17, 2024.
[296] Alexandra Nisenoff, Maximilian Golla, Miranda Wei, Juliette Hainline, Hayley Szymanek, Annika Braun, Annika Hildebrandt, Blair Christensen, David Langenberg, and Blase Ur. 2023. A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords. In *32nd USENIX Security Symposium (USENIX Security 23)*. 5127–5144.
[297] Nora Trapp. 2024. Key to Simplicity: Squeezing the hassle out of encryption key recovery. https://juicebox.xyz/blog/key-to-simplicity-squeezing-the-hassle-out-of-encryption-key-recovery. April 9, 2024.
[298] NordPass. 2024. Account Recovery for business users. https://support.nordpass.com/hc/en-us/articles/360017323858-Account-Recovery-for-business-users. Last accessed November 29th, 2024.
[299] NordPass. 2024. Overview. https://support.nordpass.com/hc/en-us/sections/26090886272529-Overview. Last accessed November 30th, 2024.
[300] Christopher Novak, Jim Blythe, Ross Koppel, Vijay H Kothari, and Sean W Smith. 2017. Modeling Aggregate Security with User Agents that Employ Password Memorization Techniques.. In *SOUPS*.
[301] Tunde Oduguwa and Abdullahi Arabo. 2024. Passwordless Authentication Using a Combination of Cryptography, Steganography, and Biometrics. *Journal of Cybersecurity and Privacy* 4, 2 (2024), 278–297.
[302] Sean Oesch and Scott Ruoti. 2020. That was then, this is now: A security evaluation of password generation, storage, and autofill in browser-based password managers. In *Proceedings of the 29th USENIX Conference on Security Symposium*. 2165–2182.
[303] Sean Oesch, Scott Ruoti, James Simmons, and Anuj Gautam. 2022. "It Basically Started Using Me:" An Observational Study of Password Manager Usage. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–23.
[304] Adam Oest, Yeganeh Safaei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Kevin Tyers. 2019. Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In *2019 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 1344–1361.
[305] Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé, and Gail-Joon Ahn. 2020. Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In *29th USENIX Security Symposium (USENIX Security 20)*.

[306] Wataru Oogami, Hidehito Gomi, Shuji Yamaguchi, Shota Yamanaka, and Tatsuru Higurashi. 2020. Observation study on usability challenges for fingerprint authentication using WebAuthn-enabled android smartphones. *Age* 20 (2020), 29.

[307] Chris Orsini, Alessandra Scafuro, and Tanner Verber. 2023. How to Recover a Cryptographic Secret From the Cloud. *Cryptology ePrint Archive* (2023).

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

[309] Bijeeta Pal, Tal Daniel, Rahul Chatterjee, and Thomas Ristenpart. 2019. Beyond credential stuffing: Password similarity models using neural networks. In *2019 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 417–434.

[310] Bijeeta Pal, Mazharul Islam, Marina Sanusi Bohuk, Nick Sullivan, Luke Valenta, Tara Whalen, Christopher Wood, Thomas Ristenpart, and Rahul Chatterjee. 2022. Might I get pwned: A second generation compromised credential checking service. In *31st USENIX Security Symposium (USENIX Security 22)*. 1831–1848.

[311] Saurabh Panjwani and Achintya Prakash. 2014. Crowdsourcing attacks on biometric systems. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. 257–269.

[312] Simon Parkin, Samy Driss, Kat Krol, and M Angela Sasse. 2016. Assessing the user experience of password reset policies in a university. In *Technology and Practice of Passwords: 9th International Conference, PASSWORDS 2015, Cambridge, UK, December 7–9, 2015, Proceedings 9*. Springer, 21–38.

[313] Dario Pasquini, Ankit Gangwal, Giuseppe Ateniese, Massimo Bernaschi, and Mauro Conti. 2021. Improving password guessing via representation learning. In *2021 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 1382–1399.

[314] Rizu Paudel, Prakriti Dumaru, Ankit Shrestha, Huzeyfe Kocabas, and Mahdi Nasrullah Al-Ameen. 2023. A Deep Dive into User's Preferences and Behavior around Mobile Phone Sharing. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–22.

[315] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, and Alain Forget. 2017. Let's go in for a closer look: Observing passwords in their natural habitat. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 295–310.

[316] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2019. Why people (don't) use password managers effectively. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 319–338.

[317] Thanasis Petsas, Giorgos Tsirantonakis, Elias Athanasopoulos, and Sotiris Ioannidis. 2015. Two-factor authentication: is the world ready? Quantifying 2FA adoption. In *Proceedings of the eighth european workshop on system security*. 1–7.

[318] Katharina Pfeffer, Alexandra Mai, Adrian Dabrowski, Matthias Gusenbauer, Philipp Schindler, Edgar Weippl, Michael Franz, and Katharina Krombholz. 2021. On the usability of authenticity checks for hardware security tokens. In *30th USENIX Security Symposium (USENIX Security 21)*. 37–54.

[319] Jamie L Pinchot and Karen L Paullet. 2012. What's in your profile? Mapping Facebook profile data to personal security questions. *Issues in Information Systems* 13, 1 (2012), 284–293.

[320] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2018. Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156* (2018).

[321] Iasonas Polakis, Panagiotis Ilia, Federico Maggi, Marco Lancini, Georgios Kontaxis, Stefano Zanero, Sotiris Ioannidis, and Angelos D Keromytis. 2014. Faces in the distorting mirror: Revisiting photo-based social authentication. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 501–512.

[322] Iasonas Polakis, Marco Lancini, Georgios Kontaxis, Federico Maggi, Sotiris Ioannidis, Angelos D Keromytis, and Stefano Zanero. 2012. All your face are belong to us: Breaking facebook's social authentication. In *Proceedings of the 28th annual computer security applications conference*. 399–408.

[323] PreVeil. 2019. *PreVeil Security and Design: A Description of the PreVeil System Architecture*.

[324] Proton. 2024. *Set account recovery methods in case you forget your Proton password*.

[325] Nils Quermann, Marian Harbach, and Markus Dürmuth. 2018. The state of user authentication in the wild. *WAY* 18 (2018).

[326] Ariel Rabkin. 2008. Personal knowledge questions for fallback authentication: Security questions in the era of Facebook. In *Proceedings of the 4th Symposium on Usable Privacy and Security*. 13–23.

[327] Rachel Hall. 2025. Apple to appeal against UK government data demand at secret high court hearing. https://www.theguardian.com/technology/2025/mar/12/apple-to-appeal-against-uk-government-data-demand-at-secret-high-court-hearing. March 12, 2025.

[328] George E Raptis, Christina Katsini, Andrew Jian-Lan Cen, Nalin Asanka Gamagedara Arachchilage, and Lennart E Nacke. 2021. Better, funner, stronger: a gameful approach to nudge people into making less predictable graphical password choices. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.

[329] Hirak Ray, Flynn Wolf, Ravi Kuber, and Adam J Aviv. 2021. Why older adults (Don't) use password managers. In *30th USENIX Security Symposium (USENIX Security 21)*. 73–90.

[330] Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin RB Butler. 2016. Sending out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways. In *2016 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 339–356.

[331] Elissa M Redmiles, Everest Liu, and Michelle L Mazurek. 2017. You Want Me To Do What? A Design Study of Two-Factor Authentication Messages.. In *SOUPS*, Vol. 57. 93.

[332] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, and Kent Seamons. 2019. A usability study of five two-factor authentication methods. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 357–370.

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

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

[335] Hossein Rezaeighaleh and Cliff C Zou. 2019. New secure approach to backup cryptocurrency wallets. In *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

[336] Scott Ruoti, Jeff Andersen, and Kent Seamons. 2016. Strengthening password-based authentication. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*.

[337] Napa Sae-Bae, Kowsar Ahmed, Katherine Isbister, and Nasir Memon. 2012. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In *proceedings of the CHI Conference on Human Factors in Computing Systems*. 977–986.

[338] Sena Sahin, Suood Al Roomi, Tara Poteat, and Frank Li. 2023. Investigating the Password Policy Practices of Website Administrators. In *2023 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 552–569.

[339] Sena Sahin and Frank Li. 2021. Don't forget the stuffing! revisiting the security impact of typo-tolerant password authentication. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 252–270.

[340] Nithya Sambasivan, Garen Checkley, Amna Batool, Nova Ahmed, David Nemer, Laura Sanely Gaytán-Lugo, Tara Matthews, Sunny Consolvo, and Elizabeth Churchill. 2018. "Privacy is not for me, it's for those rich women"': Performative Privacy Practices on Mobile Phones by Women in South Asia. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. 127–142.

[341] Alessandra Scafuro. 2019. Break-glass encryption. In *Public-Key Cryptography–PKC 2019: 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II 22*. Springer, 34–62.

[342] Stuart Schechter, Serge Egelman, and Robert W Reeder. 2009. It's not what you know, but who you know: a social approach to last-resort authentication. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1983–1992.

[343] Fabian Schwarz, Khue Do, Gunnar Heide, Lucjan Hanzlik, and Christian Rossow. 2022. Feido: Recoverable FIDO2 tokens using electronic ids. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2581–2594.

[344] Sean M Segreti, William Melicher, Saranga Komanduri, Darya Melicher, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. 2017. Diversify to survive: Making passwords stronger with adaptive policies. In *Thirteenth symposium on usable privacy and security (SOUPS 2017)*. 1–12.

[345] Sunyoung Seiler-Hwang, Patricia Arias-Cabarcos, Andrés Marín, Florina Almenares, Daniel Díaz-Sánchez, and Christian Becker. 2019. "I don't see why I would ever want to use it" Analyzing the Usability of Popular Smartphone Password Managers. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1937–1953.

[346] Tobias Seitz, Manuel Hartmann, Jakob Pfab, and Samuel Souque. 2017. Do differences in password policies prevent password reuse?. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2056–2063.

[347] Asuman Senol, Alisha Ukani, Dylan Cutler, and Igor Bilogrevic. 2024. The Double Edged Sword: Identifying Authentication Pages and their Fingerprinting Behavior. In *The Web Conference (WWW), 2024*.

[348] Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613.

[349] Richard Shay, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Alain Forget, Saranga Komanduri, Michelle L Mazurek, William Melicher, Sean M Segreti, and Blase Ur. 2015. A spoonful of sugar? The impact of guidance and feedback on password-creation behavior. In *Proceedings of the 33rd annual ACM Conference on Human Factors in Computing Systems*. 2903–2912.

[350] Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2014. Can long passwords be secure and usable?. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2927–2936.

[351] Maliheh Shirvanian, Stanislaw Jarecki, Nitesh Saxena, and Naveen Nathan. 2014. Two-Factor Authentication Resilient to Server Compromise Using Mix-Bandwidth Devices.. In *Proceedings of the Symposium on Network and Distributed System Security*.

[352] Hossein Siadati, Toan Nguyen, Payas Gupta, Markus Jakobsson, and Nasir Memon. 2017. Mind your SMSes: Mitigating social engineering in second factor authentication. *Computers & Security* 65 (2017), 14–28.

[353] David Silver, Suman Jana, Dan Boneh, Eric Chen, and Collin Jackson. 2014. Password managers: Attacks and defenses. In *23rd USENIX Security Symposium (USENIX Security 14)*. 449–464.

[354] Silviu Stahie. 2021. Microsoft Teams Rolls Out End-to-End Encryption. https://www.bitdefender.com/en-gb/blog/hotforsecurity/microsoft-teams-rolls-out-end-to-end-encryption. October 25, 2021.

[355] Ivo Sluganovic, Marc Roeschlin, Kasper B Rasmussen, and Ivan Martinovic. 2016. Using reflexive eye movements for fast challenge-response authentication. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1056–1067.

[356] Garrett Smith, Tarun Yadav, Jonathan Dutson, Scott Ruoti, and Kent Seamons. 2023. "If I could do this, I feel anyone could:" The Design and Evaluation of a Secondary Authentication Factor Manager. In *32nd USENIX Security Symposium (USENIX Security 23)*. 499–515.

[357] Trevor Smith, Scott Ruoti, and Kent E Seamons. 2017. Augmenting Centralized Password Management with Application-Specific Passwords.. In *SOUPS*.

[358] Juraj Somorovsky, Andreas Mayer, Jörg Schwenk, Marco Kampmann, and Meiko Jensen. 2012. On breaking SAML: Be whoever you want to be. In *21st USENIX Security Symposium (USENIX Security 12)*. 397–412.

[359] Yunpeng Song, Zhongmin Cai, and Zhi-Li Zhang. 2017. Multi-touch authentication using hand geometry and behavioral information. In *2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 357–372.

[360] Yunpeng Song, Cori Faklaris, Zhongmin Cai, Jason I Hong, and Laura Dabbish. 2019. Normal and easy: Account sharing practices in the workplace. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–25.

[361] StartMail. 2024. I forgot my password - now what? https://support.startmail.com/hc/en-us/articles/360007388898--I-forgot-my-password-now-what. Last accessed 1 June 2024.

[362] Vlasta Stavova, Vashek Matyas, and Mike Just. 2016. Codes v. people: A comparative usability study of two password recovery mechanisms. In *Information Security Theory and Practice: 10th IFIP WG 11.2 International Conference, WISTP 2016, Heraklion, Crete, Greece, September 26–27, 2016, Proceedings 10*. Springer, 35–50.

[363] Elizabeth Stobert and Robert Biddle. 2018. The password life cycle. *ACM Transactions on Privacy and Security (TOPS)* 21, 3 (2018), 1–32.

[364] He Sun, Kun Sun, Yuewu Wang, and Jiwu Jing. 2015. TrustOTP: Transforming smartphones into secure one-time password tokens. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 976–988.

[365] San-Tsai Sun and Konstantin Beznosov. 2012. The devil is in the (implementation) details: an empirical analysis of OAuth SSO systems. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 378–390.

[366] San-Tsai Sun, Yazan Boshmaf, Kirstie Hawkey, and Konstantin Beznosov. 2010. A billion keys, but few locks: the crisis of web single sign-on. In *Proceedings of the 2010 new security paradigms workshop*. 61–72.

[367] San-Tsai Sun, Kirstie Hawkey, and Konstantin Beznosov. 2012. Systematically breaking and fixing OpenID security: Formal analysis, semi-automated empirical evaluation, and practical countermeasures. *Computers & Security* 31, 4 (2012), 465–483.

[368] San-Tsai Sun, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, and Konstantin Beznosov. 2011. What makes users refuse web single sign-on? An empirical investigation of OpenID. In *Proceedings of the seventh symposium on usable privacy and security*. 1–20.

[369] San-Tsai Sun, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, and Konstantin Beznosov. 2013. Investigating users' perspectives of web single sign-on: Conceptual gaps and acceptance model. *ACM Transactions on Internet Technology (TOIT)* 13, 1 (2013), 1–35.

[370] Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2020. Practical recommendations for stronger, more usable passwords combining minimum-strength, minimum-length, and blocklist requirements. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1407–1426.

[371] Taryn Plumb). 2024. Face off: Attackers are stealing biometrics to access victims' bank accounts. https://venturebeat.com/security/face-off-attackers-are-stealing-biometrics-to-access-victims-bank-accounts/. February 2024.

[372] Chee Meng Tey, Payas Gupta, and Debin Gao. 2013. I can be you: Questioning the use of keystroke dynamics as biometrics. (2013).

[373] Mary Theofanos, Yee-Yin Choong, and Olivia Murphy. 2021. 'Passwords Keep Me Safe'–Understanding What Children Think about Passwords. In *30th USENIX Security Symposium (USENIX Security 21)*. 19–35.

[374] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, et al. 2017. Data breaches, phishing, or malware? Understanding the risks of stolen credentials. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1421–1434.

[375] Kurt Thomas, Jennifer Pullman, Kevin Yeo, Ananth Raghunathan, Patrick Gage Kelley, Luca Invernizzi, Borbala Benko, Tadek Pietraszek, Sarvar Patel, Dan Boneh, et al. 2019. Protecting accounts from credential stuffing with password breach alerting. In *28th USENIX Security Symposium (USENIX Security 19)*. 1556–1571.

[376] Julie Thorpe, Muath Al-Badawi, Brent MacRae, and Amirali Salehi-Abari. 2014. The presentation effect on graphical passwords. In *proceedings of the CHI Conference on Human Factors in Computing Systems*. 2947–2950.

[377] Jing Tian, Chengzhang Qu, Wenyuan Xu, and Song Wang. 2013. KinWrite: Handwriting-Based Authentication Using Kinect.. In *Proceedings of the Symposium on Network and Distributed System Security*, Vol. 93. 94.

[378] Tresorit. 2023. End-to-end encryption without key rotation is a fatal shortcut. https://tresorit.com/blog/end-to-end-encryption-without-key-rotation-is-a-fatal-shortcut/. June 2, 2023.

[379] Michael Troncoso and Britta Hale. 2021. The Bluetooth cyborg: Analysis of the full human-machine passkey entry AKE protocol. *Cryptology ePrint Archive* (2021).

[380] Mark Turner, Martin Schmitz, Morgan Masichi Bierey, Mohamed Khamis, and Karola Marky. 2023. Tangible 2FA–An In-the-Wild Investigation of User-Defined Tangibles for Two-Factor Authentication. In *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. 245–261.

[381] Giannis Tzagarakis, Panagiotis Papadopoulos, Antonios A Chariton, Elias Athanasopoulos, and Evangelos P Markatos. 2018. Øpass: Zero-storage password management based on password reminders. In *Proceedings of the 11th European Workshop on Systems Security*. 1–6.

[382] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the security of graphical passwords: The case of android unlock patterns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 161–172.

[383] Enis Ulqinaku, Hala Assal, AbdelRahman Abdou, Sonia Chiasson, and Srdjan Capkun. 2021. Is real-time phishing eliminated with FIDO? social engineering downgrade attacks against FIDO protocols. In *30th USENIX Security Symposium (USENIX Security 21)*. 3811–3828.

[384] Blase Ur, Jonathan Bees, Sean M Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Do users' perceptions of password security match reality?. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3748–3760.

[385] Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, et al. 2012. How does your password measure up? The effect of strength meters on password creation. In *21st USENIX Security Symposium (USENIX Security 12)*. 65–80.

[386] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2015. "I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab. In *Eleventh symposium on usable privacy and security (SOUPS 2015)*. 123–140.

[387] Blase Ur, Sean M Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, Darya Kurilova, Michelle L Mazurek, William Melicher, and Richard Shay. 2015. Measuring Real-World Accuracies and Biases in Modeling Password Guessability. In *24th USENIX Security Symposium (USENIX Security 15)*. 463–481.

[388] USA.gov. 2024. How to replace lost or stolen ID cards. https://www.usa.gov/replace-vital-documents. Last accessed November 30th, 2024.

[389] Warda Usman, Jackie Hu, McKynlee Wilson, and Daniel Zappala. 2023. Distrust of big tech and a desire for privacy: Understanding the motivations of people who have voluntarily adopted secure email. In *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. 473–490.

[390] Rosa Van Koningsbruggen, Bart Hengeveld, and Jason Alexander. 2021. Understanding the Design Space of Embodied Passwords based on Muscle Memory. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[391] Rafael Veras, Christopher Collins, and Julie Thorpe. 2014. On semantic patterns of passwords and their security impact.. In *Proceedings of the Symposium on Network and Distributed System Security*. Citeseer.

[392] Thijs Veugen, Robbert de Haan, Ronald Cramer, and Frank Muller. 2014. A framework for secure computations with two non-colluding servers and multiple clients, applied to recommendations. *IEEE Transactions on Information Forensics and Security* 10, 3 (2014), 445–457.

[393] WABetaInfo. 2023. *Apple is releasing password reminder for end-to-end encrypted backups*.

[394] Chen Wang, Xiaonan Guo, Yan Wang, Yingying Chen, and Bo Liu. 2016. Friend or foe? Your wearable devices reveal your personal pin. In *Proceedings of the 11th*

ACM on Asia conference on computer and communications security. 189–200.
[395] Ding Wang, Xuan Shan, Qiying Dong, Yaosheng Shen, and Chunfu Jia. 2023. No single silver bullet: Measuring the accuracy of password strength meters. In *32nd USENIX Security Symposium (USENIX Security 23)*. 947–964.
[396] Ding Wang, Ping Wang, Debiao He, and Yuan Tian. 2019. Birthday, name and bifacial-security: understanding passwords of Chinese web users. In *28th USENIX Security Symposium (USENIX security 19)*. 1537–1555.
[397] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. 2016. Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1242–1254.
[398] Ding Wang, Yunkai Zou, Qiying Dong, Yuanming Song, and Xinyi Huang. 2022. How to attack and generate honeywords. In *2022 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 966–983.
[399] Ding Wang, Yunkai Zou, Yuan-An Xiao, Siqi Ma, and Xiaofeng Chen. 2023. Pass2Edit: A Multi-Step Generative Model for Guessing Edited Passwords. In *32nd USENIX Security Symposium (USENIX Security 23)*. 983–1000.
[400] Ding Wang, Yunkai Zou, Zijian Zhang, and Kedong Xiu. 2023. Password guessing using random forest. In *32nd USENIX Security Symposium (USENIX Security 23)*. 965–982.
[401] Hui Wang, Yuanyuan Zhang, Juanru Li, Hui Liu, Wenbo Yang, Bodong Li, and Dawu Gu. 2015. Vulnerability assessment of oauth implementations in android applications. In *Proceedings of the 31st annual computer security applications conference*. 61–70.
[402] Ke Coby Wang and Michael K Reiter. 2018. How to end password reuse on the web. *arXiv preprint arXiv:1805.00566* (2018).
[403] Qiwen Wang and Mikael Skoglund. 2019. Symmetric private information retrieval from MDS coded distributed storage with non-colluding and colluding servers. *IEEE Transactions on Information Theory* 65, 8 (2019), 5160–5175.
[404] Rui Wang, Shuo Chen, and XiaoFeng Wang. 2012. Signing me onto your accounts through facebook and google: A traffic-guided security study of commercially deployed single-sign-on web services. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 365–379.
[405] Rui Wang, Yuchen Zhou, Shuo Chen, Shaz Qadeer, David Evans, and Yuri Gurevich. 2013. Explicating SDKs: uncovering assumptions underlying secure authentication and authorization. In *22nd USENIX Security Symposium (USENIX Security 13)*. 399–314.
[406] Serena Wang, Cori Faklaris, Junchao Lin, Laura Dabbish, and Jason I Hong. 2022. 'It's Problematic but I'm not Concerned': University Perspectives on Account Sharing. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–27.
[407] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. 2016. Understanding password choices: How frequently entered passwords are re-used across websites. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. 175–188.
[408] Miranda Wei, Jaron Mink, Yael Eiger, Tadayoshi Kohno, Elissa M Redmiles, and Franziska Roesner. 2024. SoK (or SoLK?): On the Quantitative Study of Sociodemographic Factors and Computer Security Behaviors. *arXiv preprint arXiv:2404.10187* (2024).
[409] Wes Davis. June 2023. Apple IDs now support passkeys — if you're on the iOS 17 or macOS Sonoma betas. https://www.theverge.com/2023/6/20/23767583/apple-iphone-ios-17-beta-passkey. (June 2023).
[410] Maximilian Westers, Tobias Wich, Louis Jannett, Vladislav Mladenov, Christian Mainka, and Andreas Mayer. 2023. SSO-monitor: fully-automatic large-scale landscape, security, and privacy analyses of single sign-on in the wild. *arXiv preprint arXiv:2302.01024* (2023).
[411] WhatsApp. 2021. *Security of End-To-End Encrypted Backups*.
[412] WhatsApp. 2023. *Can't remember password for encrypted backup* .
[413] Daniel Lowe Wheeler. 2016. zxcvbn:Low-Budget Password Strength Estimation. In *25th USENIX Security Symposium (USENIX Security 16)*. 157–173.
[414] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. 2020. More than just good passwords? A study on usability and security perceptions of risk-based authentication. In *Proceedings of the 36th Annual Computer Security Applications Conference*. 203–218.
[415] Stephan Wiefling, Paul René Jørgensen, Sigurd Thunem, and Luigi Lo Iacono. 2022. Pump up password security! Evaluating and enhancing risk-based authentication on a real-world large-scale online service. *ACM Transactions on Privacy and Security* 26, 1 (2022), 1–36.
[416] Stephan Wiefling, Luigi Lo Iacono, and Markus Dürmuth. 2019. Is this really you? An empirical study on risk-based authentication applied in the wild. In *ICT Systems Security and Privacy Protection: 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings 34*. Springer, 134–148.
[417] Stephan Wiefling, Jan Tolsdorf, and Luigi Lo Iacono. 2021. Privacy considerations for risk-based authentication systems. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 320–327.
[418] Wladimir Palant. 2022. What data does LastPass encrypt? https://palant.info/2022/12/24/what-data-does-lastpass-encrypt/. December 24, 2022.
[419] World Wide Web Consortium (W3C). 2023. Web Authentication: An API for accessing Public Key Credentials Level 3. https://www.w3.org/TR/webauthn-3/. Last accessed 28th May 2024.
[420] Worldcoin Foundation. 2024. Worldcoin – A New Identity and Financial Network. https://whitepaper.worldcoin.org/. Last accessed 31st May 2024.
[421] Cong Wu, Kun He, Jing Chen, Ziming Zhao, and Ruiying Du. 2020. Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks. In *29th USENIX Security Symposium (USENIX Security 20)*. 2219–2236.
[422] Shujiang Wu, Pengfei Sun, Yao Zhao, and Yinzhi Cao. 2023. Him of Many Faces: Characterizing Billion-scale Adversarial and Benign Browser Fingerprints on Commercial Websites.. In *Proceedings of the Symposium on Network and Distributed System Security*.
[423] Yuxi Wu, W Keith Edwards, and Sauvik Das. 2022. SoK: Social Cybersecurity. In *2022 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 1863–1879.
[424] Zhihao Wu, Yushi Cheng, Jiahui Yang, Xiaoyu Ji, and Wenyuan Xu. 2023. DepthFake: Spoofing 3D Face Authentication with a 2D Photo. In *2023 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 917–91373.
[425] Leon Würsching, Florentin Putz, Steffen Haesler, and Matthias Hollick. 2023. FIDO2 the Rescue? Platform vs. Roaming Authentication on Smartphones. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16.
[426] Hui Xu, Yangfan Zhou, and Michael R Lyu. 2014. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. 187–198.
[427] Ming Xu, Jitao Yu, Xinyi Zhang, Chuanwang Wang, Shenghao Zhang, Haoqi Wu, and Weili Han. 2023. Improving real-world password guessing attacks via bi-directional transformers. In *32nd USENIX Security Symposium (USENIX Security 23)*. 1001–1018.
[428] Qiang Yan, Jin Han, Yingjiu Li, Huijie DENG, et al. 2012. On limitations of designing usable leakage-resilient password systems: Attacks, principles and usability. (2012).
[429] Yueli Yan and Zhice Yang. 2023. Spoofing real-world face authentication systems through optical synthesis. In *2023 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 882–898.
[430] Ronghai Yang, Wing Cheong Lau, Jiongyi Chen, and Kehuan Zhang. 2018. Vetting Single Sign-OnSDK Implementations via Symbolic Reasoning. In *27th USENIX Security Symposium (USENIX Security 18)*. 1459–1474.
[431] Ronghai Yang, Guanchen Li, Wing Cheong Lau, Kehuan Zhang, and Pili Hu. 2016. Model-based security testing: An empirical study on oauth 2.0 implementations. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. 651–662.
[432] Weining Yang, Ninghui Li, Omar Chowdhury, Aiping Xiong, and Robert W Proctor. 2016. An empirical study of mnemonic sentence-based password generation strategies. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1216–1229.
[433] Yulong Yang, Gradeigh D Clark, Janne Lindqvist, and Antti Oulasvirta. 2016. Free-form gesture authentication in the wild. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3722–3735.
[434] Xin Yi, Shuning Zhang, Ziqi Pan, Louisa Shi, Fengyan Han, Yan Kong, Hewu Li, and Yuanchun Shi. 2023. Squeez'in: Private authentication on smartphones based on squeezing gestures. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–15.
[435] Linghan Zhang, Sheng Tan, and Jie Yang. 2017. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 57–71.
[436] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1080–1091.
[437] Zhiyi Zhang, Michał Król, Alberto Sonnino, Lixia Zhang, and Etienne Rivière. 2021. EL PASSO: efficient and lightweight privacy-preserving single sign on. *Proceedings on Privacy Enhancing Technologies* (2021).
[438] Benjamin Zi Hao Zhao, Hassan Jameel Asghar, and Mohamed Ali Kaafar. 2020. On the resilience of biometric authentication systems against random inputs. *arXiv preprint arXiv:2001.04056* (2020).
[439] Peng Zhao, Kaigui Bian, Tong Zhao, Xintong Song, Jung-Min Park, Xiaoming Li, Fan Ye, and Wei Yan. 2016. Understanding smartphone sensor and app data for enhancing the security of secret questions. *IEEE Transactions on Mobile Computing* 16, 2 (2016), 552–565.
[440] Ziming Zhao, Gail-Joon Ahn, Jeong-Jin Seo, and Hongxin Hu. 2013. On the security of picture gesture authentication. In *22nd USENIX Security Symposium (USENIX Security 13)*. 383–398.
[441] Yuchen Zhou and David Evans. 2014. SSOScan: automated testing of web applications for single Sign-On vulnerabilities. In *23rd USENIX Security Symposium (USENIX Security 14)*. 495–510.
[442] Bin B Zhu, Jeff Yan, Dongchen Wei, and Maowei Yang. 2014. Security analyses of click-based graphical passwords via image point memorability. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*.

1217–1231.
[443] Maximilian Zinkus, Tushar M Jois, and Matthew Green. 2021. SoK: Cryptographic confidentiality of data on mobile devices. *arXiv preprint arXiv:2109.11007* (2021).
[444] Moshe Zviran and William J Haga. 1990. User authentication by cognitive passwords: an empirical assessment. In *Proceedings of the 5th Jerusalem Conference on Information Technology, 1990.'Next Decade in Information Technology'*. IEEE, 137–144.

A The State of Web Authentication and Recovery

In order to consider suitable authentication schemes for E2EE contexts specifically we first survey all available authentication schemes to glean lessons for usability and survey. We broadly group contemporary authentication and recovery schemes into one of three categories, *primary*, *secondary*, and *recovery*, based on the context in which each is most widely deployed. Primary authentication mechanisms are the first (and often the only) step required for identity verification, while secondary authentication mechanisms are usually only triggered after the primary authentication process finishes correctly. Both primary and secondary authentication mechanisms are not mutually exclusive in real deployments–providers often allow users to choose among multiple secondary authentication mechanisms, or use one secondary factor as a fallback for another (e.g., using recovery codes in case of MFA app loss).

Recovery authentication mechanisms are by definition intended to be rarely used and may justifiably require more effort or hassle on the part of the user. We devote particular discussion to schemes relevant in E2EE recovery (namely § A.3.1 - § A.3.4).

A.1 Primary Authentication Mechanisms

A.1.1 Passwords. Security and usability issues with passwords are legion [64, 133, 134, 173, 296, 374]. Decades of academic research has shown that users constantly forget passwords [62, 363], use guessable passwords [61, 309, 397], reuse passwords across different accounts and providers as a coping mechanism [9, 93, 135, 184, 346, 407], and opt not to change their password even when notified of password reuse or insecurity [151, 402]. In the contemporary threat landscape, however, even widely recommended security practices such as increasing password strength would do little to protect against phishing attacks [134, 375] and large-scale data breaches [265, 284] even if users were to adopt best practices. These concerns impact users at all skill levels—academic work has found the relationship between technical expertise and vulnerability to common attacks (including susceptibility to phishing attacks, password reuse, and choosing stronger passwords) is largely inconclusive [231, 408].

In addition to public databases allowing users to check whether their credentials have been compromised [2, 3], industry has deployed various cryptographic techniques to automatically alert users of password reuse and breach, such as Meta's Private Data Lookup (PDL) tool using private set intersection to check whether a user's password is contained within a server-side set of passwords exposed in data breaches [165]. Unfortunately, academic work has repeatedly shown that the effectiveness of user notifications is limited: Only around a quarter of warnings resulted in users changing their password [151, 375]. Given the unavoidable tensions between security and usability in any password-based authentication scheme, passwords are increasingly viewed as a "legacy authentication mechanism" [166].

Password Managers: Password managers are a key mitigation strategy to make it easier for users to handle vast quantities of credentials. While the technical community favors password managers

Authentication Category	Papers
Passwords	[1, 12, 13, 23, 24, 30, 32, 49, 56–59, 61, 70, 73, 75, 86, 93, 99, 108, 112, 135, 139, 141, 147, 149, 151, 160, 167, 179–182, 184, 186, 193, 200, 215, 218, 219, 224, 230, 240, 243, 244, 251, 252, 256, 258, 260, 262, 273, 276–279, 289, 291–293, 296, 300, 302, 303, 309, 310, 313, 315, 316, 329, 336, 336, 338, 339, 344–346, 349, 350, 353, 357, 370, 373, 384–387, 391, 395–397, 397–400, 402, 407, 413, 427, 428, 432, 442]
Biometric	[13–15, 44, 78–80, 101–104, 123, 170, 177, 190, 209, 225, 246–249, 253, 257, 311, 337, 355, 372, 377, 390, 421, 424, 426, 429, 433–436, 438]
2FA	[10, 19, 33, 89, 90, 94, 94, 110, 120, 140, 142, 145, 146, 150, 233, 241, 242, 245, 272, 288, 318, 330–334, 351, 356, 364, 380, 383]
SSO	[48, 84, 105, 106, 124, 137, 143, 144, 227, 264, 358, 365, 404, 430, 437, 441]
Graphical Password	[11, 23, 67, 81, 82, 85, 161, 210, 279, 328, 359, 376, 382, 440]
RBA	[109, 138, 140, 255, 269, 270, 347, 414, 416, 417, 422]
Passwordless	[122, 191, 216, 217, 228, 232, 234, 261, 308, 379, 425]
Device PIN	[45, 83, 168, 266, 267, 290, 294, 394]
Recovery Questions	[26, 27, 62, 92, 162, 163, 282, 439]
Social Authentication	[65, 158, 192, 220, 321, 322, 342]
Other	[169, 176, 214]

Table 2: Literature search results for authentication and recovery mechanisms. A small number of works appear under multiple categories.

for security reasons as they allow users to opt for higher-entropy passwords and eliminates the need to memorize credentials, academic work has shown that users' primary motivation for use is convenience rather than security [32], with some users even consciously avoiding storing credentials for high-value accounts in a password manager even as they use it for credentials for less sensitive accounts [32]. Users' thought process when choosing which password manager to use also tends to be driven by financial cost (e.g., if one requires a subscription fee) rather than security [291]. In practice, users frequently do not use password managers to their maximal security advantage and largely use them to autofill low-entropy passwords [32]. Moreover, users are generally still required to remember a password for the password manager itself, or else the password to a third-party email service that can be used to authenticate to the credential manager [381].

Single Sign-On: Single-sign on (SSO) is a federated login technique that centralizes the responsibility for authenticating users with a single primary provider (most commonly Google or Apple [286]) using access delegation protocols such as OAuth and OpenID Connect. SSO adoption has been limited by both legitimate privacy considerations over data sharing with big tech companies [48, 106, 286, 368] and holdouts in adoption due to lack of trust in the underlying technology [366, 369], with prior work showing users are less likely to use SSO for more sensitive accounts [84]. The crux of the privacy issue is that the centralized provider (e.g., Google) will be able to observe all authentication attempts for a particular user, though there have been several promising academic proposals to reduce data sharing [124, 159, 204, 287]. Some platforms (such as GitHub) opted not to offer federated log-in to maintain greater control over the authentication process for their website [174].

There has also been a large body of academic work showing security vulnerabilities both in the underlying protocol [264, 264, 358, 367, 405] and deployed implementations [46, 143, 144, 178, 263, 365, 401, 404, 410, 430, 431, 441], including real-world cybercrime networks that maintain honeypot websites and collect OAuth access tokens [121]. Apart from specific security and privacy concerns, single sign-on schemes inherently present a single point of failure [63, 172] and hence an attractive target for attackers.

A.1.2 Decentralized Identities. In a real-world context, government-issued identity documents form the primary authentication scheme for most people. Despite their importance for all aspects of life, even though these documents are sometimes lost or misplaced, to the point where the U.S. government has a website devoted to replacing lost or stolen identification documents (including birth certificate and social security card) [388] given the frequency with which this occurs.

Given the resilience and replaceability of real-world identity documents, a commonly floated scheme in academic and government proposals is to digitize these documents and allow individuals to authenticate to web services (E2EE and non-E2EE) using their real-world identity [343]. One such scheme electronic identity scheme, eIDAS, has been deployed in the UK and European Union for several years [183]. In the US, this has become more widely discussed as some states have passed age verification laws requiring users to verify they are above a certain age prior to accessing a service [21]. The FIDO2 Credential Exchange Protocol discussed in § 4.3.4 is not specific to passkeys and could conceivably be repurposed to enable digital identity authentication, though this will require all to trust a third-party service to convert real-world eID documents

into some form of cryptographic identity. In addition to general privacy considerations and the ease of compromising these documents, another obstacle to deployment of these types of schemes is the lack of widespread eID ownership among the general public in some regions.

A.2 Secondary Authentication Mechanisms

Common second factor authentication (2FA) mechanisms used historically and still today include recovery questions, email and SMS-based MFA, 2FA authenticator apps, hardware tokens, and risk-based authentication (e.g., using browser metadata to flag suspicious login attempts).

A.2.1 Email and SMS 2FA. Email and SMS 2FA are still widely used for recovery today [68, 250], both as the primary 2FA mechanism and as fallback authentication strategies for more secure 2FA schemes (such as backups of 2FA authenticator apps [146]). We consider email and SMS 2FA jointly as academic work has shown them to be roughly equally usable in terms of recovery success rates and user perception [62, 233]. SMS has been the most widely deployed 2FA mechanism for at least a decade [28, 140, 288], the most common form of which is a time-based one-time password (TOTP). Both code-based and link-based 2FA are vulnerable to social engineering ("real-time phishing"), as users can often be tricked into sharing the code even when told not to do so by the companies. SMS-based authentication codes have well-documented security issues and are easily intercepted via SIM swapping attacks and attacks on the SS7 protocol [201, 241, 242, 268, 352], but there are nonetheless certain scenarios (e.g., low value accounts, a user posses only a shared email account) that may make SMS 2FA more suitable for an individual use case, and vice versa. SMS 2FA also typically reveals the code on the device lock screen, and in email 2FA some prominent websites (including Google and Facebook) would reveal the code in the email header or preheader [25].

Recovery: While end-users have historically considered email and SMS to be the most usable recovery options, it is nonetheless plausible that users may lose access to one of these factors—for instance, a user may list their university or corporate email as the recovery email for their primary personal email account, and later leave the university or company. Google reported in 2018 that 10% of users fail email or SMS 2FA [284] (e.g., if they no longer have access to the recovery email), though providers attempt to mitigate this by regularly reminding users which recovery options they have set.

A.2.2 Authenticator App. Mobile authenticator apps (e.g., Duo Mobile, Microsoft Authenticator) have seen low adoption among the general public [150, 284, 317] but are frequently mandated in university settings [10, 333] or by a small number of security-sensitive providers (e.g., Github [195]). Academic usability research has found that MFA apps are generally considered easy to use [10, 90, 110] but that users perceive the extra step required by MFA as a nuisance [10, 95, 100, 271, 333]. MFA apps are widely supported among the top domains [325].

Recovery from App Loss: After several years of widespread 2FA app deployment, the academic community has begun to investigate the consequences of app loss (which can frequently occur as a result of device reset, loss, or theft, or because the app backup not including as part of larger device cloud backup).

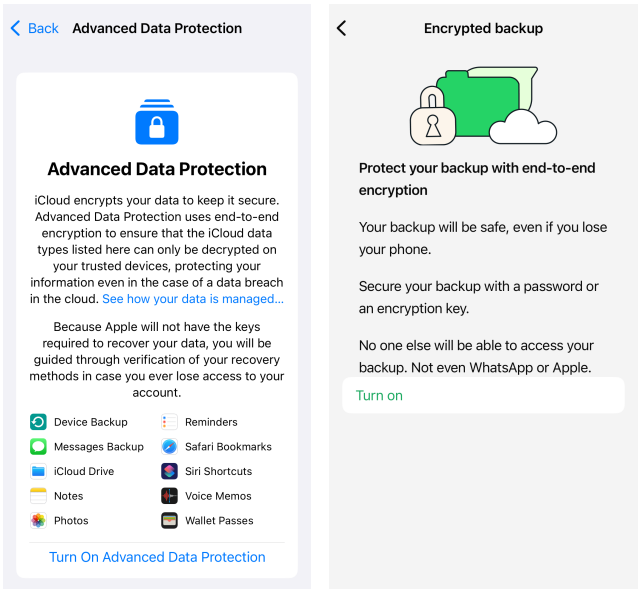

Figure 4: Onboarding screens for Apple's E2EE cloud storage service termed Advanced Data Protection (left) and WhatsApp's E2EE backup (right).

User concerns over device loss have been a frequent theme with app-based 2FA. Every widely used 2FA app offers different backup and recovery options [275]. Several offer cloud backups with iCloud, Google Drive, and sometimes other cloud services, and encrypts backups with a user-chosen password. Others use "backup codes" which the user is responsible for storing, and which are easily lost, or offer users the option of backing up to another device via QR code [146].

Prior work has found that recovery is a weak point for 2FA apps that undermines the security mobile app authentication is intended to provide by resorting to the usual array of fallback authentication mechanisms: SMS, email, passwords, and manual recovery [142, 146]. A spate of recent work has focused on the consequences of losing access to a two-factor authentication (2FA) mechanism, a common scenario for users who use a mobile app for 2FA and a recipe for disaster as mobile devices are regularly lost, changed, damaged, or stolen. Gerlitz et al. [142] studied how service providers respond to users losing a 2FA mechanism. In 2023, they created accounts at 78 popular websites or mobile apps using 2FA to see what recovery information, if any, was described to the user. They conclude that not enough attention is given to recovery, with 28 of 78 services studied not mentioning anything during the setup phase what backup or recovery procedures, if any, might exist. Amft et al. [33] went a step further and analyzed real-world deployments of multi-factor authentication on 71 websites, contacting the sites through public email addresses, support forms as though they were a user who had lost their second factor and

going through the manual account recovery process. They found significant variation and inconsistencies among sites, including discrepancies between a given site's documentation and actual procedures. Accounts on 10 of 71 sites were recovered simply by providing specific knowledge about the account that only the real account owner would know, while 13 sites required some form of official real-world identification, such as a government ID, to regain access. All in all, the authors identify 17 distinct recovery procedures and conclude that they "could not identify best practices regarding MFA recovery procedures" due to the variety.

Gilsenan et al. [146] studied backups of two-factor authentication apps that use time-based one-time passwords (e.g., a six-digit code that a user has 30 seconds to provide to a platform), finding commonly shared flaws in backup security, such as that the one-time password data is backed up in plaintext or that SMS is used to authenticate to the backup. The scope of all prior work in 2FA generally assumes two points: (1) that the user recalls or has access to the backup password and only the second factor is lost, and (2) that manual recovery is a possibility since the provider has the ability to reset or remove MFA.

A.2.3 Biometric Authentication. A commonly floated approach is for a user to provide some element of their real-world identity to the service provider, such as biometric data. While biometrics are certainly the most resilient form of user authentication, this form of data raises numerous questions around privacy and accuracy. As with recovery questions, biometrics are often used as a component of a multi-factor authentication process, such as unlocking a mobile device with Face ID when a user has also demonstrated physical possession of the device as in passkey authentication. However, with regular end-user hardware, we believe biometrics are not suitable as a standalone authentication factor for a cloud-based service. Consequently, we do not consider biometric data to be a viable authentication path at scale.

In the last few years, new attempts have been undertaken to bridge the gap to allow identifying individuals on a global scale. One example is the Orb technology that is being deployed as part of the Worldcoin project [420]. The project deploys proprietary hardware and algorithms in order to achieve unique global identification of individuals with very low false-positive and false-negative rates. One use-case is to provide strong "proof of personhood" (and therefore Sybil resistance) for Web3 services even where users do not have access to other means identification such as government issued IDs.

A.2.4 Hardware Tokens. The well-known weaknesses of authentication flows that rely solely on passwords has motivated the adoption of 2FA technology. However, SMS-based 2FA (Section A.2.1) remains vulnerable to SIM swapping and authenticator apps (Section A.2.2) do not protect against live phishing attacks. Hardware tokens that perform interactive cryptographic protocols with the online service address these issues.

Today, most hardware-based authentication protocols are based on the specifications provided by the FIDO Alliance, which many large companies are part of. The Universal Second Factor (U2F) protocol [128] allows for interoperable hardware tokens that can provide an authentication proof to different web services. This additional factor is typically only requested when the user logs in for the first time on a new device.

When registering at a new web service, the client (e.g. the browser) provides the hardware token with origin information that include the domain name. The hardware token then creates a fresh public-private key pair in its secure memory and derives a key handle based on the origin information. Both the public key and the key handle are then passed through the client to the web service.

For later authentication, the web service provides the key handle and a challenge to the client which passes it together with the origin information to the hardware token. The hardware token first verifies that the key handle and origin match. This prevents other web service from tricking the user to authenticating on a phishing website. In a second step, the hardware token signs the challenge with the stored private key which the website can verify using the public key.

Hardware tokens typically perform a simple test of user presence by requiring a simple button press on the device. This ensures that each authentication attempt is known to the user and a malicious app cannot perform authentication requests in the background. Web services can use the attestation keys provided by the hardware tokens to ensure that the user is using a device that fulfills certain certification standards. [128]

From a usability standpoint, academic work has shown repeated concerns over account lockout upon device loss [89, 94, 120, 334], in addition to general annoyance over the hassle of having to carry and retrieve an additional physical component, which is often reflected in high login timeout or cancellation rates [89, 333]. Difficulty of account sharing is another significant concern with hardware tokens [334].

A.2.5 Recovery Questions. First conceived of in 1990 [444], recovery questions ask users to answer a series of questions based on personal knowledge of the account owner, where the questions are usually determined by the provider but sometimes also user-generated. Frequently used historically, recovery questions are now widely disgraced as a viable authentication scheme after numerous studies showing that answers are low-entropy and often guessable by other individuals personally close to the account owner [202, 233, 326, 342], a result that has likely only gotten worse as users share troves of personal data online [222, 319]. To make matters worse, studies have also repeatedly found that some users provide untruthful answers as a means of improving their account security [62, 148, 233], but which has a side effect of making it more likely that the user themselves forgets the correct answer. Bonneau et al. [62] concluded back in 2015 that it is "next to impossible to find secret questions that are both secure and memorable".

Usage-Based Questions: A suggested variation to reduce the guessability of recovery questions is to use "dynamic" recovery questions in which the answer changes based on account and/or device usage patterns [27, 162, 164, 439], as compared with the traditional "static" recovery questions described above. For instance, a service provider may ask questions based on geolocation data [20, 163]. Prior work found that users were particularly worried that they wouldn't be able to recall the correct answers and might lock themselves out of their own account [27] as certain

types of data (particularly app installations) are easier to remember than others (e.g., SMS and call history) based on how frequently the data changes.

Authentication based on usage history raises particular concerns for at-risk demographics, such as older individuals who may struggle with memory recall, public figures whose location history is readily accessible, or domestic abuse victims where the attacker is generally familiar with location and communication history. In summary, we conclude that both static and dynamic recovery questions are unlikely to be suitable as either a standalone authentication mechanism or as a second factor for accounts.

A.2.6 Risk-Based Authentication. While not traditionally considered a second-factor since metadata is often unwittingly provided by the user, risk-based authentication (RBA) is a critical strategy providers deploy to distinguish legitimate logins from nefarious access [138, 416]. RBA broadly refers to any technical strategy that protects against illicit account access but does not require any information from the user, most commonly asking users to verify their email address [68]. Providers have long resorted to using metadata indicators to determine whether a recovery attempt is legitimate, though high failure rates of 2FA (even basic email and SMS 2FA) [284] require companies to carefully balance which logins should be considered suspicious. Broadly, services use an immense variety of mitigation strategies, including throttling (limiting the number of guessing attempts), previously-seen IP addresses, geolocation data, or other metadata available from browser fingerprints [109, 138, 156, 284, 347, 416, 422], and sometimes even monitoring users' behavior post-login [157]. Typically, additional verification steps are only deployed if an authentication attempt is deemed suspicious based on the particular metrics and statistical framework used by the account provider.

Both Apple and Google rely heavily on the concept of trusted devices, where logged-in devices are sent a notification to confirm any additional access attempts flagged as suspicious [265, 284], though this is only relevant when multiple devices are connected to an account. Temporal lockouts, where the user needs to wait a certain period after a correct authentication, are commonly deployed in recovery schemes. For instance, as early as 2008 Gmail only allowed account recovery through recovery questions if the account in question had not been accessed in the past five days [111]. Today, Google deploys temporal lockouts as part of the manual recovery process to ensure a legitimate user has a chance to deny the request [154], though research has shown users often fail to recognize malicious sign-in attempts [269].

RBA is almost always deployed as an additional authentication measure on top of a pre-existing primary authentication scheme, most commonly in addition to standard password-based authentication [414, 415]. Prior work has shown it is possible to evade many RBA measures using various cloaking techniques [34, 69, 255, 285, 304, 305].

A.3 Recovery Mechanisms

A.3.1 Social Authentication. Social authentication and recovery, where an account holder designates one or more recovery contacts or "trustees" who can help them regain access, is the only recovery scenario that can dependably handle various real-world disaster cases (where the user loses or forgets all devices and passwords). First proposed by Brainard et al. [65] of RSA in 2006, social recovery takes advantage of real-world trust relationships to authenticate a user by having a different user vouch for them. Brainard et al. motivated this concept of vouching by considering the financial cost to the company of password-reset staff, but in today's world social recovery is an important option because of the difficulty of authenticating the correct user as part of a manual recovery process, and because in some cases the provider may be unable to reauthorize the user (as in E2EE services), and has received renewed attention in recent cryptographic proposals [72].

Single Recovery Contact: The simplest social recovery scheme is to designate a single recovery contact (presumably a close acquaintance). LastPass, Bitwarden, and 1Password have all deployed some form of trusted contact recovery for a pre-designated user of the same password manager [5, 52, 237] (in Bitwarden only premium users are able to set an emergency access contact).

Apple's E2EE cloud backup scheme, introduced in 2022, also deploys the idea of a recovery contact, described by Apple as "a trusted friend or family member" [35]. In Apple's scheme, this contact is another iCloud user who generates a short code to send to an original user Alice enabling her to recover her account. In theory, a user with two iCloud accounts could also use one as the recovery contact for the other, though this arrangement would not adequately address the root concern. This is similar to a multi-wallet system in the cryptocurrency ecosystem, where a user leverages a secondary wallet to authenticate to the primary wallet [335]. Service providers can also allow a user to designate multiple recovery contacts, where each contact has the ability to single-handedly provide the account holder with a code to restore access.

While Apple's documentation [35] promises that a recovery contact "won't have any ability to access your account, only the ability to give you a code to help you recover your account", in practice a recovery contact could simply pretend to be the original user (assuming the contact knows a few additional basic facts, such as the iCloud account email used). Apple takes several precautions to prevent a recovery contact from gaining access to the account. Apple requires any individual entering the recovery code to answer an additional set of security questions in the first instance. Most importantly, Apple documentation suggests there is a time delay between the recovery request and regaining account access, and specifies that users should not use their devices in the intervening period because to do so would indicate that the request is not legitimate. In short, a user who is logged in and actively checking their devices and/or accounts will be able to detect that such a recovery process has been initiated. If the user has associated other traditional two-factor authentication mechanisms with the account, such as a phone number or email address with a different provider, they may also receive a notification on this platform informing them of the access request.

Threshold Social Recovery: As an alternative to designating individual users as an all-powerful recovery contact, service providers can also offer a threshold secret sharing scheme where the secret is divided among multiple recovery contacts but can be reassembled once a certain number of shares are combined [348]. In Shamir

secret sharing, for instance, a key is divided into n pieces with a recovery threshold k such that k pieces (where $k \leq n$) are needed to reassemble a valid secret. This has the benefit of making the recovery mechanism more resilient against unavailable or malicious recovery contacts by distributing trust among multiple contacts and providing redundancy in social recovery.

Schechter et al. [342] first proposed an account recovery scheme with multiple recovery contacts in 2009, where a key is split among the multiple contacts such that a minimum threshold of the total must share an account recovery code with the original user for them to regain access. Although Schechter et al. do not use the term Shamir secret and do not specify how the secret is divided up among the trustees, the scheme described functions in a similar manner to a standard cryptographic secret sharing scheme. Follow-on work has shown threshold trusted contact schemes are highly usable, with failures of trusted contact authentication due to time delays or timeouts (likely due to the perceived low value of the accounts used in experiments), rather than poor mental models or misconceptions [233, 342, 362].

Industry has already deployed secret-sharing schemes for recovering E2EE data. PreVeil, a cross-platform cloud service offering end-to-end encrypted email and file storage for enterprises, deployed an opt-in threshold scheme in 2019, stating that a system that distributes trust among a set of trustees is more secure than one which centralizes trust in a single trustee [323]. In PreVeil's design, when an account holder has initiated the recovery process, they will be presented with a list of their previously designated recovery contacts and able to select which members of the list should be used to approve the request. PreVeil's system architecture is somewhat unusual in that it does not use passwords or require the user to enter any credentials in order to log in. Instead, they store the user's private key on-device, allowing anyone authenticated to the device to access PreVeil storage. As a result, PreVeil needed a sufficiently failsafe backup mechanism in case the user loses their device(s) since no password or recovery code exists. Facebook used to offer a "Trusted Contacts" feature for regaining access to Facebook accounts (albeit not in an E2EE scenario) via a three-of-five secret sharing scheme [119], but deprecated the feature in 2022.

Social recovery has been gaining favor in the cryptocurrency ecosystem as well. In 2022 BitKey, a non-custodial hardware wallet (i.e., users store their own private keys), enabled an opt-in threshold social recovery scheme [52], where an account holder designates three recovery contacts and two of the three are needed to restore access. PreVeil does not require a certain threshold size, but similarly specifies in their documentation that two-of-three is a typical setup.

Limitations of Social Recovery: Social recovery goes a long way towards mitigating the challenge of an individual user managing their own keys, but at the same time presents several new concerns. While Apple takes several sensible precautions to prevent a recovery contact from gaining illicit access (instituting a time delay, requiring the individual requesting access to answer a series of additional security questions, etc.), this mode of recovery is nonetheless vulnerable in certain scenarios. We can reasonably assume that someone close enough to the account holder to be designated a recovery contact will likely be able to answer any additional verification information (including the email address associated with the iCloud account, date of birth, etc.), and therefore gain access to the account.

A time-delay between the access request and when access is granted, during which the account holder is notified that a request has occurred, is essential for mitigating illegitimate access. However, time-delay schemes rely heavily on users' attentiveness and assume that users check an account regularly, and recent academic work found that users often fail to recognize and respond to login attempt notifications [269]. Critically, the dependence on this proactive detection on the part of the user means that security guarantees of social recovery do not hold if the account owner has passed away. This is not a scenario most users contemplate for obvious reasons, but posthumous account access is fairly simple under Apple's individual recovery contact scheme. [1] In the case of a single recovery contact we must also consider an honest-but-curious recovery contact, such as a relative who initially requests access to recover family photos but later realizes the account also contains years of messaging history. A threshold secret sharing scheme would partly mitigate this scenario since multiple contacts would need to agree that access is acceptable.

Social recovery has also been exploited by online scammers. The process of receiving an unsolicited message from an acquaintance asking the recipient to click on a link and provide some information closely resembles real-world scams, a fact which malicious actors used to their advantage. Facebook's Trusted Contacts feature was the target of a popular scam in 2017 in which an attacker who has compromised a given Facebook account sent messages to the account owner's contacts, pretending to be the owner and asking the recipient to click on a link to help them reset their password by providing the message sender with a recovery code [18, 192]. Unfortunately for the victim, the link provided was in fact a password reset link for the recipient's account, and the code the recipient sent back to the attacker allowed the attacker to compromise the recipient's account as well. Shortly after this scam became widely publicized Facebook disabled trusted contacts as a recovery mechanism, though the company never officially provided a reason.

A more contemporary concern is that social verification may be vulnerable to manipulation by generative AI tools, such as a falsified video call or voicemail, with even close contacts unable to distinguish between genuine and artificial content. Simply speaking to another user over the phone was considered sufficient identification as part of a social recovery scheme as recently as 2016 [362], but there have been numerous voice- and video-cloning attacks in recent years used in real-world scams [15, 71, 171, 209]. Social authentication is all too easily susceptible to various social engineering attacks, such as where a contact calls from an unusual phone number and claims they have lost their smartphone and need assistance recovering an account—when in reality, the contact's voice is AI-generated.

A.3.2 Long-Term Recovery Key. There are several different terms for this concept ("recovery key", "recovery code", "master passphrase",

[1] Apple has a separate notion of a "Legacy Contact", an optional setting where a user's legacy contact can recover account access by manually presenting a death certificate to Apple—but social recovery can intentionally or unintentionally also become a legacy contact.

etc.), but they all refer to a pseudorandom string that the user presents to the service to restore access to their account. This recovery code is generally distinct from a standard user-generated password or PIN regularly used to log in in that it is arbitrary and usually substantially longer to guard against brute-force attacks.

The popularity of recovery codes as a recovery mechanism endures because they are generally the most secure and efficient way of recovering account access—provided a user stores the code in a safe place and does not lose it. Virtually every cloud backup option either requires users to use a recovery key of some sort or offers it as an option if their protocol allows for multiple recovery mechanisms: Apple iCloud lets users use a 28-character recovery key (in addition to the ordinary iCloud password) [37], and WhatsApp encrypts backups with either a 64-digit encryption key or a user-generated password [411]. Meta's Labyrinth protocol offers several different recovery mechanisms, one of which is a standard 40-character recovery code [118]. Signal does not support cloud backup but lets users encrypt a local backup using a 30-digit recovery key which the user is responsible for storing and safeguarding. These codes are generally only shown once upon creation, although a logged-in user can usually also generate a replacement recovery code even if they have lost the old one.

A recovery key option suffers from the same problem as a user-generated password: users all too often forget or lose it. A small number of users may neglect to save it at all, often out of overconfidence that they will never need it [176]. Unintentional loss is even more likely given that the nature of a recovery key is that it is used rarely, if ever. A user may store it on local device storage or on a physical piece of paper, encounter it many months later, and throw it out without realizing its significance. If the user's only backup recovery mechanism is this passcode (in addition to losing access to their device and/or regular password), they have no recourse and are locked out of their account permanently. The simplest mitigation technique is to create redundant copies of the passcode, though this increases the attack surface and potentially requires users to store the passcode where family members or others can access it.

A.3.3 Manual Recovery. Manual recovery ("ad-hoc schemes" [176]) are the recovery scheme of last resort [136, 142, 233, 312]. Some large industry providers offer a formally described manual recovery process [38, 153], while most others offer generic support contact information. Providers may also offer appeals processes in cases where a provider's content moderation scheme flags the account [206]. On a large scale, however, there is little incentive for small service providers to expend significant effort of these types of schemes, especially for users of unpaid services. Importantly, provider-assisted recovery is inherently not possible in E2EE services, which usability research has shown that some users do not understand, with users of an E2EE email service mentioning provider assistance as a possible recourse after recovery code loss [176].

A.3.4 Break-Glass Encryption. To develop an E2EE variant of manual recovery, a recent thread of academic work has attempted to tackle challenges around encrypted data loss by focusing on detecting, rather than outright preventing, account access [341]. Orsini et al. [307] proposed a cryptographic scheme for emergency access to cloud data storage, using the same threat model as in this work where a user Alice has lost all relevant credentials and all devices. They propose a credential-less authentication scheme in which any user can request access to a cloud account knowing only the associated email address or similar username, but there are only two possible states for a given account: either the legitimate user Alice is logged in and can monitor and reject illegitimate access requests within a certain timeframe, or Alice has become locked out of her account (e.g., by losing her device) and her request to regain access will be automatically granted after some time period has elapsed (since there is no legitimate user to reject it).

Such schemes are entirely dependent on detectability: the assumption is that the legitimate user will be consistently online, and confidentiality is guaranteed by proactive action on the part of the account owner. Both the Orsini et al. scheme and a similar concept for cryptocurrency wallets [55] assume an information asymmetry between the legitimate user and all other users in that the legitimate user would know when they have lost access (and request to be restored to the account) before anyone else, but this does not always hold (e.g., a device is stolen, posthumous access, etc.). Perhaps the biggest concern with these "break-glass encryption" schemes is that a deceased or incapacitated account holder is now vulnerable to any relatives or acquaintances familiar with their account name to a far greater extent than was the case with existing social recovery schemes. We are skeptical that any such scheme would ever be feasible for the general public.

B Survey of Authentication Literature

Having systematically reviewed authentication and recovery mechanisms used in E2EE services, we seek to better understand how real-world deployments compare with academic research. We compiled relevant literature on all authentication schemes found in contemporary web authentication (including smartphone authentication) by searching a range of keywords (including authentication, password, 2FA, MFA, recovery) across relevant academic venues. We search the four major security conferences (USENIX Security, IEEE S&P, NDSS, CCS) as well as other relevant conferences and workshops (PETS, SOUPS, CHI) and collect all papers published from 2012 (a year chosen to reflect changes in available authentication techniques since the comparative evaluation framework introduced by Bonneau et al. [63]) through June 2024.

Since there are numerous potential spelling variations (e.g., "authenticator" instead of "authentication", or "Multi-Factor Authentication" instead of "MFA"), to augment our keyword search results we additionally search through citations of a subset of seminal papers via Google Scholar, including highly-cited works published at a small number of additional venues, to ensure we capture the vast majority of relevant work. After manual inspection of titles and abstracts, we exclude papers on subjects adjacent but not directly related to one of the categories of end-user authentication schemes discussed in Section A (e.g., phishing, client-to-server authentication protocols). We further generally exclude academic proposals of novel authentication schemes unless the work contains a usability study or broadly applicable lessons for deployed schemes.

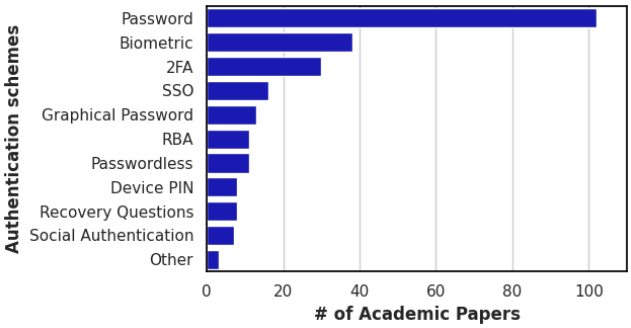

Figure 5: Literature survey of academic work on real-world authentication schemes.

B.1 Results

Our literature search resulted in 245 papers that we broadly categorize by authentication scheme as shown in Figure 5. Table 2 in the Appendix shows the specific papers included in each category. A small number ($n = 2$) papers were classified under more than one category, so the sum of the individual categories is slightly higher than the overall total.

Overall, we find that 41.6% of academic authentication research has focused on passwords, including password usability, password managers, measuring password reuse, and various other aspects. In particular, we find only 7 papers studying social authentication in some capacity (a scheme deployed in widely used E2EE services) and just 1 paper (categorized under 'Other') exploring the usability of E2EE recovery codes, even though both E2EE password managers and social authentication pose several unanswered usability and security questions (discussed in Section 8).