



An In-Depth Analysis of Password Managers and Two-Factor Authentication Tools

MOHAMMED JUBUR, Computer Science, Jazan University College of Engineering and Computer Science, Jazan, Saudi Arabia

PRAKASH SHRESTHA, Equifax Inc, Atlanta, United States

NITESH SAXENA, Texas A&M University College Station, College Station, United States

Passwords remain the primary authentication method in online services, a domain increasingly crucial in our digital age. However, passwords suffer from several well-documented security and usability issues. Addressing these concerns, password managers and two-factor authentication (2FA) have emerged as key solutions. This article examines these methods with a focus on enhancing password security without compromising usability. We utilize an adapted Bonneau et al. (IEEE S&P 2012) framework tailored to the specific challenges of password managers and 2FA. This allows us to categorize and evaluate prominent solutions from both academic research and industry practice, with a focus on their security, privacy, and usability. A crucial aspect of our study involves evaluating the effectiveness of a combined PM+2FA system in balancing security and usability. This study not only examines current trends but also suggests potential areas for future research, offering valuable insights to both users and developers in the evolving landscape of digital security.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Security and privacy** → **Authentication**; **Authentication mechanisms**; • **Human-centered computing** → *Usability*; *Human computer interaction (HCI)*;

Additional Key Words and Phrases: Authentication, password manager, two-factor authentication, security, usability, privacy

ACM Reference Format:

Mohammed Jubur, Prakash Shrestha, and Nitesh Saxena. 2025. An In-Depth Analysis of Password Managers and Two-Factor Authentication Tools. *ACM Comput. Surv.* 57, 5, Article 128 (January 2025), 32 pages. <https://doi.org/10.1145/3711117>

1 Introduction

Today's digital landscape is heavily reliant on Internet services such as online banking, e-commerce, e-government, and social networking, which are accessed by billions of users globally. A common thread among these services is their dependence on passwords for user authentication. Unfortunately, this reliance brings to the forefront several critical security and usability issues. A general user preference is to opt for weak or repetitive passwords across multiple accounts due

Authors' Contact Information: Mohammed Jubur, Computer Science, Jazan University College of Engineering and Computer Science, Jazan, Saudi Arabia; e-mail: mjabour@jazanu.edu.sa; Prakash Shrestha, Equifax Inc, Atlanta, Georgia, United States; e-mail: prakash.public@gmail.com; Nitesh Saxena, Texas A&M University College Station, College Station, Texas, United States; e-mail: saxena@tamu.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0360-0300/2025/01-ART128

<https://doi.org/10.1145/3711117>

to the challenge of remembering complex, unique passwords for each service [45, 54, 94]. This practice significantly heightens the risk of password compromise through various means, such as online guessing attacks, where attackers try to log in with the most common passwords, and offline guessing attacks, where attackers use stolen data to guess passwords without the risk of being locked out. Additionally, password reuse exacerbates the issue; a compromised password on one service could potentially grant unauthorized access to the user's accounts on other online services [48, 65].

In response to these challenges, password managers have emerged as a solution, offering users the ability to generate and manage strong, complex passwords for different online service accounts. These password managers are broadly categorized into three types, each with its operational method and security implications. Local password managers, such as those integrated into browsers (e.g., Google Chrome, Mozilla Firefox) or phone-based systems like Apple's Keychain, store passwords directly on the user's device. This approach enhances user convenience by providing easy access and offline availability of passwords. However, if the device is compromised, the passwords stored on it can also be vulnerable.

Online password managers, like LastPass and 1Password, store user passwords on remote servers. This method offers the advantage of accessibility from any device with an Internet connection. However, this central storage model also presents a risk. In the event of a server breach [80], encrypted passwords stored in these online vaults could be exposed, and sophisticated attackers might employ methods such as dictionary attacks to decrypt them. Cryptographic password managers, such as PwdHash [82] and Password Multiplier [57], take a unique approach by generating strong, service-specific passwords using cryptographic techniques. They typically combine a user's master password with additional factors (e.g., the service's domain name) to create a unique password for each service. This method significantly reduces the risk of password reuse across services but still hinges on the strength and secrecy of the master password. A notable limitation of cryptographic password managers is their restricted usability, especially in terms of browser compatibility. Unlike local and online password managers, which typically offer broad browser support, cryptographic managers often support only a limited set of browsers. This constraint can significantly impede user experience and accessibility, as users are confined to specific browsers to leverage the full benefits of these tools. Therefore, while cryptographic password managers excel in generating secure, unique passwords, their practicality is somewhat diminished due to these usability challenges.

Regardless of the category, password managers significantly reduce the cognitive load of managing multiple complex passwords, effectively addressing one of the most significant usability issues in password-based authentication. However, they introduce their own set of challenges and vulnerabilities. Centralized password storage, whether on a local device or online servers, can become a single point of failure. Furthermore, the continuous operation of these tools, especially online managers, raises privacy concerns. They have the potential to monitor and record users' browsing habits, creating comprehensive profiles of user activity. This continuous tracking can lead to traceability problems, where a user's actions across different services can be linked, undermining anonymity. Additionally, any implementation flaw could lead to unintended password leaks, exposing sensitive information across multiple services [68].

Given the inherent vulnerabilities of password-only authentication, **two-factor authentication (2FA)** has emerged as a crucial supplementary measure. 2FA enhances security by requiring a combination of two different authentication factors. These typically include "something you know" (e.g., passwords), "something you have" (e.g., software or hardware tokens), or "something you are" (e.g., biometric traits like fingerprints or facial recognition). This layered approach is

commonly implemented in various forms, including SMS-based methods and applications such as Google Authenticator and Duo Push, where a mobile device serves as a software token.

Adding a second authentication factor is designed to raise the security level during the login process substantially. However, this enhancement often comes at the cost of increased user effort. Traditional 2FA methods, such as those requiring **one-time pins (OTPs)**, necessitate user interaction with their mobile devices for code retrieval and entry into the authentication terminal.

Innovative low-effort 2FA solutions have been developed to address these usability concerns in both industry and academia. Low-effort 2FA methods, such as push notifications [19, 20] and audio-based authentications [62], aim to maintain the security benefits of 2FA while significantly reducing users' physical and cognitive load. For instance, push notifications only require a single tap to approve a login attempt, and audio-based 2FA uses ambient sounds to authenticate without user input. These solutions enhance usability by streamlining the authentication process and minimizing disruptions, making them more user-friendly.

This shift toward user-friendly 2FA solutions reflects a broader trend in digital security: the pursuit of an optimal balance between robust security measures and a seamless user experience. As technology continues to advance, refining 2FA methods remains a critical aspect of enhancing online security protocols.

Our research article aims to accomplish four primary objectives. First, we identify and discuss the leading password managers and 2FA schemes that are prominent in both academic and industry domains, including LastPass, 1Password, PwdHash, Password Multiplier, Google's 2FA, Duo, and others [6, 7, 21, 26, 39, 57, 62, 82].

Second, we delve into the intricacies of security, privacy, and usability aspects of these password managers and 2FA schemes. A particular focus is given to low-effort 2FA schemes, exploring how they enhance usability while potentially introducing hidden design vulnerabilities [86]. This part of the study critically evaluates these systems from multiple angles, including their security and usability attributes.

Our third objective involves a comprehensive analysis of the current and emerging trends in the field. We examine various research works [46, 47, 49, 68] to provide insights and future directions for research in this domain.

The analytical framework we use for evaluation is an extension of the one proposed by Bonneau et al. [43]. Originally designed for assessing general authentication schemes, we have tailored this framework to meet the specific and unique challenges posed by password managers and 2FA.

Last, we assess a scenario where both a password manager and 2FA are employed concurrently by users. This part of our study aims to evaluate the combined security and usability benefits of this integrated approach using our extended evaluation framework. While applied in nature, this research bridges foundational aspects of several fields, including usability, security, and privacy in password management and 2FA, machine learning, human-computer interaction, and cryptographic protocols.

2 Background

2.1 Password Managers

A password manager is a software application designed to enhance password security while alleviating the cognitive burden of managing numerous passwords for different online accounts. These applications generate unique, long, complex passwords for various online services and store them in secure, encrypted storage. Users can access their password vaults using a single master password, significantly streamlining the user experience. Examples include LastPass [26], Google Password Manager [28], and Password Hash [82].

Table 1. Features Offered by Deployed and Academic Password Managers

Scheme		Password Generation	Auto-Fill	Identifies Reused Password
Local	Google Chrome [28]	●	●	○
	Mozilla Firefox [29]	○	●	○
	Internet Explorer [30]	○	●	○
	Apple Keychain [8]	●	●	●
Online	1Password [6]	●	●	●
	Dashlane [9]	●	●	●
	LastPass [26]	●	●	●
	RoboForm [38]	●	●	●
Academic	PwdHash [82]	●	○	○
	Password Multiplier [57]	●	○	○
	Passpet [97]	●	○	○
	Amnesia [91]	●	○	○
	SPHINX [85]	●	○	○

● Offers the benefit; ○ Does not offer the benefit.

Password managers come in various forms, including cloud-based (online) solutions that offer accessibility across multiple devices, local managers that store information directly on the user's device for added privacy, and academic-developed managers that often introduce innovative security features. This variety caters to different user needs and preferences in password management. In today's digital age, the number of online accounts per individual has significantly increased, and password managers have become increasingly important. They simplify the management of numerous credentials and maintain security against the growing threats of data breaches and cyber-attacks. This evolution reflects the shifting landscape of digital security and the critical role of password managers in protecting personal and professional information.

2.1.1 Industrial (Deployed) Password Managers. Industrial password managers, widely utilized in real-world scenarios, specialize in generating complex and secure passwords. They encrypt passwords using a key derived from the user's master password. Storage is either on a remote server (Online-PM) or locally on the user's device (Local-PM).

Online-PMs like LastPass [26] and 1Password [6] offer multi-device syncing, password sharing, and remote vault access. These features enhance user convenience and security. RoboForm [38] provides both online and offline modes, using strong encryption algorithms like local and end-to-end 256-bit AES encryption to protect user data. Online-PMs often include features to prevent password reuse and provide secure password recovery options, addressing common security concerns.

Local-PMs, such as those integrated into Google Chrome [28] and Apple's Keychain [8], store encrypted passwords directly on the user's device. This approach offers enhanced security as data is not stored on remote servers, making them ideal for privacy-conscious users. These managers also typically include features to prevent password reuse and safeguard against unauthorized access.

For a detailed overview of the features offered by various deployed and academic password managers, refer to Table 1.

2.1.2 Academic Password Managers. Developed by researchers, these managers use mathematical functions to generate strong passwords from a master password, without storing them.

PwdHash [82] creates passwords by hashing the user's master password with the target service's domain name, generating a unique password for each service.

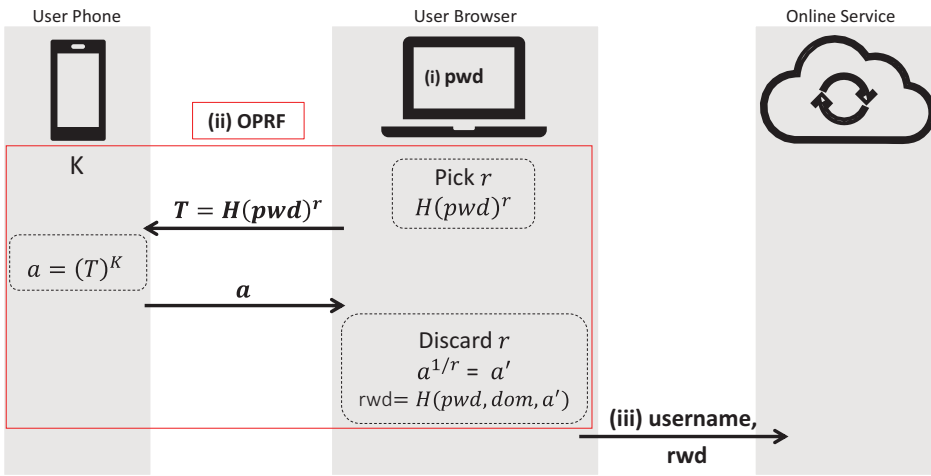


Fig. 1. SPHINX flow. (i) The user enters the master password (pwd), (ii) the device (i.e., the browser extension) and the client (i.e., the phone app) establish the Oblivious Pseudo-Random Function (OPRF) protocol to map the user pwd into randomized password (pwd), and (iii) the client sends pwd to the online service for authentication.

P-Multiplier [57] employs a two-step hashing process. Initially, it hashes the user’s master username and password iteratively, followed by a second hash that includes the domain name of the online service.

Passpet [97], *Amnesia* [91], and *SPHINX* [85] require a communication channel between a browser extension and a remote server or phone app. For instance, *Passpet* uses user-labeled credentials to generate strong passwords, which are stored in a remote server-based vault. *Amnesia* uses a unique 256-bit seed for each account added by the user to generate passwords. *SPHINX* combines a user’s memorable password with a service’s domain name and a key from the user’s phone to generate passwords. Figure 1 provides a comprehensive overview of its operational mechanism.

These academic password managers compute passwords in real time, ensuring that they are not stored on local machines or remote servers, enhancing security and privacy. Users can access online services with easy-to-remember passwords, which are used to generate complex and secure passwords.

2.2 Two-Factor Authentication

2FA [84], a significant enhancement in digital security, combines a password (“something you know”) with an additional security token. This token could be “something you have” like a software or hardware token, or “something you are” such as biometric data including fingerprints, facial recognition, or iris patterns. A typical 2FA process involves the user entering their password followed by a short, random, and one-time verification code from the token, often a phone. This dual-layer security model substantially increases account safety, as an attacker would need both the password and access to the second token.

2.2.1 Hardware Token Based 2FA. Hardware token based 2FA employs specialized devices like RSA SecureID [15] and security keys [36, 67] for second-factor authentication. Traditional hardware-based 2FA, known as *Hardware-OTP-2FA*, generates an OTP via protocols such as TOTP (time-based one-time password) [74] and HOTP (HMAC-based one-time password) [73]. The OTP, refreshed every 30 seconds, must be used within this time frame for successful authentication.

Modern hardware-based 2FA, in contrast, involves a small USB or wireless device connected to the authentication terminal during the authentication process. This is referred to as a *security key*. Devices like Yubico U2F [18] and Google Titan [35] fall into this category. They employ public key cryptography, integrating OTPs and the web service URL for authentication. The registration phase for each web service involves the creation of a key handle and a key pair, with the public key and key handle stored by the service. During authentication, after the initial user credential verification, the web server challenges the security key via the user's terminal. The user then activates the security key, which computes a response and sends it back to the server for final verification. Appendix Figure 5 illustrates the workflow of this authentication process using a security key.

In addition to traditional hardware tokens, recent advancements like Google's built-in security key on Android phones [37] have emerged, enabling the use of smartphones as security key devices. This implementation, however, is currently limited to Android versions above 7.0 and primarily supports Google services.

While hardware token based 2FA methods offer robust security, they have notable drawbacks in terms of usability, security, and privacy. From a usability perspective, hardware token based 2FA requires users to carry a physical device, which can be cumbersome and inconvenient. Users must ensure that they have the correct token for each service, and losing or forgetting the token can disrupt access. Additionally, some users, particularly those with disabilities, may find it challenging to use these tokens effectively. Although hardware tokens are generally secure, they are not immune to certain types of attacks. Physical theft of the token can enable unauthorized access if the attacker also has the user's credentials. There is also a potential risk if the token's internal secrets are compromised, either through sophisticated attacks on the manufacturing process or vulnerabilities in the device's firmware. The use of hardware tokens also raises privacy concerns, particularly regarding the traceability of user activities. The requirement to use the token with multiple services can create a linkable pattern of behavior. Additionally, some hardware tokens and security keys store a key handle and public key with the service provider, potentially exposing users to privacy risks if the provider's data is breached.

2.2.2 Traditional Software Token Based 2FA. Traditional software token based 2FA improves system deployability by utilizing software applications installed on general-purpose devices, such as smartphones, for secure login. This approach is commonly referred to as *software token based 2FA*. It typically involves either a built-in text messaging application (SMS-OTP-2FA) or a specialized software application (Software-OTP-2FA).

In the *SMS-OTP-2FA* scheme, the server sends a short, random OTP to the user's phone via the SMS messaging application. The user then enters this OTP at the authentication terminal to complete the process. This scheme primarily leverages the HOTP protocol to generate OTPs and depends on the telecommunication network for OTP delivery. It is widely used in financial institutions for services such as online banking login or transaction approval.

Software-OTP-2FA requires users to install a specialized 2FA application on their smartphones. Notable examples include Duo Mobile [7], Google 2 Step Verification [21], and Microsoft Authenticator [24]. During the authentication process, this application generates a short, random OTP, which the user supplies alongside their credentials. The registration of these apps with web services involves scanning a QR code that contains a secret key. This registration is a one-time process, binding the 2FA app with the service. Like Hardware-OTP-2FA, the OTP refreshes every 30 seconds and must be used within this time frame. Software-OTP-2FA utilizes both TOTP and HOTP protocols for OTP generation.

Despite their scalability and cost-effectiveness, traditional software-based 2FA approaches involve significant user interaction, particularly in copying the OTP during authentication. This

requirement can impact the usability and adoption of these systems. Recognizing this challenge, researchers and developers are focusing on Low-Effort-2FA schemes, aiming to simplify the 2FA process to ideally just password entry, thereby reducing user burden.

2.2.3 Low-Effort Software Token Based 2FA. Low-Effort-2FA schemes, such as Duo Push [20] and Google Prompt [19], leverage push notification services to reduce user effort, commonly referred to as *Push-2FA*. These schemes streamline the authentication process by minimizing user interactions compared to traditional methods.

In Push-2FA, the user first provides their credentials to the web service. Upon successful validation, a push notification is sent to the user's phone, the second-factor device. This notification includes details of the login attempt, such as the location and device information (e.g., operating system, IP address, browser). The user can then approve or deny the login directly from the notification, making the process more user-friendly and efficient. Appendix Figure 6 illustrates the workflow of Push-2FA, providing a visual representation of this process. Services like Amazon, Facebook, and Google have integrated Push-2FA into their native apps, allowing for seamless authentication experiences. This integration signifies a shift toward more user-centric security measures in online services.

Audio-Proximity Detection in 2FA. Several innovative Low-Effort-2FA schemes employ audio-proximity detection to verify the presence of the second-factor device. These include the following:

- *Sound-Proof* [62]: Sound-Proof utilizes ambient sound to confirm the proximity between the phone and the login terminal. It requires no action other than typing in the password, as proximity alone is sufficient for authentication. The process compares ambient sounds captured by the phone and the login terminal to determine if they are in the same environment. Appendix Figure 4 illustrates the workflow of the Sound-Proof authentication process.
- *Listening-Watch* [87]: Listening-Watch enhances Sound-Proof's approach by playing a random code in human speech, with the watch and browser capturing this sound. The similarity of these recordings, along with the presence of the code, validates the login attempt.
- *SlickLogin* [3]: SlickLogin uses near-ultrasounds to transfer OTPs from the terminal to the phone, streamlining the authentication process.
- *Proximity-Proof* [58]: Proximity-Proof employs ultrasounds for secure transmission of 2FA responses and uses device fingerprints to counteract man-in-the-middle attacks, although it raises privacy concerns due to the requirement of storing device fingerprints.
- *SoundAuth* [92], *Watermelon* [71], and *SoundLogin* [33]: Methods such as SoundAuth, Watermelon, and SoundLogin leverage audio for secure authentication, each with its unique mechanism and benefits.

While these Low-Effort-2FA methods significantly enhance user convenience, they may face challenges such as environmental noise interference or limited device compatibility. Moreover, the reliance on specific hardware (e.g., watches or smartphones with certain features) may not be universally accessible. Despite these potential limitations, the evolution of Low-Effort-2FA reflects a growing emphasis on balancing security with user experience in digital authentication.

3 Properties

The seminal work of Bonneau et al. [43] introduced a robust framework comprising 25 distinct properties, termed as *benefits*, for the analytical assessment of security, usability, and deployability aspects in authentication schemes. Recognizing the evolving landscape of digital security and the unique intricacies of password managers and 2FA systems, we extend this framework to address these specific contexts. This extension is pivotal in providing a more tailored and relevant

evaluation, given the distinct challenges and operational paradigms presented by modern authentication methods.

In the following sections, we delineate all the properties (or benefits) considered in our expanded evaluation framework. Properties marked with an asterisk (*) represent our novel contributions, signifying adaptations or additions to the original framework of Bonneau et al. These enhancements are designed to capture the unique nuances and emerging needs in the realm of password management and 2FA, reflecting the current trends and challenges in digital authentication systems.

3.1 Usability Properties

Exploring usability properties, we examine aspects fundamental to user interaction with password managers and 2FA systems. From memory effort to physical ease, these properties are key to practical, everyday usage. Let us delve into each property and its contribution to a user-friendly experience.

Memorywise-Effortless.* This property assesses if a scheme requires the user to remember any secrets, such as passwords or codes. Ideal for systems with multiple accounts, a *Memorywise-Effortless* scheme relieves the user from this cognitive burden by generating or retrieving secrets as needed. A scheme is *Quasi-Memorywise-Effortless* if it necessitates remembering just one universal secret for accessing multiple services, simplifying password management significantly.

Scalable-for-Users. This property evaluates the scheme's ability to handle an increasing number of accounts without adding to the user's cognitive load. A scheme is *Scalable-for-Users* if it can manage multiple accounts efficiently, maintaining ease of use and quick access irrespective of account quantity.

Nothing-to-Carry. This property indicates whether a scheme requires the user to carry any physical object for authentication. Ideal for ease of use, a *Nothing-to-Carry* scheme ensures authentication can be completed without additional items. *Quasi-Nothing-to-Carry* applies to systems utilizing everyday devices like smartphones or smartwatches.

Physically-Effortless. A scheme is considered *Physically-Effortless* if it requires no physical effort from the user. Systems demanding minimal actions, such as pressing a button or tapping on a screen, are *Quasi-Physically-Effortless*. This property is crucial for ensuring the system's accessibility and ease of use.

Easy-to-Learn. This property is crucial in determining how intuitively a new user can understand and operate the scheme. It should be designed such that even individuals with minimal technical background can learn to use it without significant difficulty. The scheme should be straightforward enough to allow users from diverse backgrounds and varying levels of tech-savviness to quickly grasp and remember the operational process, reducing the learning curve and making the system inclusive and accessible.

Efficient-to-Use. Efficiency is key in the day-to-day operation of a scheme. This property assesses whether users can complete tasks like password retrieval or authentication steps quickly and with minimal hassle. The goal is to design a system that streamlines regular tasks, making them as time-efficient as possible while maintaining security. An efficient system contributes to better user satisfaction by reducing the time and effort required for frequent operations, thus making the system more appealing and practical for regular use.

*For the 2FA scheme, we consider only the second factor of 2FA (and ignore the first-factor (password)) for the purpose of our evaluation. Given this, since none of the 2FA schemes considered in our study require the user to remember any secret, we discard the property when evaluating 2FA schemes.

Infrequent-Errors. This property evaluates the scheme's error tolerance, ensuring that legitimate users can successfully complete tasks without frequent rejections or errors. A user-friendly scheme minimizes the likelihood of incorrect denials, thereby enhancing reliability.

Easy-Recovery-from-Loss. This property focuses on the ease with which a user can recover or reset lost or forgotten access tokens. The scheme should provide straightforward methods for recovery, such as backup codes or secondary authentication mechanisms, to ensure uninterrupted access and maintain security.

Auto-Generation.* This property highlights the scheme's capability to automatically generate complex and secure secrets (like passwords) for the user. An *Auto-Generation* feature is particularly valuable as it relieves users from the burden of creating strong, unique passwords for each service, thereby significantly enhancing both security and convenience. The *Quasi-Auto-Generation* designation is given to schemes offering flexibility for users to either choose their own secret or use the one generated by the system, catering to diverse user preferences and needs.

Connection-less.* A *Connection-less* scheme operates independently of an Internet or mobile network connection, highlighting its ability to function solely on local devices. This feature is particularly relevant for ensuring continuous access to authentication systems regardless of network availability.

Each of these properties plays a vital role in defining the usability of password managers and 2FA systems. By thoroughly evaluating these aspects, we can show the effectiveness and user-friendliness of various authentication schemes.

3.2 Deployability Properties

Focusing on deployability properties, we assess factors crucial for the practical implementation of password managers and 2FA systems in various environments. We look at adaptability, cost, and compatibility to understand their integration into digital ecosystems. Now, let us explore each property and its role in effective deployment.

Accessible. This property ensures that any user, regardless of disabilities or physical conditions, can use the scheme with the same ease as using textual passwords. Accessibility is paramount in making authentication systems inclusive and can be achieved through adaptive technologies or design considerations that accommodate a broad spectrum of user abilities.

Negligible-Cost-per-User. The total cost for each user, encompassing both service provider and consumer ends, is negligible with this scheme. This includes installation costs at the service provider end and any required hardware or software at the consumer end. Cost-effectiveness is crucial, especially when deploying systems at a large scale, where costs can vary based on complexity and user base size.

Server-Compatible. A scheme is *Server-Compatible* if it requires no changes to the existing setup of the service provider. For 2FA schemes, which typically necessitate server modifications, this property is challenging to achieve. Understanding the types of changes required, such as additional security protocols or database modifications, is essential for assessing this property.

Browser-Compatible. This property signifies a scheme's compatibility with any web browser, without necessitating additional software. It achieves a high degree of flexibility and user convenience. A scheme is *Quasi-Browser-Compatible* if it functions on a select set of browsers or requires widely used plugins, like Flash. Examples include systems that work seamlessly across mainstream browsers but may require common extensions.

Client-Compatible.¹ This property ensures that the scheme is usable across various client devices like desktop PCs, laptops, mobile phones, and tablets. This cross-platform compatibility is key to accommodating users' diverse device preferences and usage scenarios.

Mature. A scheme is considered *mature* when it has seen extensive deployment and usage beyond research, with substantial real-world testing. Factors contributing to maturity include open source code availability, adoption by entities beyond the original implementers, substantial related literature, and recognition by standard communities. For instance, a widely adopted password management tool that has undergone rigorous user testing and extensive documentation would exemplify a mature scheme.

Non-Proprietary. A scheme is *non-proprietary* if it can be implemented and used freely, without royalty fees. This openness encourages broader implementation and innovation, allowing for widespread use and adaptation. A well-known open source authentication protocol, used and adapted by various organizations without licensing constraints, serves as an example of a non-proprietary scheme.

3.3 Security and Threat Properties

Turning to security and threats properties, we scrutinize elements central to the robustness of password managers and 2FA systems. This section dissects how these systems handle potential threats and vulnerabilities. Let us examine each property to understand its defensive mechanisms.

Resilient-to-Physical-Observation. This property ensures that observing the user during security tasks, such as authentication in 2FA or password retrieval, does not lead to scheme compromise. A scheme is deemed *Quasi-Resilient-to-Physical-Observation* if repeated observations (more than 10 times) might lead to a breach. For example, a system employing one-time tokens or behavioral biometrics could effectively resist observation-based threats.

Resilient-to-Targeted-Impersonation. This property signifies the scheme's robustness against impersonation attempts leveraging personal details like age or profession. A well-designed system will use multi-factor authentication methods that go beyond basic personal information, thwarting targeted impersonation attacks.

Resilient-to-Guessing. A scheme with this property is impervious to attackers' attempts at predicting or guessing the secret, even with brute force or dictionary attacks. Techniques enhancing this resilience include implementing high entropy in secret generation and enforcing complex password policies to prevent predictable patterns.

Resilient-to-Internal-Observation. This property ensures protection against internal threats like keylogging malware or clear-text communication interception [44, 83]. A scheme is *Quasi-Resilient-to-Internal-Observation* if it can withstand repeated internal observations, but may be vulnerable after extensive exposure. Mechanisms like end-to-end encryption and secure input fields can fortify a system against such internal surveillance methods.

Resilient-to-Leaks-from-Other-Verifiers. This property determines the scheme's ability to maintain its integrity even if another service provider experiences a data breach. It implies that the scheme's design is such that leaked information from one verifier does not compromise or weaken the security at another service, ensuring isolated security domains.

Resilient-to-Phishing. This property assesses the scheme's inherent ability to resist successful exploitation from phishing attacks. It indicates that even if an attacker deceives a user into revealing credentials on a fraudulent site, the scheme itself has safeguards that prevent these stolen credentials from granting access to the legitimate service. This resilience could be due to factors intrinsic to the scheme's design, such as unique session tokens or contextual user verification that phishing attempts cannot replicate.

Resilient-to-Theft. This property evaluates the scheme's security in scenarios where an authentication device or object is stolen. A scheme that is resilient to theft ensures that unauthorized access

is prevented even if a device involved in the authentication process falls into the wrong hands, typically through multi-layered security measures or requiring additional verification beyond mere possession of the device.

An attacker who gains possession of an object involved in the authentication or password management process (e.g., a mobile phone, a user's terminal) cannot impersonate the user. We grant *Quasi-Resilient-to-Theft* if the scheme is hard, but not impossible, to break by stealing the object involved in the authentication or password management process. For instance, to compromise a traditional 2FA scheme, the attacker needs to learn the user's password, gain possession of the second-factor device, the phone, and unlock the device, which seems hard to perform to compromise the user's account.

No-Trusted-Third-Party.¹ This property emphasizes the scheme's operational independence from external third-party services for authentication or password management processes. By design, it ensures that sensitive user information is not exposed to or vulnerable to third-party entities, thereby enhancing user privacy and security. This approach implicitly supports the principles of user consent for data sharing and data minimization, as it avoids unnecessary data exposure and reliance on external parties.

Requiring-Explicit-Consent.* This property mandates active user participation in the authentication process. It ensures that critical actions, such as logging in or filling in credentials, are initiated or confirmed only with the user's explicit consent, typically through a user action. This feature is intrinsic to the scheme, designed to prevent unauthorized or automated access.

Separation-between-Terminal-and-Device. Specifically applicable to 2FA systems, this property highlights the requirement for the authentication terminal and the second-factor device (like a mobile phone) to be distinct and separate entities. This separation is a fundamental security feature of 2FA, ensuring that the authentication process is not downgraded to a single factor, especially in scenarios where a single device attempts to serve both roles.

3.4 Privacy Properties

Finally, in privacy properties, we focus on ensuring user data protection and confidentiality in password managers and 2FA systems. This exploration looks into how these technologies shield user identity and sensitive information. Let us analyze each property to understand its role in preserving privacy.

Unlinkable. This property ensures that even if two service providers collaborate, they cannot determine whether the same user is using both services. It is essential in password managers and 2FA systems, as it means that the authentication methods used do not leave identifiable traces that can be linked across different services. This not only protects user identity but also ensures their activities are untraceable, aligning with the concept of anonymity in digital interactions.

Resilient-to-Leaks-to-Actual-Service. This property signifies that the service provider learns only the minimum necessary information about the user required for the authentication process. It is vital for maintaining user privacy, as it minimizes the risk of sensitive data being mishandled or leaked. It aligns with the principle of data minimization by ensuring that no additional user data is collected or accessible beyond what is essential for the functioning of the scheme.

4 Evaluation of Password Managers

In this section, we provide an analytical comparison of various password managers (described in Section 2.1) based on our extended framework, which includes a unique set of usability, deployability, security, and privacy properties (detailed in Section 3).

¹The property belongs to both the security and privacy aspects of the scheme.

4.1 Online Password Managers

Popular online password managers like LastPass, 1Password, and RoboForm require users to install a software application, typically as a browser extension or a mobile app. To use these password managers, a user needs to provide a master username and password. Due to this, they are *Quasi-Memorywise-Effortless*, as a single master password grants access to all stored passwords. These systems are also *Quasi-Nothing-to-Carry* since access is through commonly used devices without needing specialized hardware. The ability to reset credentials in case of loss makes these password managers *Easy-Recover-from-Loss*. Users have the choice between using passwords generated by the manager or creating their own, leading to *Quasi-Auto-Generation*. However, they are *not Connection-less*, as they depend on network connectivity and an always-on remote server.

Furthermore, these password managers are generally *Scalable-for-Users*. They can handle numerous accounts efficiently, which is crucial for users managing multiple online identities. The setup process, although varying among these managers, is designed to be *Easy-to-Learn*, accommodating users with different levels of technical expertise. In terms of *Physical-Effortlessness*, these systems require minimal physical interaction, typically limited to simple clicks or taps. The password managers are also *Efficient-to-Use*, allowing users to quickly retrieve or auto-fill passwords, thereby saving time in daily operations.

Last, these password managers aim to minimize errors, making them *Infrequent-Errors*. They are designed to accurately auto-fill credentials and provide reliable access to legitimate users, enhancing the overall user experience.

Several common factors underpin the deployability of LastPass, 1Password, and RoboForm. All three password managers offer browser compatibility, making them easily accessible across various web browsers without additional software. Their models balance free and premium features, resulting in *Negligible-Cost-per-User* for basic services. This strategy enhances their accessibility, appealing to a wide range of users. Overall, these factors contribute to the widespread adoption and practicality of LastPass, 1Password, and RoboForm in diverse digital environments.

In analyzing LastPass, 1Password, and RoboForm, we observe specific security characteristics based on their encryption and authentication mechanisms. LastPass and RoboForm rely on the strength of the master password for encryption and authentication, making them *weak* in *Resilient-to-Physical-Observation* and *Resilient-to-Internal-Observation*. The compromise of the master password in these managers could lead to access to all stored passwords (e.g., [77, 88]). Their ability to withstand *Resilient-to-Targeted-Impersonation* attacks depends largely on the unpredictability and strength of the user-generated master password.

Conversely, 1Password's additional layer of security using the secret key, along with the master password, significantly enhances its security. This dual-layer approach makes 1Password *very strong* in *Resilient-to-Physical-Observation*, offering robust protection against various forms of cyber threats, including targeted impersonation (e.g., [68]).

The 2022 security breaches experienced by LastPass [80], where unauthorized access to the company's development environment was gained, raise important considerations about the security of password managers. Despite assurances that customer data remained secure, these incidents highlight the potential vulnerabilities in such systems. This emphasizes the need for ongoing vigilance in security practices relevant to LastPass and all password management tools.

Concerning the *Resilient-to-Phishing* property, all of these password managers demonstrate strong resilience due to their feature of filling in credentials only when the login form's URL matches the one stored in the manager's database. However, there are multiple risks. First, users might be tricked into entering their master password or other credentials into a fake version of the password manager. Second, the risk remains where users can manually access and copy passwords

from the manager, potentially exposing them to phishing attacks if they input these credentials into a fraudulent website. Both scenarios underscore the importance of user awareness and caution alongside the password managers' technical security measures.

Exploring the security features of LastPass, 1Password, and RoboForm reveals various levels of defense against cyber threats. Each of these password managers demonstrates *Resilient-to-Theft* due to their master password requirement, ensuring robust protection against unauthorized access, even in cases where the physical device is compromised. The implementation of the auto-fill feature in these managers, as studied by Oesch and Ruoti [77] and Silver et al. [88], however, introduces vulnerabilities to network injection attacks, especially on HTTP sites, rendering them *Quasi-Requiring-Explicit-Consent*. In terms of *Resilient-to-Leaks-from-Other-Verifiers*, these managers effectively prevent data breaches in other services from impacting their stored data, safeguarding the integrity of encrypted information. Although these password managers are designed without reliance on external third-party services, aligning with the *No-Trusted-Third-Party* property, their dependency on proprietary servers for syncing introduces potential privacy concerns. The *Requiring-Explicit-Consent* property is partially met, with some automated features like auto-fill possibly operating without direct user intervention.

When assessing the privacy aspect, specifically the *Unlinkable* property of LastPass, 1Password, and RoboForm, certain nuances emerge. Each of these password managers requires the client application to request stored passwords from their respective databases whenever a user needs to access a service. This process potentially allows the password manager provider to track which services the user accesses and when, potentially using the user's device fingerprint [51]. Such data collection practices could impact user privacy, enabling service providers to compile user activity logs. Furthermore, while implementing robust encryption standards, these password managers are not immune to government surveillance or requests. Authorities may compel these services to disclose user data under specific legal circumstances [2, 4], potentially including sensitive information like user passwords or activity logs.

Consequently, LastPass, 1Password, and RoboForm face challenges in fully achieving the *Unlinkable* privacy property. While they secure user data through encryption, their operational model inherently involves some level of user data exposure to the service provider, thus hindering their ability to offer complete unlinkability and privacy assurance in the context of third-party and government access.

4.2 Local Password Managers

As we transition to examining local password managers, our focus shifts to popular options such as Google Password Manager (Chrome), Mozilla Firefox, and Apple Keychain. These local managers, while sharing some usability, deployability, and security properties with their online counterparts, present unique characteristics and challenges. This introductory analysis aims to highlight these differences and set the groundwork for a more in-depth evaluation.

It is important to note that a key feature of these local managers is their synchronization ability, similar to online managers, which enhances usability across various devices. However, when this synchronization feature is enabled, they function akin to online password managers. Our primary analysis focuses on their core functionality as local password managers, where data is stored and managed directly on the user's device. This fundamental approach to data storage and management significantly influences their deployability and security features, distinguishing them from online password managers.

Local password managers, being integrated within browsers or operating systems, offer advantages such as offline access and minimal or no cost per user. However, they also face unique

challenges like restricted cross-browser compatibility and specific security vulnerabilities inherent to local data storage.

In our analysis of local password managers, we find that many usability, deployability, and security properties align with those of online password managers. However, there are key distinctions that set local managers apart. This section will delve into these aspects, providing a comprehensive understanding of the capabilities and limitations of local password managers in the contemporary digital landscape:

- *Connection-less*: Unlike online managers, local password managers store passwords directly on the user’s device, eliminating the need for network connectivity to retrieve passwords. This enhances their convenience and reliability in offline scenarios.
- *Negligible-Cost-per-User*: Local password managers are typically built into browsers and operating systems, thus incurring no additional cost for the user. This built-in nature makes them inherently accessible without requiring external app or extension installations.
- *Browser compatibility*: An important aspect of local password managers is their compatibility with different browsers and operating systems. For instance, Google Password Manager and Apple Keychain exhibit *Quasi-Browser-Compatible* characteristics. They are functional across various applications and browsers that operate on Android and Apple OS, respectively. This cross-platform compatibility significantly enhances their usability in diverse environments. However, Mozilla Firefox’s password manager does not exhibit the same level of flexibility. It is specifically designed to work within the Firefox browser, making it *not Browser-Compatible* in the broader sense.
- *Privacy*: Local password managers provide notable *privacy* advantages. Since these managers store and manage passwords locally on the user’s device, retrieving passwords does not involve transmitting information about the user’s website visits or timings externally. This local data handling offers a distinct privacy benefit compared to online password managers, where user activity and password access might be tracked externally.

These local managers particularly excel in the property of being *Unlinkable*. Without transmitting user activity or password data to external servers, it becomes challenging for external entities to track or link a user’s activities across different services. This feature greatly enhances user privacy, protecting personal information against unauthorized tracking and external access.

Additionally, these local password managers inherently embody the *No-Trusted-Third-Party* characteristic. They do not require syncing with external servers, as they operate independently and rely on local data storage for their basic functionality. This independence from external entities ensures that the user’s sensitive data is confined to their own device, bolstering the security and privacy of their digital identity. The absence of third-party reliance minimizes risks associated with data breaches or surveillance, providing users with a secure and private digital environment.

However, local password managers have their own set of limitations. They are not *Resilient-to-Internal-Observation* and *Resilient-to-Theft* as their online counterparts. Unlike online password managers, which often incorporate additional layers of security, local password managers primarily depend on the security of the user’s device. In terms of *Resilient-to-Internal-Observation*, local password managers are vulnerable. If malware or a keylogger infiltrates the user’s device, it could potentially access and compromise the locally stored passwords [59]. This risk is inherent in the nature of local storage, where the device’s security is the primary line of defense.

Furthermore, these local managers are also less *Resilient-to-Theft*. If the physical device storing the passwords is stolen and the device’s security is breached (e.g., bypassing a lock screen

[56, 75, 76] or exploiting a security vulnerability), the thief could gain access to all stored passwords. This vulnerability underscores the importance of robust physical device security and vigilant personal device management when using local password managers. Additionally, local password managers do not offer an *Easy-Recovery-from-Loss* benefit. In local password managers, the passwords are stored locally on the user's device. If the device is lost, damaged, or compromised, the passwords stored on it could be irretrievably lost. This highlights a significant risk in terms of data recovery, making it crucial for users to have additional backup mechanisms or be aware of this inherent limitation.

Moreover, a study by Karole et al. [64], which compared online and local password managers (including portable USB and phone-based systems), found that participants often prefer local managers despite their lower usability compared to online managers. This preference is attributed to the perceived security and privacy benefits of storing passwords locally. In summary, while local password managers like Google Password Manager, Mozilla Firefox, and Apple Keychain share several attributes with online managers, they stand out due to their connection-less operation, built-in nature, and enhanced privacy. However, this comes at the cost of reduced resilience to certain security threats and limited browser compatibility.

4.3 Academic Password Managers

PwdHash, P-Multiplier, SPHINX, and Amnesia share a common goal: enabling users to generate strong and complex passwords. Additionally, these password managers require activation through specific user actions, such as pressing F2 or entering “@@” in the password field, as seen in SPHINX and PwdHash. Despite these similarities, their implementations differ significantly. We anticipate that these differences will be reflected in their usability and deployability benefits. Unlike these schemes, Passpet employs a user interface for password generation, setting it apart in its approach.

PwdHash, P-Multiplier, SPHINX, and Amnesia are categorized as *Quasi-Memorywise-Effortless*, requiring users to remember a master password. In contrast, Passpet demands users recall specific labels assigned to each service, thus not qualifying as *Memorywise-Effortless*. These schemes excel in scalability, allowing users to utilize a single master password across various services seamlessly. Being browser extensions, they are *Nothing-to-Carry*, eliminating the need for any additional physical devices. These schemes are inherently *Physically-Effortless*, *Easy-to-Learn*, and offer *Easy-Recovery-from-Loss*. Users interact with them similarly to regular passwords, although activation via specific key combinations (e.g., “@@”) is required.

However, SPHINX and Amnesia differ slightly, necessitating user response to phone notifications for password generation, hence classified as *Quasi-Physically-Effortless*. A core functionality of these schemes is *Auto-Generation* of complex and robust passwords. SPHINX, Amnesia, and Passpet require a communication channel between the user's phone and a remote server, contrasting with PwdHash and P-Multiplier, which operate *Connection-less*.

Studies by Chiasson et al. [46] and Shirvanian et al. [85] on the usability of PwdHash, P-Multiplier, and SPHINX revealed a notable issue: participants often forgot to activate these schemes, leading to incorrect usage. This frequent user error challenges their classification as *Infrequent-Errors*. As for browser compatibility, PwdHash, P-Multiplier, SPHINX, Amnesia, and Passpet currently support a limited range of browsers. For example, SPHINX is compatible only with Google Chrome, whereas PwdHash, P-Multiplier, and Passpet work with Mozilla Firefox, making them *Quasi-Browser-Compatible*. Moreover, these schemes are *Server-Compatible* since their usage does not require server-side modifications.

The security of PwdHash is predominantly dependent on the master password and the domain name of the service. This dependency exposes PwdHash to potential compromise through guessing attacks. Consequently, PwdHash is categorized as *Quasi-Resilient-to-Physical-Observation*,

Table 2. Analysis of Various Password Managers Using Our Evaluation Framework

Category	Scheme	Usability						Deployability						Security				Privacy									
		Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Easy-Recovery-from-Loss	Infrequent-Errors	Auto-Generate*	Connection-less*	Accessible	Negligible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Guessing*	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-Trusted-Third-Party	Requiring-Explicit-Consent*	Unlinkable
Deployed	LastPass	●	●	●	●	●	●	●	○	●	●	●	●	●	●	○	●	○	●	●	●	●	○	●	○	○	○
	1Password	●	●	●	●	●	●	●	○	●	●	●	●	●	●	○	●	○	●	●	●	●	○	●	○	○	○
	RoboForm	●	●	●	●	●	●	●	○	●	●	●	●	●	●	○	●	○	●	●	●	●	○	●	○	○	○
	Google Password Manager	●	●	●	●	●	○	●	○	●	●	●	●	●	●	○	●	○	●	○	○	○	○	○	○	○	○
	Mozilla Firefox	●	●	●	●	●	○	●	○	●	●	●	○	●	●	○	●	○	○	○	○	○	○	○	○	○	○
Academic	Apple Keychain	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	PwdHash [82]	●	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	P-Multiplier [57]	●	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	SPHINX [85]	●	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Amnesia [91]	●	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Passpet [97]	○	●	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	

● Offers the benefit; ● Almost offers the benefit; ○ Does not offer the benefit.

Note: *Almost offers the benefit* indicates that the scheme fulfills most but not all criteria for the benefit, often due to minor limitations or specific conditions where the benefit may not be fully realized.

Quasi-Resilient-to-Targeted-Impersonation, *Quasi-Resilient-to-Guessing*, and *Quasi-Resilient-to-Leaks-from-Other-Verifiers*.

In contrast, SPHINX, Amnesia, P-Multiplier, and Passpet demonstrate stronger resilience. They are *Resilient-to-Physical-Observation*, *Resilient-to-Targeted-Impersonation*, *Resilient-to-Guessing*, and *Resilient-to-Leaks-from-Other-Verifiers*. However, compromising SPHINX and Amnesia requires access to the user’s phone to acquire the secret used for password generation, making them somewhat vulnerable.

P-Multiplier and Passpet, although robust, are *Quasi-Resilient-to-Internal-Observation*. An attacker could potentially intercept the master password and username, disrupting the Internet connection during password generation. Nevertheless, these schemes offer strong *Resilient-to-Theft* properties, as accessing user passwords necessitates knowledge of the master password.

All of these schemes operate independently, without involving any trusted third parties, enhancing their security posture. Additionally, they mandate explicit user consent for activation, usually through a specific key combination, thus adhering to the *Require-Explicit-Consent* property.

4.4 Summary of Evaluation of Password Managers

Our analysis of various password managers, including online, local, and academic options, reveals a diverse range of benefits and limitations across usability, deployability, security, and privacy properties.

Usability. Online password managers like LastPass, 1Password, and RoboForm provide high scalability and ease of use, although they require a master password, which can be a single point of failure. Local password managers, such as Google Password Manager, Mozilla Firefox, and Apple Keychain, offer user-friendly interfaces with offline accessibility, reducing dependency on network connectivity. Academic password managers, including PwdHash, P-Multiplier, SPHINX, and Amnesia, are efficient in auto-generating passwords but often require specific user actions for activation, potentially impacting the user experience.

Security. Online password managers face challenges in resilience to physical and internal observation due to their dependency on a master password. However, they exhibit strong resilience against phishing and targeted impersonation attacks. Local password managers enhance security through offline storage, reducing exposure to online threats, but are less resilient to device theft and internal observation. Academic password managers show varied resilience depending on their design. For instance, SPHINX and Amnesia, which rely on additional devices, offer higher security but may reduce usability.

Deployability. Online password managers are typically easy to deploy, as they are available as browser extensions or mobile apps compatible with multiple platforms. Local password managers are inherently integrated into browsers or operating systems, providing seamless deployment without the need for additional installations. Academic password managers vary in deployability; while some like PwdHash and P-Multiplier can be used as browser extensions, others may require more complex setups involving additional devices or user actions.

Privacy. Local password managers provide enhanced privacy by storing passwords locally, thus minimizing data transmission and exposure to third parties. Online password managers rely on remote servers, which can lead to privacy concerns related to data breaches and unauthorized access. Academic password managers, designed to operate independently without trusted third parties, generally offer strong privacy protections.

These detailed comparisons are summarized in Table 2, providing a comprehensive overview of how these password managers perform across the evaluated properties, highlighting both their strengths and areas where they fall short.

5 Evaluation of 2FA Schemes

In this section, we evaluate each of 2FA schemes presented in Section 2.2 based on the properties defined in Section 3.

5.1 Hardware Token-OTP Based 2FA

This category of 2FA schemes requires the user to carry a hardware token for each service. Some of the widely used services, such as online banking (National Bank of Abu Dhabi, Commonwealth Bank of Australia, Bank of America, etc.), provide such hardware tokens to their customers. However, they are expensive to deploy, as services having multiple accounts need to provide several hardware tokens to the user. Thus, they are neither *Scalable-for-Users* nor do they feature *Nothing-to-Carry*. Further, they do not feature *Easy-Recovery-from-Loss* because the token must be removed and replaced with a new token if the hardware token is lost or stolen. In these schemes, since the user needs to get the right token that ties to the service and supply the PIN generated on the token within a short period of time (usually within 30 seconds), they are *Quasi-Efficient-to-Use* and incur *Quasi-Infrequent-Errors*.

Hardware token based 2FA schemes face accessibility challenges, particularly for visually impaired users, due to their reliance on visual displays for OTPs, resulting in the inability to read OTPs generated on the hardware token's screen. Additionally, the need for a separate hardware token for each service further complicates their use, as visually impaired users may struggle to differentiate between multiple tokens. This significantly limits the practicality of hardware token based 2FA schemes for a segment of users, underscoring an important area for accessibility improvement. Moreover, these schemes are not *Negligible-Cost-per-User*. The service provider must distribute a token to each user, which can be a considerable expense, especially for services with a large user base. Despite these drawbacks, hardware token based 2FA schemes excel in other aspects of deployability.

Fortunately, hardware token based 2FA schemes are *Resilient-to-Physical-Observation*, *Resilient-to-Targeted-Impersonation*, and *Resilient-to-Guessing*. The hardware token is a specialized device designed to generate and present an OTP to the user via its display. Unlike general-purpose devices (e.g., smartphones), it is challenging for the attacker to install malware on such special-purpose devices, and hence make them *Resilient-to-Internal-Observation* and *Resilient-to-Phishing*. They are *Resilient-to-Leaks-from-Other-Verifiers* because each hardware token is tied to each service (via a shared secret). Further, the PIN expires shortly, generally after 30 seconds. However, they are not *Resilient-to-Theft*. Once the attacker has gained physical access to the hardware token, he can easily login on behalf of the victim user (assuming the attacker has already obtained the victim's credentials). The OTP generated on the hardware token is based on a secret (i.e., a seed) that the issuer or the manufacturer assigns. If the attacker somehow obtained this seed, such as through the leakage from the manufacturer itself, and is well aware of the algorithm used to generate the OTP, the attacker may be able to generate the OTP and impersonate the user. Thus, these schemes do not offer *No-Trusted-Third-Party* feature. They require explicit consent from the user, as the user typically needs to press a button to generate the PIN. Further, the hardware token is a specialized and independent device, and it offers *Separation-between-Terminal-and-Device*.

Since these schemes require the user to own one hardware token per service, are hardware device based, and the user needs to have one such device for each service, it makes the scheme *Unlinkable* and *Resilient-to-Leaks-to-the-Actual-Service*.

5.2 SMS-OTP Based 2FA

To employ the SMS-OTP based 2FA scheme, each online service implements its own SMS-OTP scheme. Most implementations use either four or six digit verification PINs, which are sent to the user's phone via a text messaging application. The message's header (embedded with OTP) sent to the user is typically tied to the service name. Since most people typically possess a phone and a single phone (or phone number) can be used for multiple services, the SMS-OTP based scheme features *Quasi-Nothing-to-Carry* and is *Scalable-for-Users*.

In our analysis, we consider the primary setup for the SMS-OTP. While this setup is neither *Physically-Effortless* nor *Efficient-to-Use*, as it requires the user to reach out to his phone, unlock the device, and copy the OTP to the authentication terminal, the evolving smartphone technology has brought significant changes. Many modern smartphones now offer an auto-fill feature for SMS-OTPs. This feature automatically fills the OTP received via text message into the authentication form on the same device, making the process both *Physically-Effortless* and *Efficient-to-Use*. The manual task of copying the OTP from one device to another is eliminated, streamlining the authentication process within the smartphone itself. This integration has significantly enhanced the user experience by making the authentication process faster and more user-friendly.

However, it is important to note that not all users may have access to smartphones with these auto-fill capabilities. Therefore, the primary setup of SMS-OTP based 2FA schemes, which requires manual input of the OTP, remains relevant in our analysis. Despite the advancements in smartphone technology, this primary setup continues to influence the overall assessment of the SMS-OTP's usability. Since the OTP in the SMS-OTP based 2FA scheme is generated automatically by the service, it offers the *Auto-Generation* feature.

The SMS-OTP based 2FA scheme is favorable in terms of deployment, as the scheme provides all of the benefits under deployment. It is the most widely used 2FA scheme by online services because it is open source and easy to implement. The user does not need to install any phone app or pay additional costs to use this scheme.

The SMS-OTP based 2FA scheme exhibits strong *Resilient-to-Physical-Observation* and *Resilient-to-Guessing*. This is because each authentication attempt generates a new, unique OTP, which is sent directly to the user's phone. This OTP is only valid for a short duration and changes with each login attempt, making it extremely difficult for an attacker to guess or use a previously observed OTP for unauthorized access [49].

Unfortunately, the scheme is vulnerable to phone number porting scams [1, 12]. Attackers can intercept or redirect text messages intended for the user by employing the SS7 hack [5] or spoofing cell phone towers [14]. Furthermore, Reaves et al. [79] found that some services fail to send completely random codes for each message. As a result, the SMS-OTP scheme is neither *Resilient-to-Targeted-Impersonation* nor features *No-Trusted-Third-Party*.

Additionally, this scheme is not *Resilient-to-Internal-Observation*. Malware installed on the user's phone can capture snapshots of text messages or log user inputs, posing a significant security risk [66, 83]. However, since the OTP is tied to the service, the scheme is *Resilient-to-Leaks-from-Other-Verifiers* and *Resilient-to-Phishing*. The reliance on the user's phone number, which can be stolen via phone number porting scams, makes the scheme not *Resilient-to-Theft*. Furthermore, when the phone is used as an authenticating terminal, there is no *Separation-between-Terminal-and-Device*.

The scheme relies on third-party services (e.g., mobile network providers) to send the OTP to the user. While this third party has access to the user's phone number, it does not learn other private information about the users, thus making the scheme *Unlinkable* and *Resilient-to-Leaks-to-the-Actual-Service*.

5.3 Software Token Based 2FA

Software token based 2FA schemes, widely used in various online services, are recognized as *Scalable-for-Users* and offer *Quasi-Nothing-to-Carry* benefits. These schemes leverage general-purpose smartphone devices, enabling users to access multiple services with a single device. However, the essentiality of transferring the OTP from the smartphone to the authentication terminal within a standard 30-second time frame renders these schemes *Quasi-Efficient-to-Use* and prone to *Quasi-Infrequent-Errors*. The recovery process in software token based 2FA schemes is quasi-easy. Users can replace an old secret key with a new one, typically provided as a QR code, thus facilitating *Quasi-Easy-Recovery-from-Loss*. On the smartphone app, the user can view the OTP, its timer, and the corresponding service logo. As the OTP generation is local and automated, these schemes are *Connection-less* and feature *Auto-Generation*.

In terms of deployability, software token based 2FA schemes excel across different platforms. Most services offer apps compatible with Android, Apple iOS, and ensuring broad accessibility and user convenience.

Security-wise, software token based 2FA schemes are similar to hardware token based 2FA in being *Resilient-to-Physical-Observation*, *Resilient-to-Targeted-Impersonation*, and *Resilient-to-Guessing*. However, they are only *Quasi-Resilient-to-Internal-Observation* due to the potential risk of malware on the smartphone compromising the secret key. The schemes' resilience to theft is conditional on the device's security measures; hence, they are *Quasi-Resilient-to-Theft*.

To utilize these schemes, users must unlock their smartphone, launch the OTP app, and transfer the OTP to the authentication terminal, making them *Requiring-Explicit-Consent*. However, a notable limitation is the absence of *Separation-between-Terminal-and-Device* when the phone serves as both the authenticating terminal and the second-factor device.

In conclusion, software token based 2FA schemes strike a balance between ease of use and security. They provide significant benefits in terms of scalability and deployment, but their effectiveness

is closely tied to the security of the smartphone and the user's active management of the authentication process.

5.4 Security Key Based 2FA

Security key based 2FA offers a scalable and cost-effective solution for users, allowing a single key to be used across multiple services. However, it necessitates carrying the physical key, thus not qualifying as a *Nothing-to-Carry* scheme. Users find this scheme physically effortless and efficient to use, with low error rates due to the simple action of pressing a button on the key. Most service providers recommend having an alternative authentication method, like SMS-OTP, due to the potential loss of the security key, indicating a limitation in *Easy-Recovery-from-Loss*.

A notable aspect of the security key is its limited compatibility. Our tests across various browsers and devices revealed that while it works well with browsers like Google Chrome and Mozilla Firefox, it faces compatibility issues with others, such as Microsoft Edge, and certain devices like iPhones and iPads.

The security key's reliance on a challenge-response cryptographic protocol imparts robustness against targeted impersonation and guessing attacks. The key generates a unique cryptographic pair for each service, enhancing its *Resilient-to-Leaks-from-Other-Verifiers* and *Resilient-to-Phishing* properties. However, its *Quasi-Resilience-to-Internal-Observation* becomes evident if the user's device is compromised, as demonstrated in the work of Bui et al. [44]. The attacker, having access to the user's device, can observe the usage patterns and potentially breach the system. The scheme's vulnerability to theft arises when the physical key is stolen and the attacker circumvents any device-level security like PINs or biometrics.

The security key's unique pairing protocol ensures *Unlinkable* and *Resilient-to-Leaks-to-the-Actual-Service*, protecting user privacy and preventing tracking of authentication activities across different accounts.

While the security key based 2FA is a step forward in authentication security, its effectiveness is contingent on the physical security of the key and the integrity of the user's device. This scheme balances ease of use with strong security features but has limitations in device compatibility and potential vulnerability to device-level breaches.

5.5 Push Notification Based 2FA

In the context of 2FA, Push-2FA schemes, utilized by services like Duo Push [20], Google [19], Last-Pass [27], and Facebook [52], present a unique blend of convenience and security. These schemes are inherently scalable, as a single smartphone or connected smartwatch can serve multiple online services, thus being *Scalable-for-User* and *Quasi-Nothing-to-Carry*. The ease of simply approving a login notification on a device makes Push-2FA *Physically-Effortless*, *Easy-to-Learn*, *Efficient-to-Use*, and prone to *Infrequent-Errors*. Additionally, the *Easy-Recovery-from-Loss* attribute is significant, as users can swiftly transition to a new smartphone and continue utilizing the service without undue hurdles. However, their dependency on Internet connectivity for cloud messaging services like Firebase Cloud Messaging [55], Apple Push Notification [40], and Windows Notification Service [72] renders them non-*Connection-less*.

Deployment-wise, Push-2FA excels in most aspects. However, it faces limitations in terms of accessibility. The visual nature of notifications can be a barrier for blind users despite the fact that some systems, such as Braille displays, provide conversions. This limitation calls for more innovative solutions to make Push-2FA schemes universally accessible, thus currently making them *Quasi-Accessible*.

Security-wise, Push-2FA schemes demonstrate robustness against physical observation, targeted impersonation, and guessing attacks, primarily due to their reliance on unique, user-specific

OTPs and ownership-based authentication methods. However, the resilience of these schemes to internal observation is only partial (quasi-resilient). This vulnerability is exemplified by malware on the device, potentially compromising the secret key. Additionally, the user habituation effect can lead to inadvertent responses to attacker-generated notifications, a risk further highlighted in the study by Jubur et al. [60]. Their work demonstrated how concurrent malicious notifications could deceive users, underlining the necessity for heightened user vigilance.

Despite their strengths, Push-2FA schemes show *Quasi-Resilience-to-Theft*. Unauthorized access could potentially occur if an attacker gains physical access to the device and bypasses its security measures. The study by Jubur et al. [60] contrasts the Push-2FA approach, which focuses on generating a single malicious notification during an active attack, making it stealthier and more challenging for users to detect. This nuanced approach to generating concurrent attacks adds more sophistication and risk to the Push-2FA methodology.

Regarding phishing and leaks from other verifiers, Push-2FA schemes maintain resilience due to the unique key pairing between the service and the user's device. However, the inherent risks associated with the potential compromise of the device and the habituation effect on users call for continuous improvement and user education to maintain the effectiveness of Push-2FA systems.

A critical aspect of Push-2FA is its dependence on third-party services for delivering notifications, which can pose privacy concerns. Services like Duo [7], Authy [11], and RSA [32] necessitate access to sensitive user information, like device details and location, potentially compromising privacy. This reliance on third-party services contradicts the *No-Trusted-Third-Party*, *Unlinkable*, and *Resilient-to-Leaks-to-the-Actual-Service* principles, highlighting a need for more privacy-preserving mechanisms in Push-2FA implementations.

While Push-2FA schemes offer significant user convenience and security benefits, they also present challenges in accessibility, privacy, and reliance on third-party services. Addressing these challenges is crucial for making Push-2FA a more universally applicable and secure 2FA solution.

5.6 Low-Effort Audio-Based 2FA

All audio-based low-effort 2FA schemes are *Scalable-for-Users* because the user can use his phone with multiple online services. However, each service may require the user to install its respective app on the user's phone. Similar to software token based 2FA, since these schemes can be used with phones and/or the watch, they feature *Quasi-Nothing-to-Carry*. As the name suggests, they are *Physically-Effortless* and *Efficient-to-Use*—they do not require any physical effort from the user during the authentication process (except for typing in the password). They are *Quasi-Infrequent-Errors* because they require the second-factor device to be close to the authenticating terminal (e.g., to transmit the OTP via an audio channel or capture the keystroke sounds). Since these schemes trigger the recordings on the phone and/or the authentication terminal without any user involvement, they feature *Auto-Generation*. They are not *Connection-less* because the user's phone requires an Internet connection to communicate with the authentication server.

Similar to prior schemes, since audio-based schemes leverage general-purpose mobile phone and/or wrist-worn wearable devices and authenticate the user with minimal *Physical-Effort*, they are *Accessible* and involve *Negligible-Cost-per-User*. Many users have started using tablets (e.g., iPads). Since this tablet browser is not as powerful as other client browsers (i.e., PS and laptop), as they do not allow installation of plugins or extensions, they are not *Client-Compatible*. These schemes are *Non-Proprietary* because they are academic and open to the public.

SlickLogin, Proximity-Proof, SoundAuth, Typing-Proof, Ultrasonic-Watch, and Watermelon are not *Resilient-to-Physical-Observation* and *Resilient-to-Targeted-Impersonation* because the proximity attacker can defeat these schemes. Sound-Proof is neither *Resilient-to-Physical-Observation* nor *Resilient-to-Targeted-Impersonation* [86]. The attacker who knows the victim's phone number can

Table 3. Comparison of Various 2FA Schemes Using Our Evaluation Framework

Scheme	Usability										Deployability				Security							Privacy					
	Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Easy-Recovery-from-Loss	Infrequent-Errors	Auto-Generation*	Connection-less*	Accessible	Negligible-Cost-per-User	Client-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Guessing	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-Trusted-Third-Party	Requiring-Explicit-Consent*	Separation-between-Terminal-and-Device*	Unlinkable	Resilient-to-Leaks-to-the-Actual-Service*
SMS-OTP [10]	●	●	○	○	○	○	○	○	○	●	●	●	●	●	●	○	●	○	●	●	●	○	○	●	●	●	●
Hardware token-OTP [31]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Software token [24]	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Security key [23]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Push-2FA [20]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
SlickLogin [3]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Proximity-Proof [58]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sound-Proof [62]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
SoundAuth [92]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Typing-Proof [69]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ultrasonic-Watch [99]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Watermelon [71]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

● Offers the benefit; ○ Almost offers the benefit; ○ Does not offer the benefit.

manipulate the victim’s background noise by making the user’s phone create a predictable sound through a simple call or notification. Assuming the attacker has already obtained the victim’s credentials, he can compromise the security of Sound-Proof. These schemes are *Resilient-to-Guessing*, *Resilient-to-Phishing*, *Resilient-to-Theft*, and *Resilient-to-Leaks-from-Other-Verifiers*. These schemes are neither resilient to man-in-the-browser [66] nor *Resilient-to-Internal-Observation*. Since these schemes are often transparent where the user only needs to supply his password, they do not *Require-Explicit-Consent* from the user. They offer *Separation-between-Terminal-and-Device* because they require an additional device (i.e., smartphone) besides the authentication terminal.

Proximity-Proof, Sound-Proof, SoundAuth, and Watermelon are not *Resilient-to-Leaks-to-the-Actual-Service*. Proximity-Proof requires the user to register his fingerprint to the service provider. Sound-Proof, SoundAuth, and Watermelon require recording the user’s ambient sound through the phone and/or the browser. Given this, they may capture and leak the personal and sensitive information, compromising the privacy of the user.

5.7 Summary of the Evaluation of 2FA Schemes

To summarize, our evaluation of various 2FA schemes, including the hardware token-OTP, SMS-OTP, software token, security key, push notification, and low-effort audio-based 2FA, reveals a wide range of benefits and limitations across usability, deployability, security, and privacy properties.

Usability. Hardware token-OTP schemes are less user-friendly due to the need for multiple tokens and manual PIN entry. SMS-OTP schemes, while commonly used, can be cumbersome without smartphone auto-fill features. Software token based 2FA and security key based 2FA are more user-friendly with easy-to-use apps and simple key presses, respectively. Push notification based 2FA schemes are highly user-friendly, requiring minimal user interaction. Low-effort audio-based 2FA schemes also provide ease of use but can be limited by environmental noise conditions.

Deployability. SMS-OTP schemes are the most widely deployed due to their simplicity and low implementation cost. Software token based 2FA schemes are also easy to deploy with broad platform compatibility. Hardware token-OTP schemes are less scalable due to the need for distributing physical tokens. Security key based 2FA schemes face compatibility issues with certain browsers and devices. Push notification based 2FA schemes are generally easy to deploy but depend on reliable Internet connectivity. Low-effort audio-based 2FA schemes require specific app installations, which can limit their deployability.

Security. Hardware token-OTP schemes provide strong security but are vulnerable to physical theft. SMS-OTP schemes are susceptible to phone number porting scams and malware but offer good resistance to guessing and phishing. Software token based 2FA schemes balance ease of use with security but are vulnerable to malware on smartphones. Security key based 2FA schemes offer robust security through unique cryptographic pairs but are vulnerable if the physical key is stolen. Push notification based 2FA schemes provide strong security against various attacks but are vulnerable to habituation effects and malware. Low-effort audio-based 2FA schemes offer strong security against guessing and phishing but are less resilient to physical observation and targeted impersonation.

Privacy. Hardware token-OTP and software token based 2FA schemes generally provide good privacy by limiting data transmission. SMS-OTP and push notification based 2FA schemes depend on third-party services, which can raise privacy concerns. Security key based 2FA schemes offer strong privacy protections with *Unlinkable* and resilient-to-leaks features. Low-effort audio-based 2FA schemes, while providing good privacy in most aspects, can potentially leak sensitive information through audio recordings.

These detailed comparisons are summarized in Table 3, providing a comprehensive overview of how these 2FA schemes perform across the evaluated properties, highlighting both their strengths and areas where they fall short.

6 Discussion

This section provides an in-depth analysis of our findings from evaluating various password managers and 2FA schemes. We discuss the balance between security and usability for these systems, address the security implications of the “Remember Me” feature, and explore the benefits of combining password managers with 2FA for enhanced security. By synthesizing these insights, we aim to present a comprehensive view of the current authentication landscape and suggest future directions for improving these systems.

6.1 Summary

Password managers significantly enhance both the security and usability of authentication systems, as summarized in Table 2. These managers typically generate random, complex, and strong passwords, bolstering resilience against guessing attacks [41, 96] and preventing password reuse across multiple services. Their auto-fill feature streamlines the authentication process by eliminating manual credential entry. However, it is imperative to implement robust security policies for password managers to safeguard stored passwords against theft [42, 61].

Despite the auto-fill feature’s convenience, studies have shown its susceptibility to network injection attacks and password vault exfiltration [77, 88, 89]. For a more detailed understanding, Figure 2 provides an illustrative depiction of how auto-fill mechanisms can be exploited, emphasizing these security concerns. To counteract these vulnerabilities, one effective strategy is to incorporate additional authentication methods, such as PINs or biometric verification, within the password manager. This approach not only maintains usability but also significantly enhances the overall security of the system. Comparatively, deployed password managers, commonly used

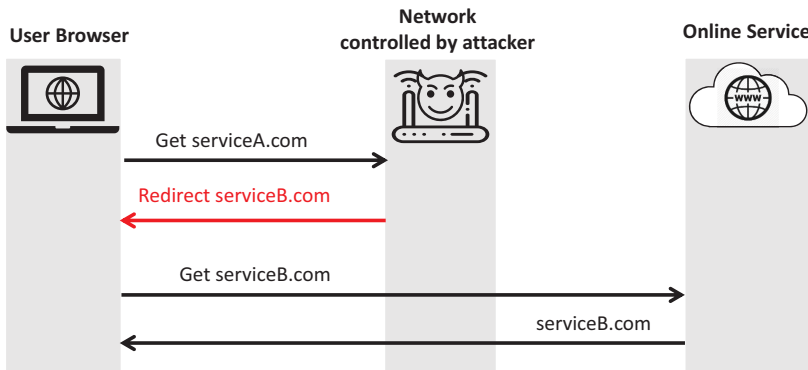


Fig. 2. Auto-fill attack flow.

in practice, and academic password managers, proposed in research, exhibit key differences. Deployed managers often prioritize user experience and widespread applicability, sometimes at the cost of advanced security features found in academic proposals. Conversely, academic password managers may introduce innovative security mechanisms that are not yet common in commercial products.

Interestingly, user preferences often tilt toward usability over security. This trend is particularly evident in the adoption of 2FA schemes. While no 2FA method provides all benefits (detailed in Section 5 and summarized in Table 3), 2FA remains more secure than password-only authentication. This increased security stems from the requirement for an attacker to compromise both the user’s credentials and the second-factor device, such as a hardware token or phone. Typically, 2FA schemes with higher usability are favored, even if they offer lower security. However, we advocate for improving the security of such schemes without compromising usability. For instance, the Sound-Proof system [62], known for its high usability, is susceptible to audio manipulation attacks [86]. This vulnerability could be addressed by restricting the audio resource usage (e.g., microphone, speaker) of the second-factor device exclusively to authentication processes. Additionally, raising user awareness regarding potential security threats and appropriate responses is crucial. Most security issues in 2FA schemes arise from user failure during the authentication process. Educating users on recognizing and responding to potential security threats is key to enhancing the overall security posture of these systems.

6.2 Remember Me

The “Remember Me” feature, often found on websites, enhances user convenience by reducing the need for repeated authentication on the same device/browser [16, 22]. It typically uses cookies to remember the user’s browser, allowing login to a 2FA-enabled service with just user credentials, bypassing the second-authentication factor. These cookies usually have a validity period, often up to 30 days. A variant of this feature, known as “keep me signed in” [25, 34], offers even greater ease by requiring only the username for login, without the need for any authentication factors. While using the “Remember Me” feature, users must consent to the service accessing and storing their browser session cookies. This practice, however, raises privacy concerns because it enables service providers to track users’ online activities across their websites [17]. Additionally, these cookies are vulnerable to theft through cross-site scripting (XSS) attacks [13] or session hijacking, compromising user session security. Another critical risk involves the physical theft of the device itself. If an unauthorized individual gains access to a device where the “Remember Me” feature is active, they can easily breach the user’s accounts. This risk is particularly pronounced in environments

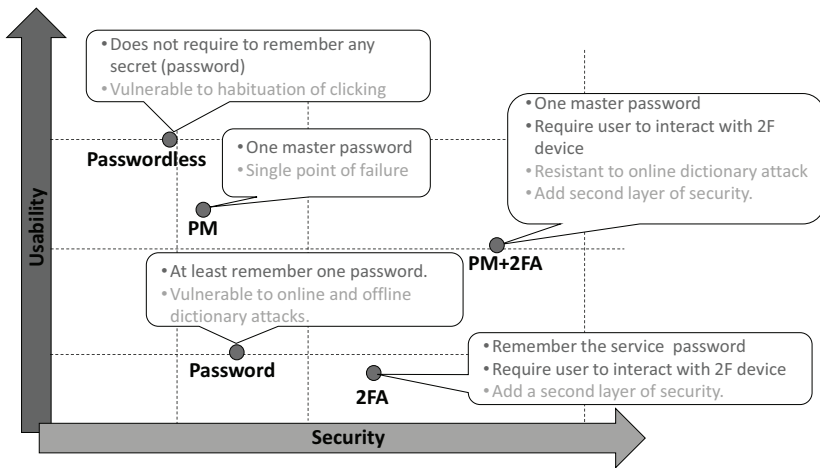


Fig. 3. Security and usability features of password, PM (password manager), passwordless, 2FA, and the combination of password managers with 2FA (PM+2FA).

where close attackers might access the device, such as in shared or public spaces. Additionally, malware installed on the user's device can exploit this feature, enabling unauthorized access to online services without requiring the user's credentials. While the "Remember Me" feature significantly improves system usability by reducing authentication efforts, it introduces notable security and privacy challenges that need careful consideration and mitigation.

6.3 Password Manager and 2FA as a Single System

As noted earlier, using password managers enhances both security and usability, whereas 2FA adds an additional security layer. Implementing a password manager alongside a 2FA scheme, referred to as *PM+2FA*, significantly enhances the overall system's security and usability. Figure 3 contrasts this combined system with traditional password-only authentication, stand-alone PM, the traditional OTP-2FA scheme, passwordless Push-2FA, and a security key.

Security. *PM+2FA* offers superior security compared to individual schemes. While passwordless Push-2FA is prone to user negligence and click-through habituation [53, 90], and the security key is vulnerable to compromised terminals [44], *PM+2FA* provides robust defense against these vulnerabilities, including common threats like phishing and brute-force attacks.

Usability. In terms of usability, most schemes, except passwordless Push-2FA, require at least one password/PIN, making them *Quasi-Memorywise-Effortless*. Most schemes are *Quasi-Nothing-to-Carry*, as they utilize a general-purpose phone, except for the security key, which requires carrying a specialized device. Passwordless Push-2FA excels in being *Efficient-to-Use* and *Easy-to-Learn*. The security key, although initially challenging to set up [81], becomes *Easy-to-Learn* post setup. OTP-2FA and *PM+2FA* are less *Efficient-to-Use* due to required interactions with a second-factor device. This efficiency can be enhanced in *PM+2FA* by incorporating low-effort 2FA schemes like Sound-Proof [63].

Adoption and Privacy Considerations. The adoption of *PM+2FA* systems is likely to be smoother for users, as many are already accustomed to using password managers and 2FA as separate entities. Integrating these systems into a single cohesive unit should, therefore, not pose significant adoption challenges. However, it is important to maintain a focus on user training and awareness to ensure that users can effectively utilize the combined system. Additionally, while privacy concerns are always paramount in the design of such systems, the integration of password

Table 4. Analytical Comparison of PM+2FA with Password-Only Authentication, PM (Password Manager), OTP-2FA, Passwordless Push-2FA, and the Security Key

Scheme	Usability								Deployability					Security							Privacy						
	Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Easy-Recovery-from-Loss	Infrequent-Errors	Auto-Generation*	Connection-less*	Accessible	Negligible-Cost-per-User	Client-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Guessing*	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-Trusted-Third-Party*	Requiring-Explicit-Consent*	Separation-between-Terminal-and-Device*	Unlinkable	Resilient-to-Leaks-to-the-Actual-Service*
Password	○	●	●	●	○	●	●	○	●	●	●	●	●	●	○	●	○	○	○	○	○	○	○	○	○	○	○
PM	●	●	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○
OTP-2FA	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Passwordless Push-2FA	●	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Passwordless security key	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
PM+2FA	●	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

● Offers the benefit; ◐ Almost offers the benefit; ○ Does not offer the benefit.

managers and 2FA—both familiar to users—should not introduce new privacy issues, provided that the implementation is carried out with due diligence to data protection.

Summary. In comparison to using only password managers or 2FA, the combined use of password managers and 2FA (PM+2FA) provides a more comprehensive security solution. A password manager alone improves password security and usability by generating and storing complex passwords, but it does not protect against phishing attacks or account compromise if the master password is stolen. However, 2FA adds a significant layer of security by requiring a second factor for authentication, making unauthorized access much more difficult. However, traditional 2FA methods can be cumbersome for users.

PM+2FA integrates the strengths of both approaches, offering robust security against both password theft and unauthorized access through a second factor while maintaining high usability with features like auto-fill and automated password management. This combination achieves the highest security level among the schemes compared, and its usability is particularly enhanced when integrating low-effort 2FA schemes, making it closer to password managers in terms of user experience. Therefore, PM+2FA is highly recommended for users who prioritize both security and usability.

By adopting PM+2FA systems that address known vulnerabilities in stand-alone systems, users can enjoy an optimal balance of robust security and convenient usability. Furthermore, users who are already familiar with these systems are likely to embrace the integration of PM and 2FA, facilitating smoother adoption. Continuous focus on user training, coupled with careful attention to privacy concerns, is essential to maximize the benefits of this integrated approach. A detailed comparison of these schemes is summarized in Table 4.

7 Future Scope

Our study highlights key areas for enhancing the effectiveness of password managers and 2FA schemes. Addressing the security and privacy concerns of password managers and improving the usability of 2FA is critical for advancing these tools. Future research should focus on integrating advanced encryption techniques and privacy-preserving technologies, such as homomorphic encryption [98] and differential privacy [50], to protect user data in password managers.

Additionally, exploring passwordless authentication methods like push notifications, security keys, and biometric systems (e.g., Apple Face ID) [70] can provide more user-friendly alternatives, reducing the cognitive and physical load on users. **Single sign-on (SSO)** [93] also offers a simplified authentication process, although ensuring the security of SSO providers remains essential. The incorporation of **artificial intelligence (AI)** in authentication systems is a promising trend, as AI can enhance security through behavioral biometrics [78] and real-time threat detection, learning and adapting to users' behavior patterns to provide an extra layer of protection. Furthermore, implementing risk-based authentication [95] can significantly enhance security by dynamically adjusting authentication requirements based on the assessed risk level of each login attempt, thereby balancing security with user convenience. By focusing on these areas, improving security and privacy in password managers, enhancing the usability of 2FA with passwordless methods and SSO, leveraging AI, and incorporating risk-based authentication, future research can lead to the development of more secure, user-friendly, and privacy-preserving authentication systems, contributing to a safer digital environment.

8 Conclusion

In this research, we conducted an analytical evaluation of various password managers and 2FA schemes, as currently deployed and proposed in academic literature. Utilizing a modified version of the framework of Bonneau et al. [44], tailored for the specific challenges of password managers and 2FA, we assessed these systems across multiple dimensions, including usability, deployability, security, and privacy. Our study reveals that no single scheme achieves perfection across all properties. We observed a tradeoff between usability and security: schemes scoring high in usability often compromise on security, and vice versa. For example, hardware token based 2FA schemes offer solid security but may fall short in usability, whereas online password managers enhance convenience but may pose security risks due to centralized password storage. However, our analysis indicates that a combined approach of password managers with 2FA (PM+2FA) might offer a more balanced score between security and usability. This integrated system could capitalize on the strengths and counterbalance the weaknesses of each individual method. The field of digital authentication is in a state of constant evolution, shaped by technological advancements and changing user requirements. Future research should explore creative combinations like PM+2FA and develop new solutions that refine the balance between usability and security. The ultimate objective is to design systems that are not only secure but also practical and accessible for users.

References

- [1] Infobip. 2020. What is Mobile Number Portability (MNP)? Retrieved from <https://www.infobip.com/glossary/mobile-number-portability>
- [2] Craig Timberg. 2013. NSA slide shows surveillance of undersea cables. *Washington Post*. Retrieved January 6, 2025 from <https://wapo.st/2HFZBPQ>
- [3] TechCrunch. 2013. SlickLogin Aims to Kill the Password by Singing a Silent Song to Your Smartphone. Retrieved January 6, 2025 from <https://tcrn.ch/3fC4dos>
- [4] Barton Gellman and Laura Poitras. 2013. U.S., British intelligence mining data from nine U.S. Internet companies in broad secret program. *Washington Post*. Retrieved January 6, 2025 from <https://wapo.st/2KSDaqD>
- [5] Samuel Gibbs. 2016. SS7 hack explained: what can you do about it? Retrieved from <https://www.theguardian.com/technology/2016/apr/19/ss7-hack-explained-mobile-phone-vulnerability-snooping-texts-calls>
- [6] 1Password. 2017. Information for Law Enforcement. Retrieved January 6, 2025 from <https://bit.ly/2Fw2JNI>
- [7] Cisco. 2017. Duo Security Two-Factor Authentication. Retrieved January 6, 2025 from <https://goo.gl/e38UnB>
- [8] Apple. 2017. Manage Passwords Using Keychains on Mac. Retrieved January 6, 2025 from <https://support.apple.com/en-us/guide/mac-help/mchl375f392/mac>
- [9] Dashlane. 2017. The App That Makes the Internet Easier. Retrieved January 6, 2025 from <https://www.dashlane.com/features>

- [10] Broadnet. 2020. What is OTP SMS and How It Works. Retrieved January 10, 2025 from <https://www.broadnet.me/what-is-otp-sms-and-how-it-works/>
- [11] Twilio. 2018. Authy | Two-Factor Authentication (2FA) App and Guides. Retrieved March 30, 2018 from <https://www.authy.com/>
- [12] Chris Hoffman. 2018. Criminals Can Steal Your Phone Number. Here's How to Stop Them. Retrieved January 10, 2025 from <https://www.howtogeek.com/358352/criminals-can-steal-your-phone-number-heres-how-to-stop-them/>
- [13] KirstenS. 2018. Cross-Site Scripting (XSS). Retrieved January 6, 2025 from <https://bit.ly/1Claka9>
- [14] Kim Zetter. 2010. Hacker Spoofs Cell Phone Tower to Intercept Calls. Retrieved January 10, 2025 from <https://www.wired.com/2010/07/intercepting-cell-phone-calls/>
- [15] RSA. 2018. RSA SecurID Hardware Tokens | Two Factor Authentication. Retrieved March 30, 2018 from <https://google.com/rcuQZK>
- [16] University of Washington IT Connect. 2019. Use the "remember me" option. Retrieved January 10, 2025 from <https://itconnect.uw.edu/tools-services-support/access-authentication/2fa/remember-me/>
- [17] E. V. Abhilash. 2018. Cookie Profiling. Retrieved January 10, 2025 from <https://www.linkedin.com/pulse/cookie-profiling-e-v-abhilash>
- [18] Yubico. 2018. Yubico | Trust the Net with YubiKey Strong Two-Factor Authentication. Retrieved March 30, 2018 from <https://www.yubico.com/>
- [19] Google. 2019. 2-Step Verification Phone Prompts. Retrieved January 6, 2025 from <https://bit.ly/2W309oo>
- [20] Cisco. 2019. Duo Push Notification. Retrieved January 6, 2025 from <https://duo.com/resources/videos/duo-push-demonstration>
- [21] Google. 2019. Google 2-Step Verification. Retrieved January 6, 2025 from <https://bit.ly/1AyTGig>
- [22] Matt Martin. 2019. Reduce your Duo logins with "Remember me." Retrieved January 10, 2025 from <https://michigan.umich.edu/news/2019/09/17/reduce-your-duo-logins-with-remember-me/>
- [23] Stefan Etienne and Barbara Krasnoff. 2019. How to use a two-factor security key. Retrieved January 10, 2025 from <https://www.theverge.com/2019/1/31/18203905/two-factor-authentication-security-key-how-to-yubico>
- [24] Microsoft. 2019. How to Use the Microsoft Authenticator App. Retrieved January 6, 2025 from <https://support.microsoft.com/en-us/help/4026727>
- [25] Microsoft Learn. 2024. Manage the "Stay signed in" prompt in Microsoft Entra ID. Retrieved January 10, 2025 from <https://learn.microsoft.com/en-us/entra/fundamentals/how-to-manage-stay-signed-in-prompt>
- [26] LastPass. 2019. How LastPass Works. Retrieved January 6, 2025 from <https://bit.ly/2OtJN3Y>
- [27] LastPass. 2019. LastPass Authenticator. Retrieved January 6, 2025 from <https://bit.ly/3nOeulV>
- [28] Google. 2019. Manage Saved Passwords. Retrieved January 6, 2025 from <https://bit.ly/38N0XUw>
- [29] Mozilla Support. 2019. Password Manager - Remember, delete and edit logins and passwords in Firefox. Retrieved November 25, 2019 from <https://support.mozilla.org/en-US/kb/password-manager-remember-delete-edit-logins>
- [30] Microsoft Support. 2019. Remember passwords and fill out web forms for Internet Explorer 11. Retrieved November 25, 2019 from <https://support.microsoft.com/en-us/windows/remember-passwords-and-fill-out-web-forms-for-internet-explorer-11-6883f6ce-0d1c-c2b9-e21e-705976d1c886>
- [31] RSA Community. 2019. SecurID Hardware Tokens. Retrieved August 29, 2019 from <https://community.rsa.com/s/article/SecurID-Tokens-e9b663a7>
- [32] RSA. 2019. RSA Security. Retrieved January 6, 2025 from <https://www.rsa.com/en-us/index>
- [33] Sound Login. 2019. Sound Login. Retrieved January 6, 2025 from <https://www.soundlogin.com/>
- [34] Google Support. 2019. Stay signed in or out of your Google Account. Retrieved January 10, 2025 from <https://support.google.com/accounts/answer/54490?hl=en>
- [35] Google Cloud. 2019. Titan Security Key. Retrieved January 6, 2025 from <https://cloud.google.com/titan-security-key>
- [36] Yubico. 2019. U2F Technical Overview. Retrieved January 6, 2025 from <https://bit.ly/2LC5Aqv>
- [37] Google. 2019. Use Your Phone's Built-In Security Key. Retrieved January 6, 2025 from <https://bit.ly/2JVKnri>
- [38] RoboForm. 2020. RoboForm Home Page. Retrieved January 6, 2025 from <https://www.roboform.com/>
- [39] Fadi Aloul, Syed Zahidi, and Wassim El-Hajj. 2009. Two factor authentication using mobile phones. In *Proceedings of the 2009 IEEE/ACS International Conference on Computer Systems and Applications*. IEEE, 641–644.
- [40] Apple. 2018. Local and Remote Notification Programming Guide. Retrieved July 1, 2020 from <https://apple.co/2CY1gRO>
- [41] Steven M. Bellovin and Michael Merritt. 1992. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy*. IEEE, 72–84.
- [42] Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh. 2010. Kamouflage: Loss-resistant password management. In *Proceedings of the European Symposium on Research in Computer Security*. 286–302.

- [43] 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 *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 553–567.
- [44] Thanh Bui, Siddharth Prakash Rao, Markku Antikainen, Viswanathan Manihatty Bojan, and Tuomas Aura. 2018. Man-in-the-machine: Exploiting ill-secured communication inside the computer. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security'18)*. 1511–1525.
- [45] Oliver Burkeman. 2012. Online passwords: Keep it complicated. *The Guardian*. Retrieved April 5, 2019 from <https://bit.ly/2OQHARr>
- [46] Sonia Chiasson, Paul C. van Oorschot, and Robert Biddle. 2006. A usability study and critique of two password managers. In *Proceedings of the USENIX Security Symposium*. 1–16.
- [47] 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*. ACM, 456.
- [48] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The tangled web of password reuse. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'14)*. 23–26.
- [49] Alexandra Dmitrienko, Christopher Liebchen, Christian Rossow, and Ahmad-Reza Sadeghi. 2014. On the (in) security of mobile two-factor authentication. In *Proceedings of the International Conference on Financial Cryptography and Data Security*. 365–383.
- [50] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Proceedings of the International Conference on Theory and Applications of Models of Computation*. 1–19.
- [51] Peter Eckersley. 2010. How unique is your web browser? In *Proceedings of the International Symposium on Privacy Enhancing Technologies Symposium*. 1–18.
- [52] Facebook. 2019. Two-Factor Authentication for Facebook Now Easier to Set Up. Retrieved May 10, 2019 from <https://bit.ly/2MpF3vP>
- [53] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the 8th Symposium on Usable Privacy and Security*. ACM, 3.
- [54] 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*. ACM, 657–666.
- [55] Google. 2018. Firebase Cloud Messaging | Firebase. Retrieved February 1, 2018 from <https://firebase.google.com/docs/cloud-messaging/>
- [56] Hackaday. 2015. Bypassing the Windows Lock Screen. Retrieved December 10, 2023 from <https://hackaday.com/blog/?s=bypassing+the+Windows+lock+screen>
- [57] J. Alex Halderman, Brent Waters, and Edward W. Felten. 2005. A convenient method for securely managing passwords. In *Proceedings of the 14th International Conference on World Wide Web*. ACM, 471–479.
- [58] Dianqi Han, Yimin Chen, Tao Li, Rui Zhang, Yaochao Zhang, and Terri Hedgpeth. 2018. Proximity-proof: Secure and usable mobile two-factor authentication. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 401–415.
- [59] Threat Intelligence. 2022. Malware Attacks—How They Work, Attack Vectors, and Prevention. Retrieved December 10, 2023 from <https://www.threatintelligence.com/blog/malware-attacks-how-they-work-attack-vectors-and-prevention>
- [60] Mohammed Jubur, Prakash Shrestha, Nitesh Saxena, and Jay Prakash. 2021. Bypassing push-based second factor and passwordless authentication with human-indistinguishable notifications. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. 447–461.
- [61] Ari Juels and Ronald L. Rivest. 2013. Honeywords: Making password-cracking detectable. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 145–160.
- [62] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-Proof: Usable two-factor authentication based on ambient sound. In *Proceedings of the 24th USENIX Security Symposium*. 483–498.
- [63] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. Sound-Proof: Usable two-factor authentication based on ambient sound.. In *Proceedings of the 24th USENIX Security Symposium*. 483–498.
- [64] Ambarish Karole, Nitesh Saxena, and Nicolas Christin. 2010. A comparative usability evaluation of traditional password managers. In *Proceedings of the International Conference on Information Security and Cryptology*. 233–251.
- [65] 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 *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 523–537.

- [66] Radhesh Krishnan Konoth, Victor van der Veen, and Herbert Bos. 2016. How anywhere computing just killed your phone-based two-factor authentication. In *Proceedings of the International Conference on Financial Cryptography and Data Security*. 405–421.
- [67] Juan Lang, Alexei Czeskis, Dirk Balfanz, Marius Schilder, and Sampath Srinivas. 2016. Security keys: Practical cryptographic second factors for the modern web. In *Proceedings of the International Conference on Financial Cryptography and Data Security*. 422–440.
- [68] Zhiwei Li, Warren He, Devdatta Akhawe, and Dawn Song. 2014. The emperor's new password manager: Security analysis of web-based password managers. In *Proceedings of the 23rd USENIX Security Symposium*. 465–479.
- [69] Ximing Liu, Yingjiu Li, and Robert H. Deng. 2018. Typing-Proof: Usable, secure and low-cost two-factor authentication based on keystroke timings. In *Proceedings of the 34th Annual Computer Security Applications Conference*. ACM, 53–65.
- [70] 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 *Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP'20)*. IEEE, 268–285.
- [71] Joshua Meier, Jesse Zhang, Richard Zou, and James Mickens. 2017. Zero-effort two-factor authentication using audio signals. In *Proceedings of the 2017 International Symposium on Cyber Security Cryptography and Machine Learning (CSCML'17)*. 10.
- [72] Microsoft. 2020. Windows Push Notification Services (WNS). Retrieved July 1, 2020 from <https://bit.ly/3jSSEML>. (2020)
- [73] David M'Raihi, Mihir Bellare, Frank Hoornaert, David Naccache, and Ohad Ranen. 2005. *HOTP: An HMAC-Based One-Time Password Algorithm*. RFC 4226. Internet Engineering Task Force.
- [74] David M'Raihi, Salah Machani, Mingliang Pei, and Johan Rydell. 2011. *TOTP: Time-Based One-Time Password Algorithm*. RFC 6238. Internet Engineering Task Force.
- [75] Sophos News. 2018. Lock Screen Bypass Already Discovered for Apple's iOS 12. Retrieved December 10, 2023 from <https://news.sophos.com/en-us/2018/10/02/lock-screen-bypass-already-discovered-for-apples-ios-12/>
- [76] Trend Micro News. 2022. Android Phones Vulnerable to Lock Screen Bypass Exploit. *Trend Micro News* Retrieved December 10, 2023 from <https://news.trendmicro.com/2022/11/14/android-phones-lock-screen-vulnerability-bypass-exploit-google-pixel/>
- [77] Sean Oesch and Scott Ruoti. 2019. That was then, this is now: A security evaluation of password generation, storage, and autofill in thirteen password managers. *arXiv preprint arXiv:1908.03296* (2019).
- [78] Yris Brice Wandji Piugie, Joël Di Manno, Christophe Rosenberger, and Christophe Charrier. 2021. How artificial intelligence can be used for behavioral identification? In *Proceedings of the 2021 International Conference on Cyberworlds (CW'21)*. IEEE, 246–253.
- [79] Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin R. B. Butler. 2016. Sending out an SMS: Characterizing the security of the SMS ecosystem with public gateways. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP'16)*. IEEE, 339–356.
- [80] Jonathan Reed. 2023. LastPass Breaches Cast Doubt on Password Manager Safety. Retrieved January 6, 2025 from <https://securityintelligence.com/news/lastpass-breaches-cast-doubt-on-password-manager-safety/>
- [81] 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 *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP'18)*. IEEE, 872–888.
- [82] Blake Ross, Collin Jackson, Nick Miyake, Dan Boneh, and John C. Mitchell. 2005. Stronger password authentication using browser extensions. In *Proceedings of the USENIX Security Symposium*. 17–32.
- [83] Margaret Rouse. 2017. Keylogger (keystroke logger or system monitor). *TechTarget*. Retrieved January 6, 2025 from <https://searchsecurity.techtarget.com/definition/keylogger>
- [84] Bruce Schneier. 2005. Two-factor authentication: Too little, too late. *Communications of the ACM* 48, 4 (2005), 136.
- [85] Maliheh Shirvanian, Stanislaw Jarecki, Hugo Krawczyk, and Nitesh Saxena. 2017. SPHINX: A password store that perfectly hides passwords from itself. In *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS'17)*. IEEE, 1094–1104.
- [86] Babins Shrestha, Maliheh Shirvanian, Prakash Shrestha, and Nitesh Saxena. 2016. The sounds of the phones: Dangers of zero-effort second factor login based on ambient audio. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 908–919.
- [87] Prakash Shrestha and Nitesh Saxena. 2018. Listening Watch: Wearable two-factor authentication using speech signals resilient to near-far attacks. In *Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 99–110.
- [88] David Silver, Suman Jana, Dan Boneh, Eric Chen, and Collin Jackson. 2014. Password managers: Attacks and defenses. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security'14)*. 449–464.

- [89] Ben Stock and Martin Johns. 2014. Protecting users against XSS-based password manager abuse. In *Proceedings of the 9th ACM Symposium on Information, Computer, and Communications Security*. ACM, 183–194.
- [90] Joshua Sunshine, Serge Egelman, Hazim Almuhammedi, Neha Atri, and Lorrie Faith Cranor. 2009. Crying wolf: An empirical study of SSL warning effectiveness. In *Proceedings of the USENIX Security Symposium*. 399–416.
- [91] Luren Wang, Yue Li, and Kun Sun. 2016. Amnesia: A bilateral generative password manager. In *Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS'16)*. IEEE, 313–322.
- [92] Mingyue Wang, Wen-Tao Zhu, Shen Yan, and Qiong Xiao Wang. 2018. SoundAuth: Secure zero-effort two-factor authentication based on audio signals. In *Proceedings of the 2018 IEEE Conference on Communications and Network Security (CNS'18)*. IEEE, 1–9.
- [93] 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 *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 365–379.
- [94] Rob Waugh. 2012. No wonder hackers have it easy: Most of us now have 26 different online accounts—but only five passwords. *Daily Mail*. Retrieved April 5, 2019 from <https://dailym.ai/2uNS2Qk>
- [95] Stephan Wiefeling, Luigi Lo Iacono, and Markus Dürmuth. 2019. Is this really you? An empirical study on risk-based authentication applied in the wild. In *Proceedings of the 34th IFIP 11 International Conference on ICT Systems Security and Protection (SEC'19)*. 134–148.
- [96] Thomas D. Wu. 1998. The secure remote password protocol. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'98)*. 97–111.
- [97] Ka-Ping Yee and Kragen Sitaker. 2006. Passpet: Convenient password management and phishing protection. In *Proceedings of the 2nd Symposium on Usable Privacy and Security*. ACM, 32–43.
- [98] Xun Yi, Russell Paulet, Elisa Bertino, Xun Yi, Russell Paulet, and Elisa Bertino. 2014. *Homomorphic Encryption*. Springer.
- [99] Dimitra Zarafeta, Christina Katsini, George E. Raptis, and Nikolaos M. Avouris. 2019. UltraSonic watch: Seamless two-factor authentication through ultrasound. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

Appendix

A. Schemes Workflow

Figures 4, 5, and 6 show the workflow of some protocols we evaluate in this article.

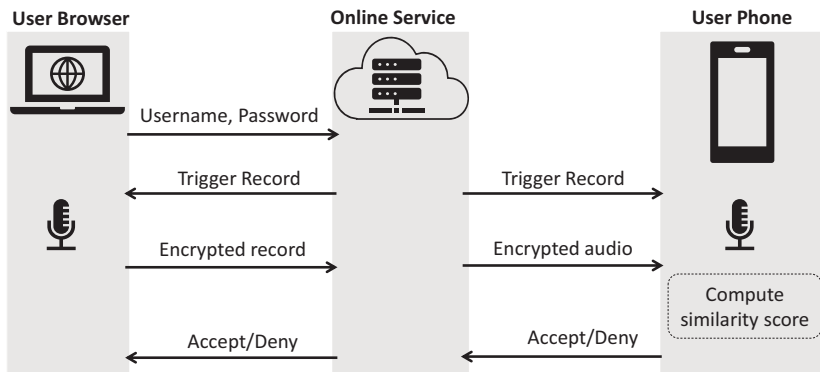


Fig. 4. Sound-Proof authentication workflow. During login, the online service activates microphones on both the user's browser device and phone. It then encrypts and sends the audio recording from the browser to the phone for comparison, determining similarity for authentication.

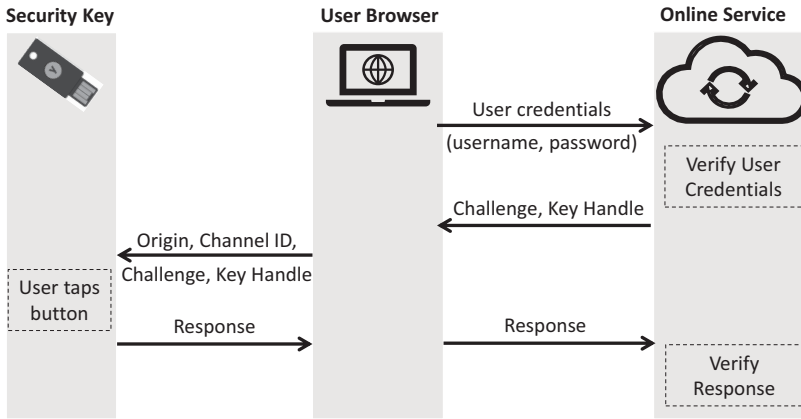


Fig. 5. Authentication flow of a security key. During login, the online service sends a challenge and a key handle to the security key through the authenticating terminal. The login succeeds when the user approves the authentication by pressing the security key [36].

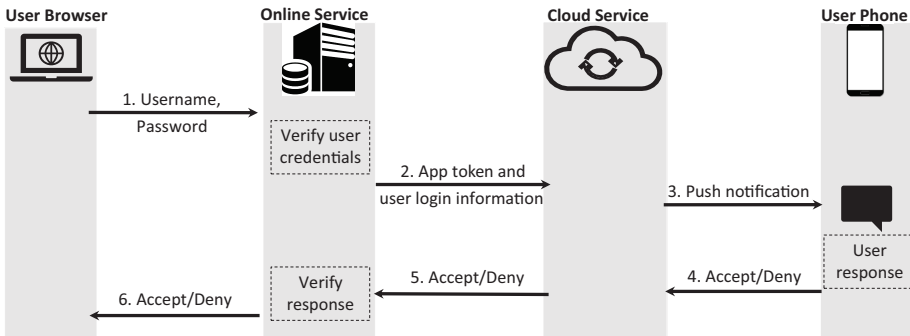


Fig. 6. A high-level overview of Push-2FA.

Received 1 January 2024; revised 16 July 2024; accepted 30 November 2024