University of Duisburg-Essen
Faculty of Business Administration and Economics

Beyond Passwords? About the current state of FIDO2 authentication

Nils Höll

3087271
nils.siegle@stud.uni-due.de

Supervisor: Peter M. Schuler

May 2020

Abstract

Passwords are insecure and annoying to use, especially if one tries to use them in a secure way. We know the problems regarding this form of authentication, there are dozens of studies and articles about why passwords should not be used anymore, how they can be made more secure, and why people still reuse already weak credentials despite better knowledge [1, 9, 17, 38, 27].
Those findings and their conclusions - that we need better forms of authentication for web applications - are supported by the almost regular credential leaks from companies in all branches, collected by sites like Have I Been Pwned or the Hasso-Plattner-Institut Identity Leak Checker.
One of the many proposals on how to tackle this problem comes from the Fast IDentity Online (FIDO) Alliance. Their new authentication framework, Fast IDentity Online 2 (FIDO2), promises an open standard for secure and easy to use web authentication.

This paper analyzes the current body of knowledge regarding the security and usability of FIDO2 and tries to draw a conclusion whether or not it could replace legacy passwords in the future.

Contents

Abbreviations

1FA	Single-Factor Authentication
2FA	Two-Factor Authentication
authn	Authentication
authz	Authorization
API	Application Programming Interface
CTAP	Client to Authenticator Protocol
CTAP2	Client to Authenticator Protocol 2
FIDO	Fast IDentity Online
FIDO2	Fast IDentity Online 2
HOTP	HMAC-based One-time Password
IT	Information Technology
MFA	Multi-Factor Authentication
NIST	National Institute of Standards and Technology
NFC	Near-Field Communication
OS	Operating System
OTP	One-time Password
OWASP	Open Web Application Security Project
PIN	Personal Identification Number
RP	Relying Party
TPM	Trusted Platform Module
TOTP	Time-based One-time Password
U2F	Universal Second Factor
UAF	Universal Authentication Framework
W3C	World Wide Web Consortium

I. Introduction

Passwords are everywhere in the daily online life of almost every person. No matter if it is a new social media platform, an e-commerce website or online banking - dozens of services require internet users to authenticate using a preferably strong and unique password [15].

As many of these accounts contain very sensitive data, users are afraid of their credentials becoming exposed during one of the many data breaches of the recent years [4]. Monitoring services like Have I Been Pwned[1] or the Hasso-Plattner-Institut Identity Leak Checker[2] try to gather information about those breaches, and to inform affected internet users. The former service currently reports almost ten billion breached accounts from 440 different websites, the latter over ten billion from over 1,000 leaks [16, 22].

A known problem of those breaches is that many people reuse their mostly weak passwords [1] - around 70% of the credentials in a new breach published in [16] are already in the system. Some websites are trying to get around this problem by enforcing arbitrary password requirements like length, use of special characters or banning known weak passwords. Oftentimes those measures do not increase password security, but do confuse the users trying to generate valid credentials for a new website [20]; therefore, the American National Institute of Standards and Technology (NIST) does no longer recommend using such rules [15].

Although password managers provide an effective way of generating and storing long, random and therefore unique passwords, they introduce an additional step in the authentication flow [25]. But even those do not protect users against phishing or misconfigured services that store the password in plain text instead of hashed and salted, or accidentally log it like Facebook did until 2019 [13]. After all, most security researchers as well as empirical evidence agree that passwords are a less than optimal form of authentication, especially because their security relies on the choices users make - and studies show that those are more often than not weak ones [17, 38].

Is a secure and easy-to-use authentication possible without passwords or even storing any credentials?

The FIDO Alliance, a project supported by hundreds of tech giants like Amazon, Google, Facebook, Apple, American Express, Paypal and many others[3] claims to have a solution: The FIDO2 authentication framework. Logging in would no longer require a password[4], but a hardware token (*authenticator*) like a USB stick. Because the technology utilizes public-key cryptography, the actual secret would never leave the authenticator, drastically improving security.

This paper aims to summarize the state of the art of research about FIDO2 and to answer the following research question:

> **R:** *Can the FIDO2 framework provide secure and easy-to-use web authentication?*

Considering the potential impact of weak, reused and leaked passwords, and the recent support of FIDO2 by many companies with the goal to get rid of passwords [30, 29, 28, 12], a closer look at the current research regarding this topic and possible open questions is relevant for both academics and practitioners.

Previous literature has already shown evaluations of many different authentication methods, including FIDO2's predecessors Universal Second Factor (U2F) and Universal Authentication Framework (UAF), coming to rather depressing conclusions [2, 19, 23, 6] - although passwords are known to be insecure and prone to a variety of attacks, most other options are considered too complex to either implement or use.

[1] https://haveibeenpwned.com

[2] https://sec.hpi.de/ilc/

[3] https://fidoalliance.org/members/

[4] FIDO2 can also be used as an additional factor, see chapter III.

II. Methods

To get detailed information about the current body of knowledge regarding FIDO2 and to answer the research question, a three-phased literature review is conducted.
The first phase is an unstructured internet search using common search engines like Google[5] or DuckDuckGo[6]. The aim of these searches is to get a basic understanding of what FIDO2 is, how it works, and how it is perceived in different media outlets.
These findings enable the second phase, a structured search for scientific literature regarding different subtopics or aspects of the FIDO-Framework, password-based authentication and other foundations. It makes use of more research-oriented search engines like Google Scholar[7], ScienceDirect[8], IEEE Xplore[9], AISeL[10] and the official FIDO Alliance whitepaper page[11].
In the last phase a search for secondary literature is conducted in the results of phase two, aiming for related scientific research on the topic.

To answer the research question the framework proposed by Bonneau et al. (2012) is used as a base, as the different benefits used for evaluating authentication methods are often referred in literature. However, the dimension *Deployability* is dropped, leaving a focus on *Usability* and *Security* as proposed in the research question.
Those two parts are again supported by using other evaluation frameworks. For usability this will be the categorization proposed by Nielsen (1993) to make sure all aspects of acceptability and usability are covered. The security section is backed by a guide from the Open Web Application Security Project (OWASP), more specifically the second vulnerability in the 2017 edition of OWASPs Top 10, *Broken Authentication*.

III. Foundations

The following chapter explains some basic concepts of web authentication, digital identities and the basic functionality of the FIDO2 framework.

i. Authentication and Authorisation

Authentication (authn) and Authorization (authz) are concepts present in virtually every protected Information Technology (IT) ressource. The former describes the process of identifying a user by verifying their digital identity, hence checking their *authenticity*. Oftentimes only certain users are *authorized* to access a protected ressource (e.g. specific parts of a website, sensitive data etc.). Therefore, an authorization has to take place both to ensure that the correct information is distributed to each user, and that no one without permission is able to manipulate data.
The most common and widespread type of web authentication is the so-called password-based authentication. In this case a user has to provide a *username* and a *password* or *passphrase*, a secret string only known to the user. Only the correct combination of username and password grants access to the protected resource.

In the above case the password is also referred to as an *authentication factor*, of which there are three categories [34]:

Knowledge factors Something (only) the user knows like a password/-phrase or a Personal Identification Number (PIN)

Ownership factors Something the user owns like a smart card or a Time-based One-time Password (TOTP) token

Inherence factors Mostly biometric factors like fingerprints or facial recognition

[5] https://google.com
[6] https://duckduckgo.com/
[7] https://scholar.google.com/
[8] https://www.sciencedirect.com/
[9] https://ieeexplore.ieee.org/
[10] https://aisel.aisnet.org/
[11] https://fidoalliance.org/white-papers/

This classification allows for a basic evaluation of possible authn factors: While inherence factors are almost always available (the user carries them around), they often require special hardware like fingerprint sensors and are therefore not best suited for web authentication.
Ownership factors on the other hand are less resilient against (physical) theft - if the smartphone or hardware token is stolen, an attacker could successfully authenticate.
The security of passwords and other knowledge factors is dependent on the users and how they store and generate them. Reusing passwords or using obvious, common or dictionary-based strings is undeniably less secure than using long, randomly generated ones stored in a password manager [25, 21, 17]. Therefore, application developers have little influence on the security of possibly very important accounts, and if they try to enforce strong passwords by using arbitrary requirements, this can easily backfire [20] and just decrease usability instead of increasing security. For that reason, the NIST has dropped all recommendations to use password complexity requirements and instead suggests to rely on length and black lists of known (and therefore insecure) passwords [15].

ii. Multi-Factor Authentication

To counter the security problems inherent to each single factor, Multi-Factor Authentication (MFA) can be used for further securing any authentication process by requiring more than one factor. Often this is realized as a Two-Factor Authentication (2FA) using a normal password (knowledge factor) as the first, and some kind of One-time Password (OTP) (for example TOTP, HMAC-based One-time Password (HOTP) or an SMS/eMail-Code) as the second.
In this case knowing the password of an account is not sufficient for logging in; a possible attacker would also have to get access to the device generating or receiving the second factor, often the users phone (ownership factor) [8, 19].
It is important for MFA in general that at least two of these factors are in different categories (e.g. a knowledge and an ownership factor) and at least one of them is not reusable (i.e. a OTP) [15, 34].

Using multiple factors can protect accounts against standard brute-force attacks like credential stuffing, where an attacker tries many combinations of known usernames and popular passwords.
However, MFA is not a reliable protection against phishing attacks [26], because a malicious site can easily fake an input field for OTPs.

iii. FIDO2 Overview

The FIDO Alliance was founded in 2012 by a consortium of software and hardware companies with the goal to create an open, universal authentication framework. In December of 2014 the specifications for the Universal Authentication Framework, a protocol for passwordless authentication, and Universal Second Factor, a protocol designed for hardware key based second factors, were published by the alliance. The experiences from those protocols were later condensed into the FIDO2 framework, launched only recently in April of 2018 [10].

FIDO2 aims to provide a strong authentication factor (either as single factor for passwordless login or in a MFA setting) using hardware keys, so called *authenticators*. Those can be on the device itself (for example using *Windows Hello*, Apples *TouchID* or a Trusted Platform Module (TPM) chip), and therefore *internal* authenticators. The alternative are external or *roaming* authenticators, often realised as USB keys or Near-Field Communication (NFC) cards.
The party providing the protected ressource (e.g. a website or other web application) is referred to as *relying party*.
The FIDO2 Framework consists mainly of two

components: The *WebAuthn* web Application Programming Interface (API), standardized by the World Wide Web Consortium (W3C), is the interface between the browser or application and the authentication servers of the relying party [36]. WebAuthn is currently supported by the Windows 10 and Android Operating Systems (OSs) as well as by the Google Chrome, Mozilla Firefox, Microsoft Edge and Apple Safari web browsers [11].
The second part is the Client to Authenticator Protocol 2 (CTAP2)[12] that is responsible for connecting roaming authenticators to the clients device (e.g. a smartphone, laptop or personal computer) [12, 3].

As seen in figure 1, the Relying Party (RP) will never know what kind of authenticator was used - as long as it can communicate over WebAuthn (or indirectly over CTAP2).
The WebAuthn API is used for two major operations: Either to register a new authenticator with an application, creating new public key credentials, or to login to one of these services. For logging in, the RP sends an *authentication assertion*. The authenticator has to verify the presence and consent of the user, usually using a button or fingerprint sensor on the security key, before the assertion is signed with the correct private key and send back to the RP.
The RP can now check the cryptographic signature on the assertion, verifying the authenticity of the user trying to log in [36, 37].

IV. Results

To answer the research question, the literature is analyzed for two main topics: usability and security.
While there is some amount of research regarding the usability of FIDO2, literature regarding the preceding protocols U2F and UAF is also taken into account due to the similarity of the involved hardware and authentication flows.
The security part of this chapter is mainly based on evaluations of the proposed concepts and mechanisms, as there is not yet any research regarding the security of actual implementations of either hardware or protocols.

i. Usability

To determine whether a new authentication concept is accepted by users, and could therefore replace passwords, the usability plays a significant role. Regular passwords have been the prevalent authentication mechanism since the 1960's [27], so people are very familiar with their usage. The concept of sending credentials, checking them, and granting access to the protected ressource is also very intuitive. Using a hardware token, or in case of internal authenticators only a system dialogue, is a radically different approach.
A good starting point to evaluate whether this hardware-dependent approach is suitable is the study conducted by Lang et al. (2017). The authors analyzed the roll-out of hardware tokens for 50,000 employees of Google as a second factor. In comparison to other 2FA methods like TOTP, they found using a security key to be significantly faster [23].
On the other hand, a more qualitative and in-depth study of Das et al. (2018) showed major usability problems for participants without technical backgrounds, especially during the initial setup. They report that users had trouble finding the correct instructions and where not able to intuitively use the device, as they had not yet developed a mental model regarding the functionality of the keys. The authors also explicitly state that one of the major problems was that users still had to enter their passwords, leading to additional work and complexity without any perceived benefit [6]. This factor becomes less of a problem when a FIDO2 token is used for passwordless authentication.

The most important scientific work for this section is a recent study by Lyastani et al., published in 2020, that focuses on the usability of FIDO2 as a means of passwordless,

[12] In this paper the acronyms CTAP and CTAP2 are used synonymously. This is due to the fact that CTAP was later re-branded as CTAP2, while U2F became CTAP1.

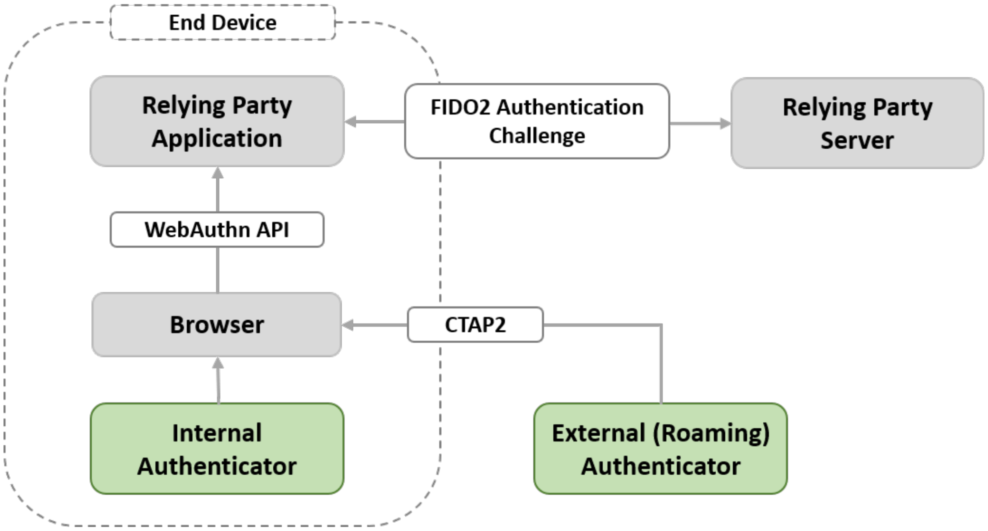

End Device
Relying Party Application
FIDO2 Authentication Challenge
Relying Party Server
WebAuthn API
Browser
CTAP2
Internal Authenticator
External (Roaming) Authenticator

Figure 1: *FIDO2 WebAuthn & CTAP2 Authentication Flow*

Single-Factor Authentication (1FA). They conclude that participants rated logging in via FIDO2 and an external authenticator as more usable than standard passwords, and were able to show a higher acceptance than the latter [26].
However, the authors also identified some challenges, of which the most important ones are listed below:

Complex Setup Especially the initial registration of the key was seen as a challenge by many participants, this corresponds with the findings of [6].

Requires Physical Device Users criticized that they had to carry an authenticator, and that the absence of this devices effectively logs them out of the application, substantially limiting spontaneous use.

Support of Other Devices Logging in on devices without any way to connect the authenticator (e.g. a tablet or public PC without USB-A ports or NFC) is impossible.

Account Recovery after Loss If the security key gets lost or stolen, account recovery is much more complex than with passwords. The current recommendation is to register a second backup key with each account [14].

No Account Delegation It is not possible to grant a trusted third party (e.g. a colleague) access to an account without giving away the hardware key.

The above findings are now applied to the framework proposed by Nielsen (1993), where acceptability is split into social and practical acceptability. Due to the lack of literature and relevance for the research question, only the practical usability of FIDO2 is considered in this paper.
Practical acceptability is now divided into *cost*, *compatibility*, *reliability* and *usefulness* [31, pp. 25 sq.]:

Cost In the study by [26], around a third of the participants stated that they would rather not use FIDO2 due to the cost of external authenticators. This is partly countered by the possible use of internal authenticators (such as a Windows 10 PC or an Android Smartphone), but if a roaming authenticator is required for mobility, prices range between 20€ and 45€ per key [39] - which means at least 40€ if a backup key is registered.

Compatibility As WebAuthn is a browser protocol that is currently implemented in almost every major browser, compatibility should not be an issue. Additionally, it is currently supported in Windows 10 and Android, with full support for Apples iOS on its way [35]. As Client to Authenticator Protocol (CTAP) and WebAuthn are open standards, compatibility should further increase in the near future.

Reliability The reliability of a FIDO2 authentication is hard to estimate, as many components are required to work. For WebAuthn, the standardisation by the W3C should be sufficient to conclude that it is robust. An unknown factor are hardware keys, as their reliability depends on the manufacturer.

The usability, according to [31, pp. 25 sq.] a subcomponent of *usefulness*[13], is again divided into five metrics, that are evaluated in the following paragraph:

Easy to learn The basic usage of the FIDO2 framework is easy to learn, although better and more user-friendly documentation is required as most people are only used to passwords [26, 18, 6].

Efficient to use Using a security key is very fast once the users have learned it [23].

Easy to remember As the process of logging in itself is quite simple and, contrary to passwords, identical for each application, it should be easy to reproduce. Nevertheless, there is currently no long-term research to further support this evaluation.

Few errors Once setup correctly, there is very little room for errors. However, this also depends on the authenticator used, as for example biometrics could confuse a user and therefore introduce errors [26].

Subjectively pleasing Participants in recent studies have found security keys enjoyable to use in 1FA scenarios [26], but were less convinced in 2FA settings [6].

There is one major challenge that does not really fit in one of the above categories, but was identified by all studies conducted on the use of hardware security keys, no matter if FIDO2 as 1FA or with U2F: If the key gets lost, stolen or destroyed, there is currently no easy way to recover access or revoke the key. If the user has no other way of authentication (like a second registered key as proposed by [14] or a backup password), access for all linked accounts has to be recovered individually - probably using a multitude of different approaches chosen by each provider. But even with a backup authentication method, the lost key has to be revoked and a new one registered for each service.

ii. Security

From a security point of view we can mainly analyze the concepts behind the protocols in the FIDO2 framework, as there is not yet any research regarding actual protocol implementations or hardware. Therefore, this section is mainly based on the official documentation and first evaluations of security researches. Additionally, we will look mostly at using FIDO2 in the passwordless, single factor mode.
The main difference to standard passwords is the use of public-key cryptography instead of storing a shared secret. For every new application the authenticator generates a separate key pair, of which only the public key is stored within the RPs servers. This has two implications: First, different credentials are used for every service [37, 36], which means accounts cannot be linked across services based on the key. As the user has no influence on the key generation, there is no way to derive user information from the public key. Second, as no secret is stored on the RPs side, a credential leak has no effect - knowing the

[13] The other sub-component, *utility*, is not further evaluated in this paper, as the only use of FIDO2 is authentication.

public key is of no use for taking over an account. In addition to that resilience against credential leaks, the public-key infrastructure also prevents against phishing. Even if, for example, a malicious website could forge a FIDO2 authentication challenge using stolen public keys, the challenge response can not be used to login to the real account [11, 12].
In general, public-key infrastructures are considered very robust and mature, although their security depends on the algorithm and key length used.

To further judge the security of the framework criteria are derived from a guide from the OWASP. The *OWASP Top 10*[14] are a set of ten common security vulnerabilities in web applications based on the number of reported vulnerabilities, user surveys and ranked industry surveys. Although this list is not directly based on scientific research, it is very well known and broadly used, also in other research [33].
For this paper the second vulnerability in the top ten is most relevant: *A2 - Broken Authentication*. The following criteria are based on OWASPs guide to detect such vulnerabilities, leaving out session-related issues, as this is out of scope for FIDO2 [32].

Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords. To perform such an attack, an attacker would need the private key of a user, which can never leave the authenticator. As the private key is never stored on any server, obtaining large amounts of keys is extremely difficult and of not much use, as different keys are used for each application.

Permits brute force or other automated attacks. Not in the protocol itself, but the use of asymmetric cryptography for authentication prevents off-line brute force attacks of leaked credentials.

Permits default, weak, or well-known passwords, such as “Password1” or “admin/admin“. Again, the users have no influence on the credentials created; the keys are entirely random.

Uses weak or ineffective credential recovery and forgot-password processes, such as “knowledge-based answers”, which cannot be made safe. This is mostly dependent on the application itself, implementing weak recovery is still possible with FIDO2.

Uses plain text, encrypted, or weakly hashed passwords. The actual credentials (private keys) are not stored on the server, and there is no meaningful way of deriving those from the stored public keys.

Has missing or ineffective multi-factor authentication. As said before, FIDO2 can be used either as a single or a (strong) second factor. In the end, this is down to the user.

While the protocols and general authentication flow can be seen as very secure and less prone to various attacks than passwords, another big factor of the general security are the authenticators themselves. In all cases, it should not be possible to extract the private keys from the authenticator. Additionally, the presence and consent of the user has to be determined for every login, meaning the person has to physically act by pressing a button or accepting a system dialog. This ensures that malicious services cannot trigger and sign assertions without the users knowledge. Many authenticators, like the popular YubiKey 5, also implement another factor like a PIN or a fingerprint sensor to further protect against theft [39, 7]. Further research is needed to make any sound statements on the safety of authenticators, either internal or roaming. There are currently no known side-channel attacks to extract private keys.

[14] https://owasp.org/www-project-top-ten/

iii. Comparison

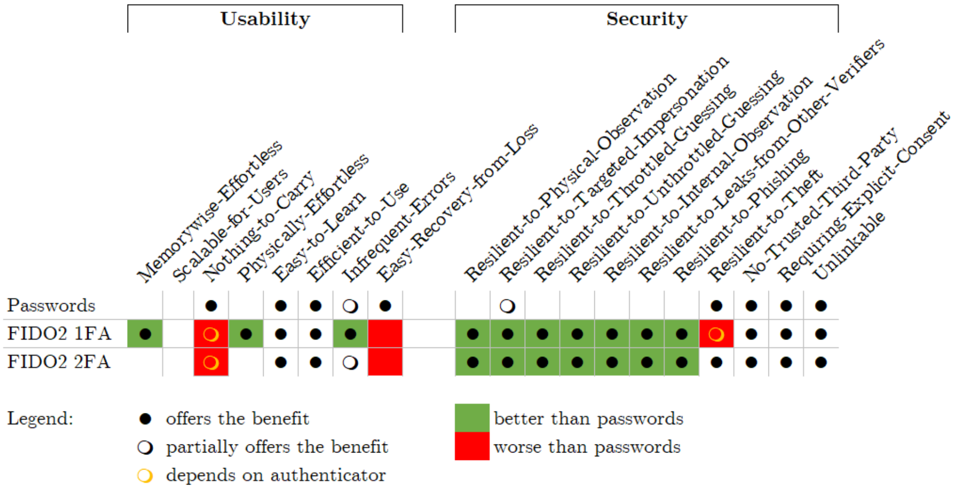
The below comparison of passwords, FIDO2 passwordless login and FIDO U2F is based on the authentication method evaluation framework developed by Bonneau et al. [2]. The evaluation of simple passwords, also proposed by the former authors and supported by others [26] is not changed.
To fill in the gaps for FIDO2 we can now make clear statements based on the results above. For the dimension of usability, some of the benefits (*Easy-to-Learn*, *Efficient-to-Use* and *Infrequent-Errors*) can be directly mapped from the categories developed by [31] - FIDO2 in 1FA offers all of these benefits, while 2FA has limited efficiency due to the additional need for passwords. If used for passwordless auth, FIDO2 is also memorywise and physically effortless. Again, if used as a second factor those benefits are countered by the additional passphrase. None of the options really scales for many accounts, although federated authentication could be possible. However, this is outside the scope of this paper. As mentioned above, an easy recovery from loss is not possible with FIDO2.
The security perspective is also pretty straightforward: As nothing is entered by the user (except maybe a PIN), FIDO2 is resilient to observation. The use of asymmetric cryptography for the assertions also protects sufficiently against all forms of guessing, internal observation (like keyloggers), credential leaks and phishing. Theft might be a problem depending on the authenticator used, especially if no second factor (PIN or fingerprint) is used on the security key itself.

As figure 2 shows, both FIDO2 modes score exceptionally well. In fact, as is also mentioned in [26], no other authentication method offers as many benefits as the former.

V. Discussion

As in the previous chapter, the evaluation of the results is also split into the two dimensions usability and security. Looking at the results, the research question probably has to be answered with *Yes, but. . .*
In general, FIDO2 can indeed provide secure and easy-to-use authentication. In fact, the comparison in figure 2 shows that it is significantly more secure than using passwords. From a usability point of view passwords seem to be better in two, but worse in three categories. Especially the memory effort is a key factor in why passwords tend to be insecure, because users can't memorize too many different ones [25, 9, 38].

The results also clearly show that the FIDO Alliance has to provide better answers to the problem of account recovery than to recommend using a second key for each site. Having a standardized process or even a way of revoking the lost key from all accounts at once would probably have a huge impact on usability and therefore acceptance.
Some other challenges rely heavily on the relying party and how they implement the application. As shown by [6], having clear and easy to understand instructions could drastically decrease the complexity of the setup. Account delegation might also be possible by using additional, password-like authentication tokens. Such tokens could be randomly generated by the main user inside the application, and then distributed like regular passwords. Although this would increase the attack surface and re-introduce security problems of passwords, the risk could be minimized by limiting the token to either a time frame (e.g. valid for 24h) or a number of logins allowed.
What remains is mainly the issue of having to carry around a physical device. With support of FIDO2 in Android and soon iOS, using a smartphone as an authenticator reduces that problem, as many people already carry their phone at all times. This, however, would decrease the compatibility, as the device on which the user wants to authenticate would require a connection using Bluetooth or NFC due to the lack of a standrad USB-A port.



	Usability								Security										
	Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Throttled-Guessing	Resilient-to-Unthrottled-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
Passwords			●		●	●	◐	●		◐						●	●	●	●
FIDO2 1FA	🟩 ●		🟥 ○	🟩 ●	●	●	🟩 ●	🟥	🟩 ●	🟩 ●	🟩 ●	🟩 ●	🟩 ●	🟩 ●	🟩 ●	🟥 ○	●	●	●
FIDO2 2FA			🟥 ○		●	●	◐	🟥	🟩 ●	🟩 ●	🟩 ●	🟩 ●	🟩 ●	🟩 ●	🟩 ●	●	●	●	●

Legend:

● offers the benefit

◐ partially offers the benefit

○ depends on authenticator

🟩 better than passwords

🟥 worse than passwords

Figure 2: *Comparison of standard passwords with FIDO 1FA/2FA using a modified version of the framework proposed by Bonneau et al. (2012)*

i. Threats to Validity

As this work is mostly a literature review, the limitations of the results above are mainly dependent on said literature.

As seen in table 1, the first phase returned most of the FIDO2 sources. This is due to the fact that these are mostly media outlets like technical blogs [18, 24, 5, 30, 29] or from the official FIDO Alliance website [12, 11, 10].

Table 1: *Overview of literature used in this paper.*

Phase	Foundations	FIDO Framework
Phase 1	5	8
Phase 2	6	6
Phase 3	4	2

As of today, there is rather little scientific research regarding FIDO2 specifically. In fact, most of the citations on FIDO2 are either technical documentation (provided by the FIDO Alliance), whitepapers or blog posts by security professionals. Although these sources do not necessarily follow the scientific method, they are still very useful in getting insights into how FIDO2 works.
It is for the same reason that only the conceptual security of the whole framework could be evaluated, those concepts however are based on mature and well-tested techniques like public-key cryptography and challenge-response authentication, so the results on these parts can still be considered sound.
The main part of scientific research focuses on the usability or acceptability of the system.

VI. Conclusion

FIDO2 shows some very good potential to solve many of todays most pressing problems regarding password-based authentication. It is a open framework, based on an open specification by the W3C, supported by browser and operating system vendors as well as many application providers. This broad support could, step by step, introduce many people to new concepts of authentication.
But the biggest factor in whether or not FIDO2 will actually become successful will remain user acceptance. For now, most of these users have not yet been directly affected by credential leaks, which lowers the perceived necessity for any changes, especially such breaking ones. Or, as security researcher Troy Hunt put it in an article called "Here's Why [Insert Thing Here] Is Not a Password Killer" [19]:

> *Every single solution I've seen that claims to "solve the password problem" just adds another challenge in its place thus introducing a new set of problems.*

For the start, FIDO2 might be a good method for secure authentication in business environments. In such a case, the amount of accounts secured is rather low, while the potential business impact in case of stolen accounts can be exceptionally high. The local IT could coordinate the roll-out and provide support, thus increasing acceptability, as shown by [23].

Future research should focus on the long-term usability of the framework, to show if and how fast users get accustomed to the new authentication concepts, how often account recovery actually becomes a problem, and what other challenges may arise in day-to-day use. Additionally, further research is needed regarding the security of both the protocol implementations and roaming as well as internal authenticators.

For security-conscious persons willing to invest some money and time for setting up a fast, secure and easy-to-use authentication method, FIDO2 could be a very good alternative to other single- and multi-factor authentication methods.

References

[1] Daniel V. Bailey, Markus Dürmuth, and Christof Paar. "Statistics on Password Re-use and Adaptive Strength for Financial Accounts". In: *Security and Cryptography for Networks (SCN)* 8642 (2014), pp. 218–235.

[2] Joseph Bonneau et al. "The quest to replace passwords: a framework for comparative evaluation of web authentication schemes." In: *IEEE Symposium on Security and Privacy* (2012), pp. 553–567.

[3] Christiaan Brand et al. *Client to Authenticator Protocol (CTAP)*. FIDO Alliance, Jan. 2019. URL: https://fidoalliance.org/fido2/fido2-web-authentication-webauthn/ (visited on 04/07/2020).

[4] Léonie Brandt. *Statista Befragung Cybersecurity und Cloud 2018*. Oct. 2018. URL: https://de.statista.com/statistik/studie/id/58204/dokument/cybersecurity-und-cloud/ (visited on 05/09/2020).

[5] Jerrod Chong. *10 Things You've Been Wondering About FIDO2, WebAuthn, and a Passwordless World*. Aug. 2018. URL: https://www.yubico.com/blog/10-things-youve-been-wondering-about-fido2-webauthn-and-a-passwordless-world/ (visited on 04/20/2020).

[6] Sanchari Das et al. "A qualitative study on usability and acceptability of Yubico security key". In: *Proceedings of the 7th Workshop on Socio-Technical Aspects in Security and Trust*. Association for Computing Machinery, 2018, pp. 28–39.

[7] Phil Dunkelberger. "FIDO2 puts biometrics at heart of web security". In: *Biometric Technology Today* (2018), pp. 8–10.

[8] Duo Security. *Share of internet users in the United States who use two-factor authentication in 2013 and 2017*. Nov. 2017. URL: https://www.statista.com/statistics/789473/us-use-of-two-factor-authentication/ (visited on 05/23/2020).

[9] Jon D. Elhai and Brian J. Hall. "Anxiety about internet hacking: Results from a community sample". In: *Computers in Human Behavior* 54 (2016), pp. 180–185.

[10] FIDO Alliance. *History of FIDO Alliance*. URL: https://fidoalliance.org/overview/history/ (visited on 05/26/2020).

[11] *FIDO2: Web Authentication (WebAuthn)*. URL: https://fidoalliance.org/fido2/fido2-web-authentication-webauthn/ (visited on 04/07/2020).

[12] *FIDO2: WebAuthn & CTAP*. URL: https://fidoalliance.org/fido2/ (visited on 04/01/2020).

[13] Sean Gallagher. *Facebook apps logged users' passwords in plaintext, because why not*. Mar. 2019. URL: https://arstechnica.com/information-technology/2019/03/facebook-developers-wrote-apps-that-stored-users-passwords-in-plaintext/ (visited on 05/25/2020).

[14] Hidehito Gomi, Bill Leddy, and Dean H. Saxe. *Recommended Account Recovery Practices for FIDO Relying Parties*. Feb. 2019. URL: https://fidoalliance.org/recommended-account-recovery-practices/ (visited on 05/14/2020).

[15] Paul A. Grassi, Michael E. Garcia, and James L. Fenton. "Digital Identity Guidelines". In: *Special Publication (NIST SP)* 800-63-3 (2017).

[16] Troy Hunt. *';–have i been pwned?* URL: https://haveibeenpwned.com/ (visited on 04/15/2020).

[17] Troy Hunt. *86% of Passwords are Terrible (and Other Statistics)*. May 2018. URL: https://www.troyhunt.com/86-of-passwords-are-terrible-and-other-statistics/ (visited on 04/29/2020).

[18] Troy Hunt. *Beyond Passwords: 2FA, U2F and Google Advanced Protection*. Nov. 2018. URL: https://www.troyhunt.com/beyond-passwords-2fa-u2f-and-google-advanced-protection/ (visited on 03/25/2020).

[19] Troy Hunt. *Here's Why [Insert Thing Here] Is Not a Password Killer*. Nov. 2018. URL: https://www.troyhunt.com/heres-why-insert-thing-here-is-not-a-password-killer/ (visited on 03/25/2020).

[20] Troy Hunt. *Password Strength Indicators Help People Make Ill-Informed Choices*. July 2017. URL: https://www.troyhunt.com/password-strength-indicators-help-people-make-dumb-choices/ (visited on 04/29/2020).

[21] Troy Hunt. *The only secure password is the one you can't remember*. Mar. 2011. URL: https://www.troyhunt.com/only-secure-password-is-one-you-cant/ (visited on 04/28/2020).

[22] *Is someone spying on you?* URL: https://sec.hpi.de/ilc/search (visited on 04/15/2020).

[23] Juan Lang et al. "Security Keys: Practical Cryptographic Second Factors for the Modern Web". In: *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2017, pp. 422–440.

[24] Felix von Leitner. Oct. 2019. URL: https://blog.fefe.de/?ts=a3695c14 (visited on 04/20/2020).

[25] Sanam Ghorbani Lyastani et al. "Better managed than memorized? Studying the Impact of Managers on Password Strength and Reuse". In: *USENIX Security Symposium* 27 (2018), pp. 203–220.

[26] Sanam Ghorbani Lyastani et al. "Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication". In: *41st IEEE Symposium on Secruity and Privacy (IEEE S&P)*. Oakland Association, 2020.

[27] Robert McMillan. *The World's First Computer Password? It Was Useless Too*. Jan. 2012. URL: https://www.wired.com/2012/01/computer-password/ (visited on 05/17/2020).

[28] Yogesh Mehta. *Building a world without passwords*. May 2018. URL: https://www.microsoft.com/security/blog/2018/05/01/building-a-world-without-passwords/ (visited on 03/25/2020).

[29] Ken Mingis, Juliet Beauchamp, and Lucas Mearian. *FIDO Alliance and the future of passwords*. Mar. 2020. URL: https://www.computerworld.com/article/3530435/fido-alliance-and-the-future-of-passwords.html (visited on 04/01/2020).

[30] Alfred Ng. *Google looks to leave passwords behind for a billion Android devices*. Feb. 2019. URL: https://www.cnet.com/news/google-looks-to-leave-passwords-behind-for-a-billion-android-devices/ (visited on 04/01/2020).

[31] Jakob Nielsen. *Usability Engineering*. London: Academic Press, 1993.

[32] OWASP. *A2:2017-Broken Authentication*. 2017. URL: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication (visited on 05/26/2020).

[33] S. Rafique et al. "Web application security vulnerabilities detection approaches: A systematic mapping study". In: *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. 2015, pp. 1–6.

[34] Dawn M. Turner. *Digital Authentication - the basics*. Aug. 2016. URL: `https://www.cryptomathic.com/news-events/blog/digital-authentication-the-basics` (visited on 05/23/2020).

[35] Steven Vaughan-Nichols. *Apple joins FIDO Alliance, commits to getting rid of passwords*. Feb. 2020. URL: `https://www.zdnet.com/article/apple-joins-fido-alliance-commits-to-getting-rid-of-passwords/` (visited on 05/23/2020).

[36] *Web Authentication: An API for accessing Public Key Credentials*. Mar. 2019. URL: `https://www.w3.org/TR/webauthn/` (visited on 04/01/2020).

[37] *Web Authentication API*. URL: `https://developer.mozilla.org/de/docs/Web/API/Web_Authentication_API` (visited on 04/15/2020).

[38] Monica Whitty et al. "Individual Differences in Cyber Security Behaviors: An Examination of Who Is Sharing Passwords". In: *CYBERPSYCHOLOGY, BEHAVIOR, AND SOCIAL NETWORKING* 18.1 (2015), pp. 3–7.

[39] *YubiKey 5 NFC*. Feb. 2019. URL: `https://www.yubico.com/product/yubikey-5-nfc` (visited on 05/14/2020).