CISC 866 FIDO2 Authentication

Nicholas Dionne (24kg1@queensu.ca)

ABSTRACT

FIDO2 (Fast Identity Online 2) is a significant advancement in a password-less approach to authentication by leveraging public key cryptography. This paper explores FIDO2's architecture and core specifications: WebAuthn and CTAP2; furthermore, the paper evaluates three implementation options for FIDO2 by analyzing their security and usability trade-offs: Platform Authenticators, Roaming Authenticators, and Passkeys. Through a critical and comparative review of three papers that examine FIDO2 through various studies on usability and user perception, this paper dives deeper into the challenges posed by FIDO2, such as account recovery, implementation complexity, and user education. Lastly, after highlighting potential solutions and insight into the challenges posed in the adoption of FIDO2, this work covers new unexplored use cases and future predictions for FIDO2 within the next decade.

1 INTRODUCTION

This project will aim to approach the subject of FIDO2 (Fast Identity Online 2) on a fundamental level, hoping to address both its architecture and specifications, namely WebAuthn (Web Authentication API) and CTAP2 (Client To Authenticator Protocol 2). Moreover, it will compare and contrast various implementation options for FIDO2 and provide an insightful analysis and review of the three research papers involving FIDO2 and/or passkey, ending with a comparison of the aforementioned papers.

FIDO2 is an important step in a digital environment that requires secure authentication methods. FIDO2 helps address this requirement by enabling password-less authentication, which in turn reduces the risks associated with traditional password-based systems, such as phishing. The importance of FIDO2 is not only in its potential to enhance security but also in its potential to improve the user experience while reducing their involvement and simplifying the authentication process. The process is simplified by leveraging public key cryptography and eliminating the need for shared secrets such as passwords. Lastly, this project will strive to delve into and accurately inform on FIDO2 and its potential to shape the future of secure online authentication.

2 FIDO2 ARCHITECTURE

Utilizing the resources from the two sources, FIDO Alliance and Microsoft: *FIDO Authentication: A Passwordless Vision*, and *What is FIDO2?* An attempt will be made to define FIDO2; FIDO2 is a framework that aims to improve online security through password-less authentication using public-key cryptography [1, 9]. There are to primary components to FIDO2: WebAuthn and CTAP2, together these ensure secure communication between the users' devices and the services they utilize [1, 9].

To utilize FIDO2, which is fundamentally an MFA (Multi-Factor Authentication) method, a user, when registering for a service, may provide an email or username initially; however, users won't provide a password [1, 9]. Instead, services with FIDO2 will require that users register a method for authentication; this can be almost any device provided that it can store the corresponding information [1, 9]. For example, utilizing a smartphone (or laptop, security key) a user could register the device as there authentication method with a passkey such as face ID, biometrics or a pin [1, 9]. Following the registration, a private key and public key pair will be generated on the chosen device where the private key will be permanently stored, and the corresponding public key will be sent to be stored along with your account on the service the user has registered to [1, 9]. When logging in using FIDO2, the corresponding service sends a challenge to the user's device the user then simply utilizes their device and passkey to then authenticate their intentions this signs the challenge sent by the service with the private key and sends it back where the service verifies the signed response utilizing the stored public key completing the login attempt completely password-less [1, 9].

2.1 WebAuthn

WebAuthn (Web Authentication API) can be defined by the following sources by the FIDO Alliance and W3C *FIDO2: Web Authentication (WebAuthn)*, and *Web Authentication: An API for accessing Public Key Credentials Level 1*. WebAuthn is an API that enables web applications to create and use public key-based credentials for the authentication of users [2, 5]. WebAuthn is a core component of FIDO2 as it attempts to provide a standardized framework for password-less authentication on the internet [2, 5].

WebAuthn defines a cryptographic protocol called the WebAuthn/FIDO2 protocol [5]. This establishes any communication between a relying party's server utilizing WebAuthn and an authenticator such as a user-owned device [5]. The protocol operates by having the relying party issue a unique challenge, a typically 16-byte long random sequence of numbers formed for the express reason of acting as a counter to replay attacks by having the challenge present itself as evidence of a quote-on-quote fresh authentication exchange [5]. The challenge is then received along with other following input data by the authenticator after traveling through HTTPS, the relying party's web application, WebAuthn API, and a device/platform-specific path such as USB (Universal Serial Bus) or NFC (Near Field Communication); the latter essentially envelopes near forms of wireless communication that your client device can utilize to contact the authenticator device [5]. The challenge received by the authenticator is then digitally signed by the authenticator using the private key corresponding to the relying party's public key[5]. The following response includes the signed challenge and related authenticator data; however, if the user is registering an authenticator with an account, then the authenticator will also include an attestation statement, which includes the necessary information about the authenticator device and confirming the device's strong security capabilities [5]. The authenticator response that was sent to the client is then sent to an object and passed to the relying party's script by the client [5]. Following this, the relying party's server receives the response from the client and

authenticates the signature utilizing its stored public key; if registering, the relying party server may do additional verifications on the attestation statement utilizing FIDO services before finishing the authentication and storing information for future authentication practices [2, 5].

2.2 CTAP2

CTAP2 (Client To Authenticator Protocol 2) can be defined by the following two sources provided by the FIDO Alliance: *FIDO2: Web Authentication (WebAuthn)*, and *Client to Authenticator Protocol (CTAP)* [2, 6]. CTAP2 is a protocol within FIDO2 that focuses on communication between a roaming authenticator and a client [2, 6]. Roaming authenticators include devices such as smartphones, which can be used to authenticate other platforms, such as PCs. Additionally, WebAuthn and CTAP2, while working closely in FIDO2, have some overlap as they are hard to extricate from one another and analyze individually, which may be seen in the following overview of CTAP2 below [2, 6].

CTAP2 consists of many features that FIDO2 and WebAuthn rely on for authentication [6]. One of these is known as transport bindings; these are transport protocols that help in communication, such as USB, NFC, and Bluetooth [6]. These specific bindings in the CTAP2 specification ensure secure communication between the authenticator and the client [6]. CTAP2 also features hybrid transport protocols, which can be utilized during the authentication process to allow the authenticator to access a camera on something such as a smartphone to scan a QR code provided by the client, which will then share a secret key between the authenticator and client this creates a tunnel which is then utilized for the rest of the regular authentication process [6]. Once a secure connection is made between the authentication device and the client, the client utilizes CTAP2 commands to prompt the corresponding response required by the relying party's WebAuthn authentication processes; some of these commands are authenticationGetAssertion (signs a challenge during authentication), and authenticatorMakeCredential which prompts the authenticator to generate a new public-key pair and store the private key for the registration process of a new user [6]. The CTAP2 specification also provides a large set of error codes to provide insight into any operation failures [6]. Moreover, authenticators utilizing CTAP2 are recommended to consider backward compatibility by continuing support for CTAP1 and U2F to ensure a level of availability for users, as some websites or applications may still operate on these [6]. Lastly, there remain two key components to CTAP2: User Presence (UP) and User Verification (UV), which were foreshadowed a little beforehand [6]. UP is the confirmation that the user is located/present physically at the authentication device, while UV is the verification that the user is who they claim to be, which is an individual with the authority to take actions utilizing the authentication device [6]. CTAP2 is a method of securely communicating and interpreting commands between the client and authentication device to aid in the completion of the tasks required by the WebAuthn specification in FIDO2 [2, 6].

3 IMPLEMENTATION OF FIDO2

The following resources from FIDO Alliance, USENIX, Microsoft, ACM, and W3C and their papers and/or articles: *Web Authentication: An API for accessing Public Key Credentials Level 1*, *What is FIDO2?*, *FIDO2 the Rescue? Platform vs. Roaming Authentication on Smartphones*, *User Perceptions of the Usability and Security of Smartphones as FIDO2 Roaming Authenticators*, *Replacing Password-Only Authentication with Passkeys in the Enterprise*, and *Passkeys* were utilized to form the evaluations below [3–5, 9–11]. There are three popular implementations of FIDO2 for authentication: Platform Authentication, Roaming Authentication, and Passkeys [3–5, 9–11].

3.1 Platform Authenticators

Platform Authenticators store their private keys on the device itself [2, 11]. The private key, in this case, never leaves the device, much like with roaming authenticators [2, 11]. However, because of this fact, platform authenticators come with some drawbacks, such as users only being able to authenticate from devices that store the private key; in other words, the platform authenticator itself [2, 11]. When it comes to security, there is also the concern that a platform authenticator relies entirely on the security of that platform to protect the private key, while the platform itself may also not belong solely to the user or always be on the user's person, such as with roaming authenticators [2, 11]. On the other hand, platform authenticators can often be a seamless user experience as the user does not need to remember any particular credentials other than the platforms, and the user does not need to consistently carry and or secure another device, such as with roaming authentication where the extra external device can be easily lost or even stolen [2, 11].

3.2 Roaming Authenticators

Roaming Authenticators are external devices that store the private key and that the user carries with them for authentication purposes [2, 10, 11]. This device, for example, can be a USB security key or a smartphone that utilizes NFC or Bluetooth [2, 10, 11]. This comes with the flexibility to use the same roaming authenticator for the authentication of many different devices in contrast to platform authenticators [2, 10, 11]. The private key, as with platform authenticators, never leaves the authenticator device, leaving it less vulnerable to attacks [2, 10, 11]. Some of the primary disadvantages to this implementation include users always needing to have their roaming authenticator device with them to authenticate; otherwise, they're out of luck [2, 10, 11]. Moreover, the need to always have the device at hand can be problematic if certain scenarios present themselves, such as losing or forgetting the device [2, 10, 11].

3.3 Passkeys

Passkeys seem to leverage both prior methods of implementation: platform authenticator and roaming authenticator [3, 4, 9]. Passkeys offer two distinct methods of implementation: device-bound passkeys and cloud-syncing passkeys [3, 4, 9]. Device-bound passkeys especially function as roaming authenticators [3, 4, 9]. The passkey is stored on a device, such as a smartphone, and is not synced to the cloud and is, therefore, not readily accessible

outside of that particular device [3, 4, 9]. On the other hand, cloud syncing passkeys are an approach similar to platform authenticators; however, a synced passkey is synchronized across a user's choice of devices through a cloud service such as iCloud Keychain [3, 4, 9]. The use of cloud services to provide passkeys across multiple devices allows for the platform authenticator implementation to make up for its fatal flaw of a lack of availability [3, 4, 9]. This newfound availability may, however, come at the cost of typical users having to remain in a single cloud ecosystem or applying multiple ecosystems across differing devices [3, 4, 9]. One of the primary advantages of the other FIDO2 implementation options is that the private key never leaves the authenticator device it was created on, a cloud-synced passkey; however, introduces these private keys to the cloud and potentially multiple other devices, and this will inevitably open the private key up to many more potential attacks despite how many more protections are formed around the cloud environment [3, 4, 9].

4 RESEARCH PAPER REVIEW

4.1 Paper 1

The following is a FIDO2 paper presented at an IEEE conference on privacy and security: *Challenges with Passwordless FIDO2 in an Enterprise Setting: A Usability Study* [8]. This paper presents a usability study of FIDO2 that examines the challenges of integrating password-less authentication in an enterprise environment [8]. The usability study involved one hundred and eighteen IT professionals with FIDO2 experience, a point of commonalty among those selected is that they shared significant concerns over account recovery and with the overall complexity of implementing FIDO2 into current systems [8]. Furthermore, the paper specifically details that there's a need for the FIDO2 community to confirm and come up with solutions to the practical challenges involved in the adoption of FIDO2 for larger, more complex enterprise environments [8]. One concern with the usability study is that the study takes into consideration enterprise use cases that may not fully represent challenges posed by other implementations and, therefore, potentially mislead further development of FIDO2 [8]. Similarly the professionals recruited for the study may have introduced bias as they targeted individuals more familiar and receptive to the idea of FIDO2 [8]. The implications for real-world systems in regard to this study is that FIDO2 requires an improved method of account recovery for users, followed by guidance and an address towards the issue of integration complexity for large enterprise systems [8].

4.2 Paper 2

This FIDO2 paper was presented at an IEEE conference on security and privacy: Is *FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication* [7]. This paper is a comparative usability study that directly compares the usability of FIDO2 authentication using a security key and more traditional password-based authentication [7]. The study involves ninety-four participants who had their experiences utilizing both methods measured through tasks and surveys; the study's results found that users were willing to switch to FIDO2 [7]. However, the study brought forward the following concerns, first and foremost being the loss of an authenticator and the need for the subsequent account recoveries [7]. The study focuses on the use of a security key as a chosen authenticator, which may affect the results of the study as FIDO2 offers a multitude of authentication options; maybe participants would be less concerned at the loss of their authentication device with the knowledge that due to the utilization of passkey, they may have access to all the relevant authentication information through their cloud environment of choice [7]. The study also utilizes mock websites for the controlled lab setting the study takes place in, which may not fully represent the complexities of a real-world FIDO2 deployment by a large enterprise [7]. This study, in particular, offers some valuable insights into how to promote a wider user adoption of FIDO2, which may be one of the most overwhelming challenges in the real-world application of FIDO2 [7].

4.3 Paper 3

This FIDO2 paper was presented at a SOUPS conference on usable privacy and security: *User Perceptions of the Usability and Security of Smartphones as FIDO2 Roaming Authenticators* [10]. This paper specifically studied user perceptions of FIDO2 when using smartphones as their roaming authenticators and directly compared the approach to traditional password-based authentication utilizing data from ninety-seven participants [10]. The study had participants log into a mock banking site with a smartphone as a FIDO2 roaming authenticator using a prototype called Neo or a traditional password over two weeks [10]. The results showed that the participants of the study found Neo less usable than passwords, especially during the registration and setup phases [10]. The paper brought forth the following concerns regarding FIDO2 phone availability, account recovery, and setup complexity, and some of the participants were confused or concerned due to the lack of MFA (Multi-Factor Authentication) [10]. Analyzing the study, it can be seen that the extended period of testing certainly added to their findings [10]. A point for consideration is that the initial setup of a FIDO2 authenticator can take an extended period of time and be frustrating for new users; this, in turn, could introduce a bias for what individuals are more accustomed to, such as password-based authentication [10]. Neo, which is the studies prototype for a FIDO2 authenticator on smartphones, could be limiting the findings as well due to a complex or frustrating implementation of FIDO2 compared to other methods [10].

4.4 Comparison

The following is a comparative analysis of papers 1 [8], 2 [7], and 3 [10]. The overall methodology of the papers differ: papers 1 and 2 conduct a lab-based study, which is indicated by the well-controlled environment it takes place in that can not quite capture the complete complexity of the real world, while paper 3 utilizes a longitudinal study that observes the participants user experiences over a longer period of time [7, 8, 10]. All three of the papers, even with their differing methodologies, arrived at similar conclusions regarding the concerns users had for FIDO2, the most prominent two being account recovery and the potential loss of an authenticator such as smartphone or security keys which are roaming authenticators, the user needs to constantly keep on hand to access their accounts [7, 8, 10]. However, there are some differences in

the paper's findings regarding the perceived usability of FIDO2 in comparison to passwords; paper 2 indicated that FIDO 2 is a more easily usable medium for authentication, while papers 1 and 3 indicated the opposite, the contrast in findings could likely be attributed to the differing types of authenticators used in the studies; and the different types of participants and environments, for example, paper 1 considers an enterprise environment with FIDO2 professionals as the participants the professionals are considering the implementation of FIDO2 for the entire organization and not as individuals [7, 8, 10]. In contrast, in paper 3, users likely have prior experiences losing their phones, hence the corresponding worry; the perplexing aspect is paper 2 as it would not be a stretch to suggest users would fear losing a security key as they would a phone or their car keys [7, 10]. Collectively, these papers offer important recommendations for improving FIDO2 that could impact the real-world implementation of FIDO2. The papers recommend improving user education, which could help remove knowledge gaps or misconceptions that lower user acceptance and understanding of FIDO2 [7, 8, 10]. Furthermore, the papers suggest a more robust and user-friendly approach to account recovery mechanisms in FIDO2 [7, 8, 10]. Lastly, the papers almost unanimously recommend action be taken for ease of use and integration of FIDO2 into existing multi-factor authentication systems [7, 8, 10]. Take note that a solution to the concern of the loss of roaming authenticators, such as smartphones, already exists and may considerably improve user impressions of FIDO2; passkeys allow for FIDO2 user credentials to be uploaded to the cloud environment of a user choosing and utilizing face id or a pin as a passkey gives the user access to their credentials in that cloud environment [3, 4].

5 REFLECTION

5.1 Insights

FIDO2 is a leap in authentication that helps to address many of the issues associated with password-based systems. However, despite its strong security benefits, FIDO2 faces challenges such as Account recovery, implementation complexity, and user education. Account recovery is an issue that stands across all FIDO2 implementation options; for example, the loss of a roaming authenticator can lock a user out of their account for the foreseeable future. This issue is somewhat mitigated by leveraging cloud syncing with a passkey, but this then introduces those important authentication credentials outside of the authentication device and also makes the user reliant on a particular cloud ecosystem. Implementation complexity, as in paper 1 [8] poses difficulties to large enterprises that wish to implement FIDO2 into their existing systems; this can be for a variety of reasons, compatibility issues, the need for trained IT staff, and the difficulties involved in replacing any legacy authentication methods. Lastly, many users who are participants in papers 2 [7] and 3 [10] who are not professionals in FIDO2 lack the necessary knowledge related to password-less authentication such as FIDO2; in other words, these users have yet to build any trust with this method of authentication and some of the initial stages of the registration and setup phases of FIDO2 may prove challenging for the technologically inept.

FIDO2 covers many use cases for consumers and enterprise applications; however, some remain unexplored. IoT (Internet of Things) FIDO2 could be utilized in this ever-expanding environment with overall poor security practices to enhance the security of home assistants or medical devices. One of the more important things to consider is standardizing cross-ecosystem functionality for passkeys that utilize cloud ecosystems. This could considerably reduce the fragmentation of a user's credentials across multiple cloud ecosystems while improving usability for users with more diverse devices. There's also a case to make for offline applications of FIDO2 authentication, which seems to have gained minimal support as FIDO2 is typically utilized for online web authentication with WebAuthn, considering this use case could allow FIDO2 to be an instrumental aspect of security even in offline environments where critical infrastructure is kept from any external networks.

5.2 Future of FIDO2

Predictions for the next five to ten years could include a definite steep increase in the adoption of FIDO2 or similar password-less approaches, given that the challenges involved with FIDO2 are acknowledged and improved upon. Particularly training and informing the public on FIDO2 as an approach and providing standardized methods for a large enterprise to smoothly implement FIDO2 across their existing systems and replace their legacy authenticators. In the case that FIDO2 is widely adopted, passkeys are likely to become the dominant form of implementation for FIDO2 due to the implied ease of use and ability to synchronize across multiple devices. The use of passkeys is even more likely when taking into account its potential to mitigate users' aversion to adopting FIDO2 due to cases such as losing a roaming or even platform authenticator along with all of those ever-important account credentials. As mentioned previously, for a potential use case, it could be foreseen that a standardized framework that allows for ease of interpretation across cloud ecosystems is essential in keeping users from getting locked to specific platforms, such as Apple or Microsoft, when utilizing passkeys.

5.3 Conclusion

In conclusion, FIDO2 is set to change the way that users authenticate and offers outstanding security features through WebAuthn and CTAP2, as well as impressive usability past its initial stages. However, the adoption of FIDO2 on the world stage hinges on a few key challenges: account recovery, implementation complexity, and user education. By expanding on its use cases, FIDO2 can reach new unexplored potential that improves on its current utility and, therefore, extends its widespread adoption. Lastly, over the next decade, if FIDO2 compensates for its key challenges adequately, it is likely that it will evolve into a universally accepted authentication standard that paves the way for password-less authentication and adopting a new standard of keeping secrets, not sharing them.

REFERENCES

[1] FIDO Alliance. 2024. *FIDO2: Moving the World Beyond Passwords.* https://fidoalliance.org/fido2/ Accessed: 2024-10-31.
[2] FIDO Alliance. 2024. *FIDO2 Web Authentication (WebAuthn).* https://fidoalliance.org/fido2-2/fido2-web-authentication-webauthn/ Accessed: 2024-10-31.
[3] FIDO Alliance. 2024. *Passkeys.* https://fidoalliance.org/passkeys/ Accessed: 2024-10-31.
[4] FIDO Alliance. 2024. *Replacing Password-Only Authentication with Passkeys in the Enterprise.* https://fidoalliance.org/white-paper-replacing-password-only-

authentication-with-passkeys-in-the-enterprise/ Accessed: 2024-10-31.

[5] Dirk Balfanz, Alexei Czeskis, Jeff Hodges, J.C. Jones, Michael B. Jones, Akshay Kumar, Angelo Liao, Rolf Lindemann, and Emil Lundberg. 2019. Web Authentication: An API for accessing Public Key Credentials Level 1. W3C Recommendation. https://www.w3.org/TR/2019/REC-webauthn-1-20190304/ Accessed: 2024-10-31.

[6] John Bradley, Michael B. Jones, Akshay Kumar, Rolf Lindemann, Johan Verrept, and David Waite. 2024. Client to Authenticator Protocol (CTAP) Version 2.2 Review Draft. FIDO Alliance Review Draft. https://fidoalliance.org/specs/fido-v2.2-rd-20241003/fido-client-to-authenticator-protocol-v2.2-rd-20241003.html Accessed: 2024-10-31.

[7] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. 2020. Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication. In *2020 IEEE Symposium on Security and Privacy (SP)*. 268–285. https://doi.org/10.1109/SP40000.2020.00047

[8] Michal Kepkowski, Maciej Machulak, Ian Wood, and Dali Kaafar. 2023. Challenges with Passwordless FIDO2 in an Enterprise Setting: A Usability Study. In *Proceedings of the IEEE Conference on Security and Privacy*. https://arxiv.org/pdf/2308.08096v2 arXiv preprint.

[9] Microsoft. 2024. *What is FIDO2?* https://www.microsoft.com/en-us/security/business/security-101/what-is-fido2?msockid=07580ff4d8ce6d8d2d361e02d9cf6c29#examplesoffido2 Accessed: 2024-10-31.

[10] Kentrell Owens, Olabode Anise, Amanda Krauss, and Blase Ur. 2021. User Perceptions of the Usability and Security of Smartphones as FIDO2 Roaming Authenticators. In *Proceedings of the Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX Association, Virtual Conference. https://www.usenix.org/conference/soups2021/presentation/owens

[11] Leon Würsching, Florentin Putz, Steffen Haesler, and Matthias Hollick. 2023. FIDO2 the Rescue? Platform vs. Roaming Authentication on Smartphones. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM, Hamburg, Germany, 1–16. https://doi.org/10.1145/3544548.3580993

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009