Should We Rush to Implement Password-less Single Factor FIDO2 based Authentication?

Fatima Alqubaisi
Technological Innovation
Zayed University
Abu Dhabi, UAE
201607523@zu.ac.ae

Ahmad Samer Wazan
Technological innovation
Zayed Univeristy
Abu Dhabi, UAE
Ahmad.Wazan@zu.ac.ae

Liza Ahmad
Technological innovation
Zayed Univeristy
Abu Dhabi, UAE
Liza.Ahmad@zu.ac.ae

David W Chadwick
University of Kent
UK
d.w.chadwick@kent.ac.uk

***Abstract*— Fast Identity Online (FIDO) Alliance and W3C have defined a set of specifications (called FIDO2) that allows a user to replace the password based authentication system. However, none of the high profile web sites have implemented FIDO2 yet as password-less single factor (SF) authentication (password-less SF). In this paper, we analyze the set of factors that make websites reluctant to adopt password-less FIDO SF authentication. We start by comparing the threat models of password-less FIDO SF authentication with password-based SF authentication. Our analysis shows that although password-based authentication is less secure than FIDO SF authentication, other factors related to the usability of FIDO security keys and FIDO based authentication system, the non-consideration of enterprise requirements and the lack of specifications regarding account recovery/deletion and suspension are the main obstacles to the adoption of password-less FIDO SF authentication.**

Keywords—UAF, FIDO2, WebAuthn, Password based authentication, threat model

I. INTRODUCTION

Today, the trend on the web is to move away as quickly as possible from password-based authentication by adopting stronger and more secure authentication that is based on asymmetric cryptography. A set of standards have been defined by the FIDO alliance and W3C to specify public-key based protocols that help web users authenticate securely to online services.

The specifications of FIDO2 protocol are implemented by different types of physical authenticators called security Keys such as Yubikey from Yubico and Titan from Google. Using a security key, a web user sends its public key to online services during the registration process. The corresponding private key is generated inside the physical security key, that is designed to not export the private keys from the hardware token under any circumstances.

The authentication process takes place when the online web service sends a string challenge to the web user who will authorize the generation of a digital signature (e.g. by pressing a button on the security key) upon receiving this challenge and sends it back to the online service. This method of authentication provides a protection against phishing attacks because the generated public/private key pair are scoped to the domain name of the online service.

However, when we check the current implementations of the most high-profile websites such as Gmail, Facebook and Twitter, we notice that they integrated FIDO based authentication only as a second factor authentication, and not as a single factor one. It means that web users must provide first their password to authenticate themselves, and then use their FIDO security key to proceed to their accounts. At the same time, these websites are still proposing one factor password-based authentication, thereby giving hackers or any malicious software a chance to steal the users' passwords. Our objective in this paper is to study the different obstacles that face a large scale adoption to password-less FIDO SF authentication. When it's possible we give some recommendations regarding these obstacles.

We start by showing why password-less FIDO SF authentication is more secure than password-based SF authentication. We compare the threat models of both authentication systems. We show that the security of FIDO based SF is not enough to cause the adoption of this technology. Our analysis is based on observations and participations in the FIDO Alliance discussion List (fido-dev) that is used to collect real world adoption experience by the FIDO Alliance [1]. We discuss here different elements reported in this list and we give our analysis to some of them. It should be noted that our objective is not to give an exhaustive list of the obstacles to password-less FIDO SF authentication, but only to discuss the most significant elements that hinder its adoption.

The rest of this paper is structured as follows. Section 2 overviews the FIDO based standards and their historical evolution. We will present the UAF, U2F and FIDO2 protocols. Section 3 exposes and analyses the threat model of password based authentication. Section 4 presents the threat model of FIDO based SF authentication and compares it to the password-based SF authentication model. Finally, in section 5 we discuss our results and give our conclusions.

II. FAST IDENTITY ONLINE (FIDO)

In July 2012, FIDO Alliance was created in order to develop a set of strong authentication standards whose objective is to reduce the use of passwords on the web. The basic idea consists of replacing userids and passwords with a public/private key

978-1-7281-9789-0/20/$31.00 ©2020 IEEE

pair, generated using a hardware token for each web service requested by web users.

The hardware token, called also the security key or authenticator, is used to generate the asymmetric key pairs, and is designed to store the private keys securely, with no possibility to export them from the hardware token. In addition, the authenticator is designed to avoid the storage of personal information about the web user.

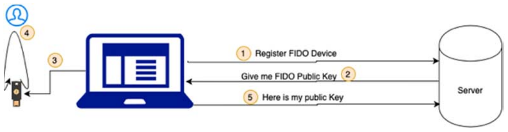


Figure 1. FIDO registration process

Two main web transactions have been specified by the FIDO Alliance: registration and authentication. FIDO registration is quite straightforward. The process starts when the user tells the server to register his/her own FIDO device. The web server asks the user's platform to generate a FIDO key pair. The platform informs the authenticator about the request of the web server. Finally, the authenticator requests the user's approval to generate a public/private key pair for the web site. If approved, the authenticator generates the public/private key pair and asks the user's platform to send the public key to the web server.

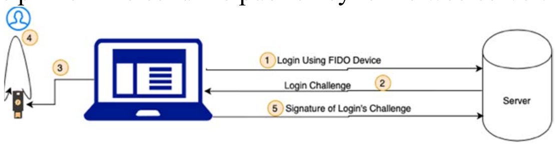


Figure 2. FIDO authentication process

The FIDO authentication process (i.e. login process) starts when the user chooses to login using a FIDO security key. The server sends a login challenge to the user's platform. The user's platform forwards the request to the user's authenticator that asks for the approval of the user. If approved, the authenticator searches for the private key corresponding to the public key of the web server (sent during the registration process) and signs the login challenge. The server checks the signature using the public key that has been stored during the registration process and lets the user login to the server if the verification of the signature is successful.

The FIDO Alliance has setup a metadata service that allows organizations deploying FIDO based authentication to verify whether the authenticators of users are certified by the FIDO Alliance, and whether they are free from known vulnerabilities. Every FIDO authenticator has an attestation key that is used by authenticators to sign any statements (i.e. Public key certificate, signature for authentication) to prove that the authenticator is certified by the FIDO Alliance. Normally, the same attestation key is shared by at least 100000 authenticators to ensure the anonymity of Authenticators' users. However, many argue that the sharing of the attestation key makes it an attractive attack target [2].

In addition, the FIDO Alliance has developed an established certification program for FIDO servers and clients to ensure that the management of the registration and the authentication processes are compatible with the FIDO standards.

Three different protocols have been defined by the FIDO alliance since 2014: Universal Authentication Framework (UAF), Universal 2nd Factor (U2F) and FIDO2/WebAuthn protocol that is adopted by the W3C.

The UAF is designed mainly for mobile devices that have an embedded authenticator certified by the FIDO Alliance. In addition to the authenticator, the user must have a certified client FIDO that will manage the authenticator through a specific set of APIs called the "Authenticator Specific Module" (ASM).

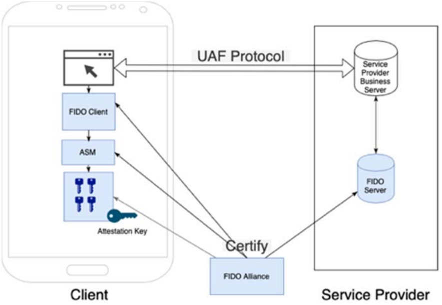
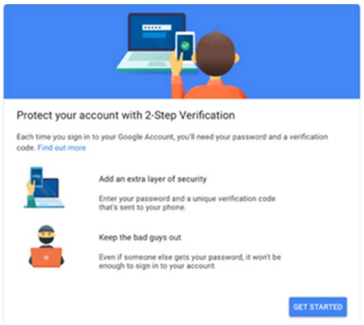


Figure 3.UAF protocol

Universal 2nd factor (U2F) is designed for online services that desire to increase the security of their existing password infrastructure [3]. This protocol allows the usage of external or embedded FIDO authenticators that can be plugged into the users' platforms. The user logs in to the server first using his/her username and password. If successful, the online service prompts the user to plug his/her FIDO device and use it as a 2nd factor. The FIDO Alliance has developed a set of U2F JavaScript APIs that have been implemented by Google Chrome, Opera and Mozilla Firefox. Many High profile websites such as Gmail and Facebook have integrated U2F protocols into their servers.



Protect your account with 2-Step Verification

Each time you sign in to your Google Account, you'll need your password and a verification code. Find out more

Add an extra layer of security

Enter your password and a unique verification code that's sent to your phone.

Keep the bad guys out

Even if someone else gets your password, it won't be enough to sign in to your account.

Figure 4.Gmail 2nd factor authentication

However, UAF and U2F were not adopted on a large scale. Other major web browsers such as Microsoft Edge and Safari didn't integrate the U2F protocol, while the UAF protocols necessitated some effort from the users to install a compatible FIDO client. As a consequence, the FIDO Alliance decided to enhance the UAF and U2F protocols in order to get more international adoption by developing with the W3C a new generation of FIDO standards that can be accepted on a large scale by the web community.

The joint effort between the FIDO Alliance and W3C resulted in the creation of the FIDO2 protocol which is composed of two sub protocols: W3C Web Authentication

(WebAuthn) that has been officially issued as a W3C web standard in March 2019, and the FIDO Client to Authenticator Protocol (CTAP) which is responsible for managing the FIDO authenticator according to the requests of the user's software client. Therefore, the scope of the FIDO Alliance has been reduced to regulate uniquely the communications between the Clients and authenticators, because from now on the regulation of communications between the servers and clients is the responsibility of W3C. This modification of the FIDO Alliance strategy has contributed to an increase in the scalability of FIDO based authentication. Now, FIDO2 is supported by Windows 10, Android platforms and all the major web browsers (Apple Safari, Mozilla Firefox and Microsoft Edge).

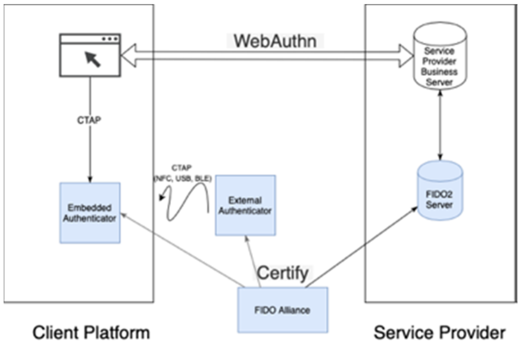


Figure 5. FIDO2 Protocol

The adoption of FIDO2 by W3C makes it possible to completely remove the use of passwords by proposing one factor FIDO2 based authentication. However, today none of the major high profile websites has adopted password-less FIDO SF authentication.

Because most of the high profile websites still use passwords as SF authentication to their services, we will compare the limits of password based SF authentication with the limits of FIDO2 based SF authentication. We will select the weakest type of FIDO2 authentication which consists of verifying the presence of a human (by pressing a button on the authenticator) to approve the registration and the authentication processes with web servers. Stronger types of approval necessitate the use of a PIN code or passing through a biometric authentication (e.g. Fingerprint, Face recognition, etc.) to authorize the registration and the authentication processes. In the following section, we detail the threat model of password-based SF authentication systems.

III. Threats to Password-based authentication

Password-based authentication was first used in the 1960s as the main authentication method to control the access to mainframe computers [6]. During this period, the authentication approach was considered as a reasonably secure approach to control the access to mainframes. However, when mainframes got connected, and with the arrival of personal computers and then the Internet, password-based authentication started to show its limits. There are many studies that have highlighted the limits of password-based authentication [4,5]. We mention here the major threats mentioned in these studies about password-based authentication:

Pass_Threat1: Weak Passwords

Password-based authentication doesn't include inherent mechanisms to incentivize users to define robust passwords.

Pass_Threat2: Local and in-transit Sniffing

Passwords can be sniffed locally by malware such as Spyware or Keyloggers. They can be sniffed by an attacker who is intercepting the remote authentication operation.

Pass_Threat3: Guessing passwords

Passwords can be guessed using either a brute force attack or dictionary based attacks.

Pass_Threat4: Password reuse

Most users use the same password on different independent online services. Creating unique password for each online service is hard for human users.

Pass_Threat5: Phishing attacks

Attackers may convince users to send their passwords to fraudulent websites that have a similar appearance to the genuine ones.

Pass_Threat6: Denial of service

Attackers can insert invalid passwords for a user, so that the user will be denied access and will not be able to login successfully any more.

Pass_Threat7: Account recovery threats

Password-based authentication doesn't include inherent recovery and reset transactions. Because most of the recovery systems are knowledge-based ones (e.g. secret questions), Attackers may use social techniques to take over the user's account.

Pass_Threat8: Unsecure password storage

The human memory is a very secure place to store passwords, but unfortunately human memory is unable to store a large number of passwords. When passwords are stored on users' platforms, malicious software can have illegal access to the database of user's passwords, or to the platform's memory to exfiltrate them to remote third parties.

Pass_Threat9: social engineering techniques

There are different social engineering techniques that may lead an attacker to infer the passwords of users, such as shoulder surfing.

Pass_Threat10: Linkability (or privacy of web users)

Two or more service providers can easily collude together to link online activities to one user.

Pass_Threat11: Non detectability

An attacker or malicious software can copy by stealth the user's passwords from the service provider's database or from user's platform. The user cannot detect the theft of his passwords in a reasonable time.

Different mitigations techniques have been proposed and implemented to cover some of these threats. For example, a site's password policy allows it to define the minimum requirements for the strength of passwords. This mitigates *pass_threat1* and *pass_threat3*. This usually must be implemented by the service provider. Salting passwords allows some random string to be added to passwords before hashing and storing them in a databases. It mitigates *pass_threat4* and *pass_threat1* because it increases the size of the user's inserted password. The Captcha mechanism mitigates *pass_threat5*. In the following table, we list mitigations to the threats against password authentication and indicate the entities that should implement them.

Table 1. Security mitigations for passwords

Mitigation	Threats	Implementers of Mitigations
Policy password	Pass_Threat1, Pass_Threat3	Service providers, enterprises
Captcha	Pass_Threat3	Service providers
Password managers	Pass_Threat8, Pass_Threat4, Pass_Threat1	Third party software, Operating systems
Anti-Phishing solutions	Pass_Threat5	Third party software, Operating systems, Web browsers
Anti-Spyware, Anti-keylogger	Pass_Threat2	Third party software, Operating systems, Web browsers
Salting passwords	Pass_Threat1 Pass_Threat4	Service providers
Encryption	Pass_Threat2	Third party software, Certification authorities, web browsers, operating systems
Email, Security questions, or SMS recovery procedures	Pass_Threat7	Service providers, Telecom operator, Email operator.
Employees Education programs	Pass_Threat9	Enterprises

Consequently, improving the security of password authentication systems depends on a set of different independent actors. This fact makes it difficult to decide about the protection status of any password authentication system because of the large number of dependent relationships that are necessary to secure the password authentication system. The security of any system can be analyzed only when a limited known number of entities intervene in its operations. Clearly, this is not the case for password authentication systems. We provide more details in our discussion section.

IV. THREATS TO ONE FACTOR FIDO2 BASED AUTHENTICATION

As mentioned earlier, we select the weakest type of FIDO2 authentication that consists of verifying only the presence of humans, by pressing the button of the authenticator to authorize the registration and the login process. We consider this authentication as one factor FIDO2 based authentication. When the user must provide a PIN code or authenticate himself using any biometric authentication method, we consider this to be 2 factor FIDO2 based authentication. For example, YubiKey 5 NFC allows users to define a PIN code or simply press a button on the authenticator to approve the registration and authentication processes.

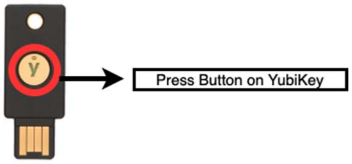


Figure 6.YubiKey 5 NFC

The FIDO Alliance provides a detailed threat analysis to the FIDO protocols here [*7*]. We consider these threats to show whether one factor FIDO2 based authentication offers stonger security than one factor password-based authentication. In the following table, we summarize the set of FIDO security goals and their related security measures [7]. For each security goal, we indicate which of the password threats are resolved by FIDO.

Table 2. Security Goals and Measures for FIDO

FIDO Security Goals	FIDO Security Measures	Resolved Password threats?
Strong User Authentication	M1: Strong protection of authentication key M2: Channel Binding that ties FIDO session credentials (e.g. FIDO assertion) to the underlying TLS session to prevent an attacker from using them in another TLS session [8]. M3: Signature counter inside FIDO authenticators that are used by service providers to detect cloned authenticators. M4: FIDO Alliance defines a list of Allowed Cryptography, which is considered to be safe. M5: Resistance to Side Channel Attacks M6: Authentication and replay-resistance M7: Scoping the generated keys to service providers (similar to a Browser's SOP) M8: Trusted path for all user interactions M9: Resistance to Remote Timing Attacks	Pass_Threat1 Pass_Threat2 Pass_Threat3 Pass_Threat4 Pass_Threat5 Pass_Threat8 Pass_Threat9
Credential Guessing Resilience	M1: Key Protection M10: Only the public key of the user is stored in the database of service provider M4: Allowed Cryptography List	Pass_Threat3
Credential Disclosure Resilience (i.e.	M1: Key Protection	Pass_Threat5

resistance to Phishing attacks)	M11: Authenticator Certification by FIDO Alliance M3: Signature counter M5: Resistance to Side Channel Attacks M9: Resistance to Remote Timing Attacks M7: Scoping the generated keys to service providers	
Unlinkability	M12: Unique Authentication Keys per service provider M13: One Attestation Key per 100000 FIDO devices	Pass_Threat10
Verifier Leak Resilience: leaks from other service providers should not permit any attacker to impersonate the user	M12: Unique Authentication Keys per service provider M10: Only the public key of the user is stored in the database of service provider M4: Allowed Cryptography List	Pass_Threat4

In addition to these security goals, FIDO devices are designed to provide security measures for DoS, replay and parallel session threats. These threats can affect any type of remote authentication system and are not only limited to password-based authentication systems.

The FIDO threat model document [7] doesn't mention explicit security goals or measures that can mitigate Pass_Threat11. However, the fact that the FIDO based authentication system is based on the use of a physical device to store the private keys of the user means that the user will be able to detect in a reasonable time the loss of his/her FIDO device. As a result, FIDO-based authentication indirectly handles this threat.

However, the FIDO based authentication system is based on the correctness of a set of assumptions to ensure the preservation of their security goals [7]:

- Assumption1: The Authenticator and its cryptographic algorithms are not subject to unknown weakness,
- Assumption2: access control mechanisms of operating systems are performing as expected,
- Assumption3: The user's platform can create a TLS connection with the remote server of the service provider,
- Assumption4: All involved elements in the FIDO operation are performing as expected,
- Assumption5: the effective assurance level of authenticators will be reduced over time as the strength of the underlying cryptographic algorithms and keys will decay with time.

While Assumption2 and Assumption3 apply also to password-based authentication systems, the other assumptions affect only the FIDO based authentication system. However, we can find similar assumptions that affect password-based authentication systems, especially with regards to the storage of passwords in service providers' databases. It is important to notice that it is not rare to detect the failure of these assumptions. For example, Wazan et al [11,12] highlight different issues regarding the TLS system. However, this will affect not only FIDO-based and Password authentication system but the entire web security system.

V. Discussion and Conclusions

The main objective of this paper is to study the different elements that hinder the adoption of password-less FIDO SA authentication. We have compared the threat models of password-based SF authentication systems and the FIDO SF authentication system. Our analysis shows clearly that FIDO SF authentication is more secure than password-based SF authentication. Indeed, our analysis shows that most of the threats of password-based authentication systems are covered by the FIDO based authentication system.

However, it should be noted that a FIDO based authentication system is not a completely secure system that can ensure the confidentiality and integrity of users' credentials. Our analysis shows only that a FIDO based authentication system has a smaller attack surface than a password-based authentication system. This comes from the fact that the protection of a password-based authentication system involves more independent and unrelated actors than the FIDO based authentication system. Indeed, a FIDO based authentication system more strictly controls the elements that intervene in the authentication and registration processes between the user and the server via its certification program. Thus, a FIDO based authentication system offers a ***semi-closed*** system rather than complete ***open*** protection system for the password authentication system.

Obviously, one factor FIDO based authentication seems the right choice for all organizations to implement because most of the assumptions on which the security of FIDO based systems are built apply also to password-based authentication systems. However, comparing the threat models of both authentication systems is not enough to make organisations rush towards the adoption of FIDO SF authentication. This is because several important features are not yet supported by the FIDO Alliance:

- Account Recovery: the FIDO Alliance doesn't specify procedures for users' account recovery. The only recommendation given to users to ensure the recovery of their accounts is to register at least two authenticators for each account. When one of the user's authenticators is lost, the other one can be used to access the user's account. We believe that this issue can be easily managed at the enterprise level, but is difficult to apply at Internet scale. Many web users will continue to use traditional recovery procedures such as knowledge-based procedures, or email-based procedures, thereby compromising the security of the FIDO authentication system. This issue necessitates some more analysis by the FIDO Alliance.

- Account Delegation: Password-based authentication systems provide an easy way to delegate access to users' accounts. In many cases, users desire to delegate access to their accounts to someone else. Under password-based authentication, a frowned-upon but frequently used method is for users to share their passwords with a person they trust. With a FIDO-based system the user has to give his/her physical authenticator to the other person. This is difficult to do if the delegated person is not physically present in the same place as the delegator. The FIDO Alliance should define a special delegation transaction that allows one user to explicitly delegate access to their accounts to someone else.

- Account deletion/Suspension: It is not clearly defined by the FIDO Alliance what procedures to follow when a user or employee loses complete control of their accounts due to the loss of their authenticator, or when a serious vulnerability is discovered in an authenticator. While the problem of lost authenticators can be handled by recovery procedures, the second one is much more difficult to handle. Indeed, from the viewpoint of web services, the user or the employee is a public key, but this public key should not be trusted anymore by the web service because it is generated by a vulnerable authenticator. The metadata service of the FIDO Alliance informs the web server to remove its trust in the vulnerable authenticator, but can the web server immediately suspend the user's access? This is a serious issue that must be studied by the FIDO Alliance. This issue can be mitigated by asking the user to use two different authenticators made by two different manufacturers. This is unlikely to happen on the scale of the Internet. Of course, it is easier to handle this issue when the user and the web service belong to the same entity (i.e. enterprise context). In this case, the enterprise can setup a verification service similar to CRLs or OCSP that are used in the domain of web PKI, to verify whether the authenticator is trustworthy or not. However, this is not implementable at the present time because it is not possible for enterprises to identify the authenticators that they distribute to their employees. According to discussions on the FIDO-Dev list, the FIDO Alliance is about to define a special format of attestation for enterprises called Enterprise Attestation that will allow them to add a serial number that will be stored in the database of each web service. However, by doing so, the FIDO Alliance will seriously compromise the security goal of *Unlinkability*. One suggestion made by the FIDO Alliance is to notify the user during the registration process that the serial number of the authenticator will be stored by the web server.

- Ease of use: The security of authenticators is the main concern of the FIDO Alliance, not their usability. The FIDO Certification Program Policy [9] defines different certification levels (level 1 to level 3+). All levels include unique security requirements but not usability requirements. This issue is left completely to the security manufacturers to handle. A recent study about the usability of YubiKey as a 2nd factor showed serious problems with the setup instructions and workflow that led the users to lock themselves out of their operating system, by thinking they had successfully enabled 2FA when they had not [10]. Clearly, the FIDO Alliance should include usability criteria in their certification programs to cope with these issues.

Finally, we think there are still several issues to address before deploying password-less FIDO SF authentication. Ideally these should be addressed in a standard way by the FIDO Alliance, rather than leaving them to individual implementors to address in proprietary ways. This is why we believe that today FIDO is used primarily as a 2nd factor because most implementors rely on the maturity of password-based authentication systems for account recovery, deletion and suspension, as well as the innate usability of password-based authentication systems.

References

[1] FIDO-Dev mailing list https://groups.google.com/a/fidoalliance.org/forum/#!forum/fido-dev

[2] https://github.com/w3c/webauthn/issues/1127

[3] FIDO U2F protocol, https://fidoalliance.org/specs/u2f-specs-master/fido-u2f-overview.html

[4] J. Bonneau, "The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords," 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, 2012, pp. 538-552.

[5] J. Bonneau and S. Preibusch, "The password thicket: technical and market failures in human authentication on the web," WEIS '10: Proceedings of the 9 Workshop on the Economics of Information Security, 2010.

[6] J. Bonneau, C. Herley, P. van Oorschot, and F. Stajano. Passwords and the evolution of imperfect authentication. Commun. ACM, 58(7):78--87, 2015

[7] FIDO Security Reference, FIDO Alliance Review Draft 28 November 2017, https://fidoalliance.org/specs/fido-uaf-v1.2-rd-20171128/fido-security-ref-v1.2-rd-20171128.html

[8] FIDO TechNotes: Channel Binding and FIDO, https://fidoalliance.org/fido-technotes-channel-binding-and-fido/

[9] FIDO Certification Program Policy Authenticator Certification, https://1nmqmp2u9dgf3jo9centu6rq-wpengine.netdna-ssl.com/wp-content/uploads/2020/01/Authenticator_Certification_Program_policy_v1.2_20190909_FINAL.pdf

[10] J. Reynolds, T. Smith, K. Reese, L. Dickinson, S. Ruoti, and K. Seamons, "A Tale of Two Studies: The Best and Worst of YubiKey Usability," in *2018 IEEE Symposium on Security and Privacy (SP)*, May 2018, pp. 872–888.

[11] Ahmad Samer Wazan, Romain Laborde, David W. Chadwick, François Barrère, Abdelmalek Benzekri, Mustafa Kaiiali, Adib Habbal Trust Management for Public Key Infrastructures: Implementing the X.509 Trust Broker in : Security and Communication Networks, Wiley, Vol. Volume 2017, (en ligne), 2017.

[12] Ahmad Samer Wazan, Romain Laborde, David W. Chadwick, François Barrère, Abdelmalek Benzekri TLS Connection Validation by Web Browsers: Why do Web Browsers still not agree? (regular paper) in : IEEE Computer Society Signature Conference on Computers, Software and Applications (COMPSAC 2017), Turin, Italie, 04/07/2017-08/07/2017, IEEE Computer Society, p. 665-674, 2017.