Security Analysis of FIDO

Jyotsna Sharma
jyotsna.sharma@ischool.berkeley.edu

Nick Keith
nickbkeith@ischool.berkeley.edu

Jason Chow
jchow@ischool.berkeley.edu

Introduction

Passwords are a ubiquitous part of the digital age…yet insecure. According to the annual Verizon Data Breach Investigations Report, more than 80% of all breaches are the result of compromised credentials. Password protected systems have well known attack vectors that include phishing, brute force and even guessing. Even more concerning, are emerging attack vectors that can bypass multi-factor authentication, leaving bolstered systems vulnerable. Fortunately, significant investments are being made to solve the compromised credentials problem, with the goal of eliminating passwords altogether.

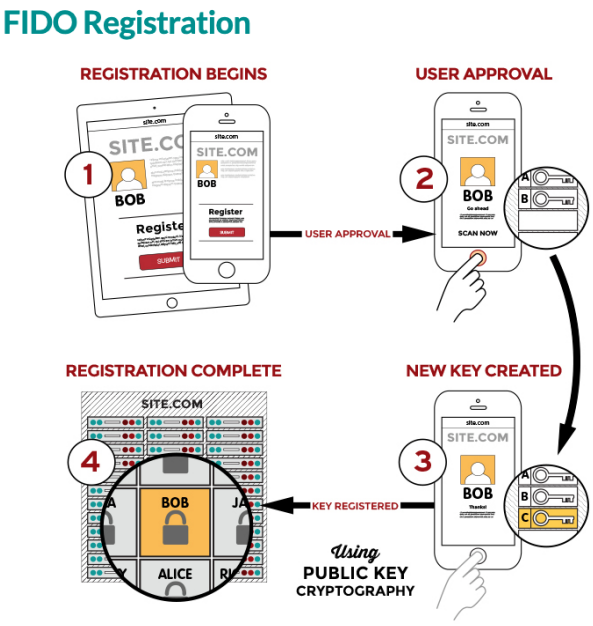
In 2012, several prominent companies, including PayPal, Lenovo and Nok Nok labs, founded the FIDO Alliance. Since its conception, the FIDO Alliance has developed a set of authentication standards known as FIDO ("Fast Identity Online"), which has dramatically accelerated the transition from passwords to passwordless alternatives. Currently, FIDO has become the dominant standard for passwordless authentication, with major players like Microsoft, Apple and Google expanding support of FIDO across their respective product lines. This seems great, but what are the security guarantees of FIDO?

Overview of FIDO

At its core, FIDO utilizes an asymmetric cryptosystem for authenticating users. The system consists of three parties: (1) a User, (2) an Authenticator (i.e., a smart phone, modern computer or security key), and (3) a Relying Party (i.e., an application). The system also consists of two distinct user flows: (1) registration and (2) authentication.

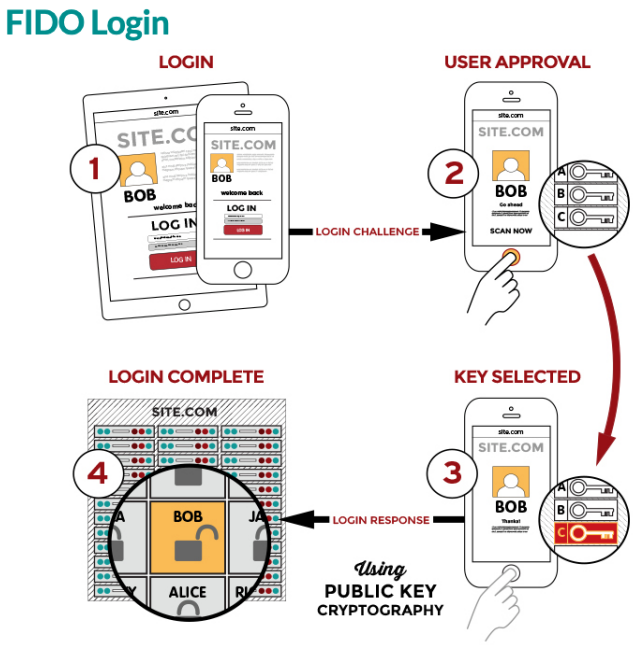
The registration flow consists of the following steps:

1) The user registers device with a relying party
2) The user provides a gesture (i.e., Touch ID) on their authenticator
3) The authenticator generates a unique public / private key pair on device
4) The authenticator digitally signs registration challenge from relying party, then returns signed challenge and public key to relying party



The authentication flow consists of the following steps:

1) The relying party sends authentication challenge to user's device
2) The user provides a gesture (i.e., Touch ID) on their authenticator
3) The authenticator digitally signs authentication challenge with user's private key
4) The relying party verifies digital signature with user's corresponding public key



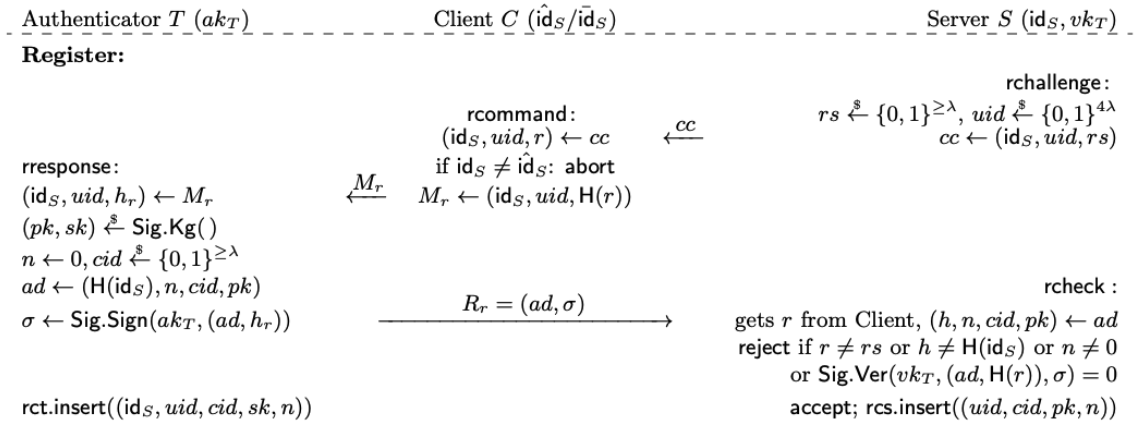
At the surface level, asymmetric cryptosystems are robust and widely adopted across the security domain. However, asymmetric cryptosystems are not immune to attack, with advancements in algorithms and computing technologies as the greatest threat. Given its rapid adoption, we will examine the FIDO cryptosystem in order to understand its current and post quantum security properties, as well as the potential attack vectors on the protocol.

FIDO Protocol

The FIDO Alliance has published three sets of specifications for its authentication protocol: (1) the FIDO Universal Second Factor (FIDO U2F), (2) the FIDO Universal Authentication Framework (FIDO UAF) and (3) the Client to Authenticator Protocols (CTAP). Currently, the FIDO Alliance, in partnership with the World Wide Web Consortium (W3C), are developing FIDO2 - a protocol that joins the evolving web authentication API (WebAuthN) with the evolving CTAP (CTAP2) protocol. As the specification continues to evolve, we will analyze the current cryptosystem primitives defined as of July 2022.

Registration

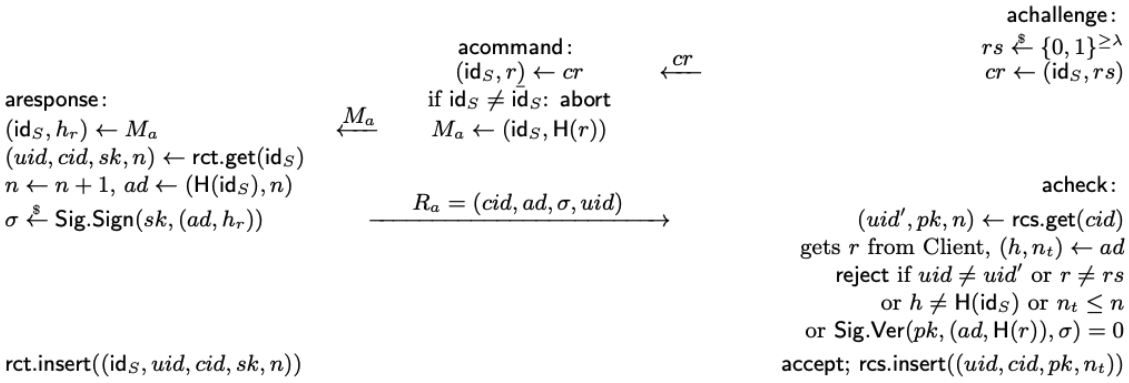
A FIDO transaction begins with a user registering an authenticator with a relying party. The user must authenticate with the relying party first, in order to establish an identity (user account) with the relying party. The relying party then sends a registration challenge **{cc}** to the user's authenticator which consists of the following key material: the relying party's server identity **{id_s}** (i.e. origin or hostname), a 512-bit user ID **{uid}**, and a 128-bit random string **{rs}** or higher. An intermediary client (i.e., web-browser) validates the relying party's server identity, then hashes **{id_s}** and **{uid} to {M_r}** with secure hashing algorithm **{H}** (SHA: 256-512 or SHA3: 256-512) before passing the hash to the authenticator. Once received, the authenticator generates a key-pair **{pk, sk}** using an RSA (2048-bit) or Elliptic Curve (256-bit) signature algorithm, then a hashed **{H}** attestation signature is generated that consists of the following: the relying party's server identity **{id_s}**, a signature counter **{n}**, a client credential **{cid}** for the user, and the user's public key **{pk}**. The attestation signature and public key are returned to the relying party, where the server checks the attestation signature and stores public key in a register in order to validate future signatures.



Authentication

Once an authenticator is registered with a relying party, the relying party can subsequently authenticate users by validating the digital signatures on authentication challenges. Authentication challenges **{cr}** consist of the following key material: the relying party's server identity **{id_s}** (i.e. origin or hostname), and a 128-bit random string **{rs}** or higher. The intermediary client validates the relying party's server identity, then hashes **{id_s}** and **{rs}** to **{M_a}** with a secure hashing algorithm **{H}** (SHA: 256-512 or SHA3: 256-512) before passing to the authenticator. The authenticator retrieves the user identity **{uid}**, client credential **{cid}**, private key **{sk}** and signature counter **{n}** using the relying party's server identity **{id_s}**. The authenticator then increments the signature counter **{n+1}**, then computes an authentication signature **{H}** using the relying party's server identity **{id_s}** and the incremented signature counter **{n+1},** then responds to the relying party with the following: relying party's server identity **{id_s}**, user identity **{uid}**, client credential **{cid}**, digital signature **{sk}** and signature counter value **{n+1}**. The relying party valides the response using the corresponding public key **{pk}** then updates its own signature counter **{n+1}** to complete the authentication process.

Authenticate:



Security Properties of FIDO

The current FIDO specification utilizes an asymmetric cryptosystem with industry recommended algorithms and protection mechanisms. Attestation key pairs are computed using an RSA (with 2048-bits or higher) or Elliptic Curve (with 256-bits or higher) digital signature algorithms. Key material is protected using SHA: 256-512 or SHA3: 256-512 secure hashing algorithms. A signature counter and relying party server identifier are included in FIDO transactions to help protect the system against replay and man-in-the-middle attack vectors. According to the "Universal Security" paper, which draws a relation between cryptosystem key lengths and the amount of energy required to boil water, the FIDO cryptosystem has a security level between 'global' and 'solar'. Said differently, the minimum amount of energy required to break the FIDO cryptosystem could boil all the water on earth, making it infeasible for attackers to break.

		bit-lengths		
security level	volume of water to bring to a boil	symmetric key	cryptographic hash	RSA modulus
teaspoon security	0.0025 liter	35	70	242
shower security	80 liter	50	100	453
pool security	2 500 000 liter	65	130	745
rain security	$0.082\,km^3$	80	160	1130
lake security	$89\,km^3$	90	180	1440
sea security	$3\,750\,000\,km^3$	105	210	1990
global security	$1\,400\,000\,000\,km^3$	114	228	2380
solar security	-	140	280	3730

Quantum Cryptography Primer

Quantum cryptography is the science of exploiting mechanical properties of quantum in order to perform cryptographic tasks. There are two categories in quantum cryptography: The first is the hardware-based approach known as Quantum Key Distribution (QKD), which uses fundamental quantum mechanics principles to facilitate secure communication without interception. The second is a software approach known as Post-Quantum Cryptography (PQC), which is based on new algorithms that, unlike RSA, are not based on factoring large prime numbers. It is believed that in the near future, large primes will be breakable by high performance quantum computers, leaving today's cryptosystems at risk.

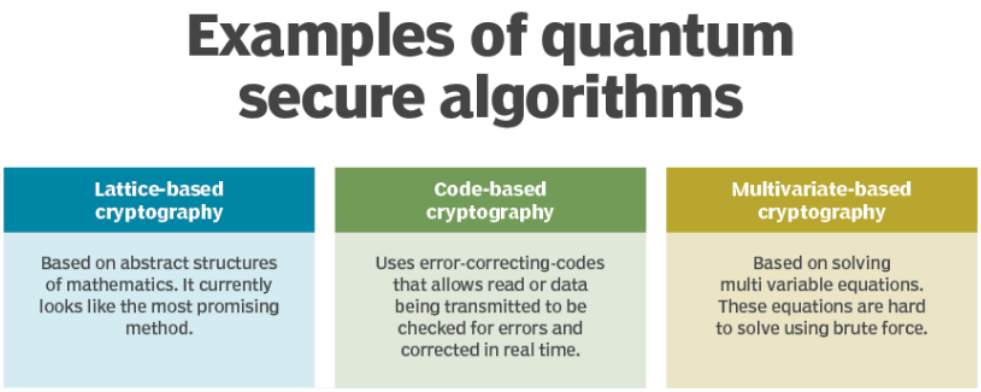
Post Quantum Cryptography

The FIDO cryptosystem relies on digital signatures to authenticate users. A compromised digital signature means an attacker can pose as a valid user, spoof the FIDO authentication process, and carry out an account-takeover attack. Today's digital signatures are implemented using public key cryptography (PKC), however, with the invention of quantum computers, RSA and Elliptic Curve digital signature algorithms will no longer be secure as quantum computers can factor large numbers in a couple hours. The National Institute of Standards and Technology (NIST) predicts quantum computers will be fully operational in a decade, and can break asymmetric key cryptography. Once factoring is possible, nearly all modern day cryptosystems, including FIDO, will no longer be secure.

To defend privacy now and down the road, out of all the post-quantum signing algorithms received, NIST has approved four quantum-resistant algorithms, namely: CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCOM and SPHINCS+. These four algorithms were designed in collaboration with experts across the globe, and rely on math problems that both classical and quantum computers should have difficulty solving. There are two main applications for which encryption is used:

1. **General encryption**: helps protect information exchanged over a public network. NIST has selected the CRYSTALS-Kyber algorithm when accessing secure websites. It has comparatively small encryption keys that two parties can exchange easily and has high speed of operation.
2. **Digital signatures**: are used for identity authentication. These are often used when we need to verify identities during a digital transaction or to sign a document remotely. NIST has selected CRYSTALS-Dilithium as the primary algorithm, FALCON for applications that require small signatures compared to what Dilithium can provide and SPHINCS+ for backup even though it is slower and larger than other two; since it is based on math approach different from others.

Types of Post Quantum Algorithms



1. **Lattice-based Cryptography:** Lattices are the most studied and flexible. The lattice algorithm can support digital signatures and key exchanges, and is capable of sophisticated encryption models too since the foundation is based on linear algebra, despite the need for complex mathematical processes for security proofs and optimization. Among the different lattice-based algorithms being considered for the NIST standards are Kyber and Dilithium. Kyber is a key-encapsulation mechanism that uses the algebraic number theory for better performance. Dilithium is a scheme based on the digital signature, and its performance has been very high. Falcon is based on the theoretical framework of Gentry, Peikert and Vaikuntanathan for lattice-based signature schemes.

2. **Hash-based cryptography:** creates signature algorithms whose security is mathematically based on the security of a selected cryptographic hash function. SPHINCS$^+$ is a stateless hash-based signature scheme which incorporates multiple improvements, specifically aimed at reducing signature size.

3. **Multivariate cryptography:** This includes cryptographic systems such as the Rainbow scheme which is based on the difficulty of solving systems of multivariate equations. Various attempts to build secure multivariate equation encryption schemes have failed. However, multivariate signature schemes like Rainbow could provide the basis for a quantum secure digital signature. There is a patent on the Rainbow Signature Scheme.

4. **Code-based cryptography:** relies on error-correcting code. The original McEliece signature using random Goppa codes is considered strong, however, there are many variants of the McEliece scheme, which seek to introduce more structure into the code (in order to reduce the size of the keys), but are proven to be insecure. The Post Quantum Cryptography Study Group sponsored by the European Commission has recommended the McEliece public key encryption system as a candidate for long term protection against attacks by quantum computers.

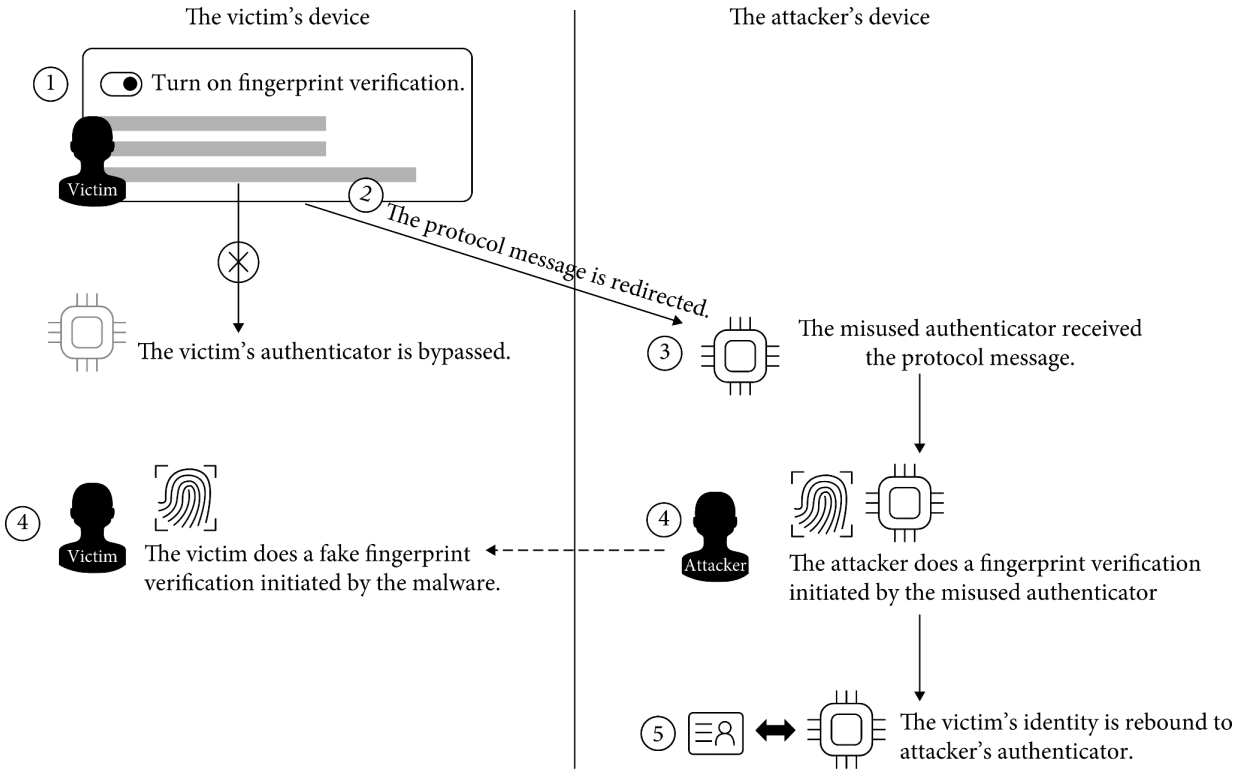
Vulnerabilities to FIDO

FIDO utilizes some of the more secure designs and methodologies currently available, but that does not make them impervious to attack. Some of the known attacks include USB timing attacks and real time phishing, which downgrades FIDO to weaker alternatives, amongst others.

One of the more intriguing FIDO attacks is called an “Authenticator Rebinding Attack” of the Universal Authentication Framework (UAF). A recent article focuses on attacking Android applications that utilize FIDO protocol. During their study, there were some threat frameworks that must be met, thus some assumptions were made: first, the attacker can install malware on the victim’s Android device. As we all know, this is relatively easy to do either using rogue APs, Pineapples, or sniffing onto an unprotected network. The second assumption is that the attacker can crack the Android OS and gain root privileges. This is also not far fetched considering the first assumption.

The attack begins when the victim enables the fingerprint or biometric function on their Android device, which is often required by some sort of installed application. From there, and considering the two aforementioned assumptions, the attacker’s malware redirects traffic from the installed application and points it to the attacker’s device instead of its intended destination. The attacker fools the victim’s authenticator to move forward with the UAF operations with the redirected, or false, message.

Once that is complete the attacker authenticator moves forward with fingerprint authentication. Concurrently, attacker’s malware running on the victim’s device uses the fake fingerprint authentication window to pretend to verify the victim’s fingerprint which makes the victim unaware of any abnormalities. The attacker completes the UAF protocol registration operation on behalf of the victim and rebinds the victim’s identity to the attacker’s misused authenticator. Thereafter, the attacker can bypass the fingerprint verification in the user’s device and perform a transfer or payment without the user’s authorization.[13]

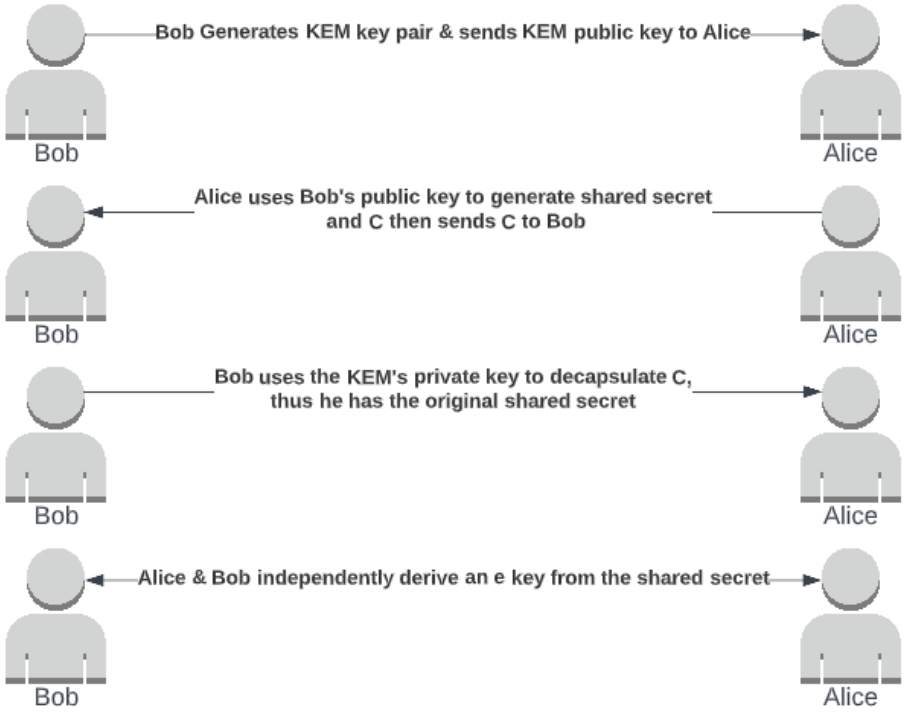


Potential Vulnerabilities Post Quantum Computing

With quantum computing becoming more than just a buzzword in academia, encryption post quantum (PQ) computing needs to be discussed. A system like FIDO has more strengths than weaknesses but we are unsure of how to secure FIDO in a PQ age. Below are a couple options or changes that could be implemented to help keep FIDO more secure.

The use of General Hybrid Signatures (GHS) is one way to help maintain integrity with FIDO. Instead of replacing the Elliptic Curve Digital Signature Algorithm, a second signature would be added. During credential registration, two public keys will need to be sent by the authenticator. By doing this the chances of breaking the encryption will be virtually nullified even PQ. [14]

Another interesting approach to maintaining encryption PQ for FIDO is to incorporate Key Encapsulation Mechanisms (KEMs). Faster and more compact, KEMs provide two easy advantages over GHS[15]. Simply put, one key exchange is made upon credential registration. Then both the authenticator and RP have a shared symmetric key and can authenticate themselves via message authentication codes.



Conclusion and Road Ahead

The world is rapidly transitioning to passwordless alternatives, using FIDO as the underlying security mechanism. Accordingly, we examined the FIDO cryptosystem in order to understand its current and future security properties, and potential attack vectors of the protocol. Industry hardened digital signature and secure hashing algorithms (with minimum required key lengths) give the current FIDO cryptosystem its strength. With advancements in quantum computers, theoretical analysis has revealed the intractable properties of factoring and elliptic curve addition (on classical computers) are easily solved with quantum computers. Fortunately, new quantum resistant algorithms are emerging, but creating new challenges of migrating to the new algorithms. While significantly more secure than password only systems, FIDO systems are still not immune to attack. As previously noted, there are a number of known attacks against FIDO. Attackers will continue to find new ways to break FIDO's encryption methodology, however, through vigilance most of the attacks will be thwarted.

References

1. Verizon Data Breach Investigations Report: https://www.verizon.com/business/resources/reports/dbir/
2. FIDO Alliance: https://fidoalliance.org/
3. WebAuthN Signature Scheme: https://www.w3.org/Submission/2015/SUBM-fido-signature-format-20151120/
4. Provable Security Analysis of FIDO2: https://eprint.iacr.org/2020/756.pdf
5. Universal Security: https://eprint.iacr.org/2013/635.pdf
6. An Introduction to Post-Quantum Public Key Cryptography: https://www.infoq.com/articles/post-quantum-cryptography-introduction/
7. NIST Announces First Four Quantum-Resistant Cryptographic Algorithms: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms
8. Quantum Cryptography vs Post Quantum Cryptography - Differences Explained: https://www.ssl2buy.com/wiki/quantum-cryptography-vs-post-quantum-cryptography
9. QKD Versus PQC - Which One To Use?: https://www.insidequantumtechnology.com/quantum-key-distribution-vs-post-quantum-cryptography/
10. Math Paths to Quantum-safe Security: Hash-based Cryptography: https://www.isara.com/blog-posts/hash-based-cryptography.html
11. Post Quantum Cryptography: https://en.wikipedia.org/wiki/Post-quantum_cryptography
12. SPHINCS+ : https://huelsing.net/wordpress/?p=558
13. Challenges of Post-Quantum Digital Signing in Real-world Applications: A Survey⋆: https://eprint.iacr.org/2019/1374.pdf
14. Authenticator Rebinding Attack of the UAF Protocol on Mobile Devices: https://doi.org/10.1155/2020/8819790
15. Bundesamt für Sicherheit in der Informationstechnik (BSI). Migration zu postquanten-kryptografie. https://www.bsi.bund.de/SharedDocs/Downloads/DE/
16. Mobile energy requirements of the upcoming NIST post-quantum cryptography standards.https://ieeexplore.ieee.org/document/9126751