# Hardware Security Tokens With a Focus on FIDO2

## Seminar: Advances in Cryptography and IT-Security

Robert Hartings

January 16, 2022
RWTH Aachen
Research Group IT-Security

**Organizer:** Ulrike Meyer
**Supervisor:** Vincent Drury

**Abstract.** The abstract should briefly summarize the contents of the paper in 150–250 words. ullamcorper eget nulla facilisi etiam dignissim diam quis enim lobortis scelerisque fermentum dui faucibus in ornare quam viverra orci sagittis eu volutpat odio facilisis mauris sit amet massa vitae tortor condimentum lacinia quis vel eros donec ac odio tempor orci dapibus ultrices in iaculis nunc sed augue lacus viverra vitae congue eu consequat ac felis donec et odio pellentesque diam volutpat commodo sed egestas egestas fringilla phasellus faucibus scelerisque eleifend donec pretium vulputate sapien nec sagittis aliquam malesuada bibendum arcu vitae elementum curabitur vitae nunc sed velit dignissim sodales ut eu sem integer vitae justo eget magna fermentum iaculis eu non diam phasellus vestibulum lorem sed risus ultricies tristique nulla aliquet enim tortor at auctor urna nunc id cursus metus aliquam eleifend mi in nulla posuere sollicitudin aliquam ultrices sagittis orci a scelerisque purus semper eget duis at tellus at urna condimentum mattis pellentesque id nibh tortor id aliquet lectus proin nibh nisl condimentum id venenatis a condimentum vitae sapien pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas sed tempus urna et pharetra pharetra massa

## 1 Introduction

Despite their problems with phishing or dictionary attacks, for example, credentials with username and password are still the most common variant of user authentication today. To create secure accounts, it is recommended to use strong passwords that contain upper and lower case letters, numbers, and special characters, and should be at least eight characters long. Also, they should not be reused. The user has to remember these complex passwords or use a password manager, which comes with its own disadvantages. If credentials are reused on multiple accounts, they are vulnerable to credential stuffing attacks, in which an

attacker uses stolen username/email and password combination from one service on different services hoping that the victim used the same or similar credentials, making it easier for him to guess the right combination of username/email and password. Username and password are always vulnerable to phishing because it cannot be ruled out that even the most experienced user will make a mistake and enter their credentials on a website owned by an attacker.

This problem is challenged by the FIDO Alliance and the World Wide Web Consortium by providing a possible solution: The Fast Identity Online 2 (FIDO2) standard. The main difference between the proposed standard and the status quo is the paradigm shift from "something a user knows" to "something a user poses". The FIDO2 standard includes a successor to the Universal 2nd Factor (U2F), which was also developed by the FIDO Alliance, and, in addition to the familiar second factor, also offers the possibility for Single Factor Authentication, therefore making passwords as we know them redundant. The cost, inconvenience, and most users' inexperience with (hardware) security tokens are the current reasons for their low uptake.[LHGU21][GLSN+20]

Similar to other authentication variants, FIDO2 has its own unsolved problems and drawbacks. In this paper, I will attempt to summarize these problems and showcase possible solutions.

## 2 Background

### 2.1 Fast Identity Online 2 (FIDO2)

The Fast Identity Online 2 (FIDO2) Project is a joint effort by the FIDO Alliance and the World Wide Web Consortium (W3C). It is an open authentication standard succeeding prior work by the FIDO Alliance on Universal 2nd Factor (U2F). [GLSN+20] It consists of two protocols. The WebAuthn protocol, maintained by the W3C, and the Client to Authentication Protocols (CTAP), which are maintained by the FIDO Alliance. Members of the FIDO Alliance include Amazon, Google, Meta, and Microsoft.

The WebAuthn usage is not limited to the FIDO2 context, because the protocol only specifies a JavaScript-based API used for communication between a WebAuthn relying party application (e.g. websites like google.com, ebay.com, or amazon.com) and a WebAuthn client like a browser (e.g. Chrome, Firefox). The defined API enables the creation of strong, scoped, public-private key pairs, which are used as credentials for user authentication. Through scoping the standard ensures that the key pair can only be accessed by origins belonging to the original executing relying party. The API is used in two cases. First in case of registration and second in case of authentication. In the case of the registration, a public-private key pair is generated on behalf of the relying party on the authenticator and is subject to user consent. When the user agrees, the private key is saved on the authenticator and the public key is sent to the relying party, along with additional information like metadata of the used authenticator. In the case of the authentication, the user is presented with a selection menu of

accepted credentials and the origin that is requesting these keys. In both cases, the user consent and scoping are enforced by conforming User Agents and the used authenticator.[HJJ+21] Nowadays, all major browsers and operating systems support WebAuthn.[Aut21]

The CTAPs standardize the communication between a client and the external also known as roaming authentication devices. The protocols expect the authenticator to obtain evidence of user interaction via a gesture, a pin, a pattern, or biometric data like a fingerprint. It is not standardized how the channel between the client and the roaming authenticator is established nor how the transport layer is encrypted. The protocol contains the legacy CTAP1 protocol formerly known as the U2F protocol and the current CTAP2 protocol. The CTAP1 authenticators are also called U2F authenticators and the CTAP2 authenticators are also known as WebAuthn or FIDO2 authenticators. The communication between the client and roaming authenticator is done using one of the following transport technologies: USB (Universal Serial Bus), NFC (Near Field Communication) or BLE (Bluetooth Low Energy). [BHJ+21][LHGU21][AWAC20]

To sum it up, the main idea of FIDO2 is to use public-private cryptography instead of known credentials like username and password. Furthermore, it creates a public-private key pair unique to a given application or website, which is used to sign challenges from the service and is only generated and stored on the authenticator itself. This is realized through mutual authentication using service identification. In the case of login on a website, the authenticator receives the domain of the requesting website, effectively rendering phishing useless, because a relaying attacker cannot provide the authenticator with right domain.[UAA+21] While also preventing replay attacks and password theft. Public keys acquired through server breaches cannot be reverted to the original private key nor the secret stored on the authenticator. Furthermore, they cannot be used to determine private keys used for other services.

To ensure quality and security the FIDO Alliance have set up a meta service that can be inquired to verify the authenticator used for the login. The relying party can check if the authenticator meets the FIDO Alliance standards and has known vulnerabilities.[AWAC20]

### 2.2 Hardware Security Tokens (HSTs)

Hardware security tokens come in different shapes and forms. In the FIDO2 context, they are called authenticators, but to the public, they are also sometimes known as security keys. In FIDO2, a distinction is made between roaming and platform authenticators. In the category of roaming authenticators, to just name a few popular security keys, YubiKeys by Yubico, FIDO Keys by Feitian, and Titan Keys by Google are counted. As required by CTAP and WebAuthn standard the evidence of the user interaction is done via a touch sensor or biometric scanner, most commonly a fingerprint sensor. Besides the USB Security Fobes, phones and laptops are also roaming authenticators. The two largest hardware security tokens on phones are the Andriod Keystore and Apple TouchID. The

communication with the client takes place via USB, NFC, or BLE. In the category of platform authenticators, Apple TouchID, Andriod Keystore, Windows Hello are counted, but also TPM and Trusted Execution Environment. The main difference between roaming and platform authenticators is that platform authenticators run on the same device as the WebAuthn client and therefore do not use cross-platform communication and CTAP. [GLSN+20]

To be compliant with the standard the user has to touch a sensor to verify his presence and unlock the secret. The security key fobs are commonly shipped with simple presence sensors, but can also be shipped with biometric sensors, most commonly fingerprint sensors. Nowadays, almost all phones are shipped either with fingerprint sensors or face recognition sensors, or both, and they provide capabilities for pins and patterns.

The hardware security tokens are used to securely store a secret used for cryptographic functions in tamper-resistant storage. The secret never leaves the secure storage and is used for deriving subsequent authentication keys for the creation of public-private key pairs. The derived keys are mainly used to sign challenges received from the relying party application, but can also be used to identify a user.[PMD+21]

## 3 Problems of FIDO

While FIDO2 seems like a good solution, there are currently some disadvantages and problems, which cannot be ignored during the evaluation of the usability and usefulness of hardware security tokens.

### 3.1 Downgrade Attacks

Besides FIDO different methods of multifactor authentication exist, like One-Time-Passwords (OTP), confirmation SMS and calls, and the usage of recovery codes. Commonly the user can choose between the configured MFA schemes, but except FIDO2 none of the mentioned schemes is secure against real-time phishing. While reviewing Alexa's top 100 websites Ulqinaku et al. found out that most of these websites force users to register at least one different MFA to use FIDO2 in the first place. Effectively creating a vulnerability even when FIDO2 is used because it undermines the security of FIDO2. Only Google with Google's Advanced Protection offers a program not relying on weak MFA, but it is opt-in and not actively advertised on the Google Account Dashboard.

The downgrade attack on FIDO2 was shown by Ulqinaku et al. and requires that the victim has different and weaker MFA registered to his account. The goal of the attacker is to display the user a fake login screen and let the victim choose a weaker MFA scheme, which is vulnerable to real-time phishing. To achieve this goal the attacker has to set up a login page that is comparable to the website from which the attacker wants the login data or sessions. Because the user expects the security key pop-up, the attacker has to display the pop-up on the page as well. The attacker has to know if and when the user has used his hardware

security token to continue with his attack, otherwise, a user could get suspicious. Normally the browser would present the pop-up above the webpage containing the domain and ask for the security key, but the WebAuthn standard also contains an API function to detect the presence of an HST without displaying this pop-up. The difference is not easy to spot for a normal user.[UAA+21]

The authors of the paper suggested a solution for this problem to disable weaker alternative schemes, but also mentioned the problem of non-scalable recovery. This recovery adds significant cost to the service provider and would not scale to millions of users, while also providing inconvenience for the user. Registering multiple keys can be another solution but may also be costly and create a barrier. To determine if a recovery is genuine risk-based authentication could be used, but recent studies showed that they can be circumvented. Another solution suggested by the authors was to always show the hint when an application tries to access the HST, making it slightly easier for a possible victim to spot the attack.

### 3.2 Threats to HST

While FIDO2 seems secure it relies on a secure HST. Therefor, if the HST is compromised FIDO2 is not secure anymore. An attacker can get a hold of an access token on the supply chain between a manufacturer and the end users, either by intercepting the delivery or inserting malicious HST as an manufacturer or a re-seller. An attacker can also buy genuine token and return malicious HST to the seller on refund, because most sellers won't check if the HST got tempered with. Malicious HST result in one or more of the following attack vectors: firmware modifications, hardware modifications or secret extraction. Through the firmware modification an attacker can pre-initialize an token or add malicious code which exploits e.g. USB interfaces. In hardware modification an attacker can wire the HST up with an wireless transceiver, like GSM or Bluetooth. But also he is able to build token replicas as instructions are public available and can be used without expert knowledge. The main goal of both modification is to extract secrets, e.g. keys or seeds, from the HST. This is done most commonely via falut injections, timing side-channels and bus snooping. [PMD+21] This results in the following attack scenarios.

*Run-time secret extraction* The run-time secret extraction can be subdivided into in-band and out-of-band attacks. In the in-band case, the HST is modified in a way that it leaks secrets through in-protocol (covert) channels like the signature or other channels used in the transcation. In the out-of-band case, the HST sends the secrets via a different covert channel outside of the protocol like Bluetooth or GSM.

*Delivery-time secret extraction* An attacker can extract pre-configured keys or seeds through the above mentioned mehtods, which allows him to determine the used keys.

This attack is only releveant to HST which are shipped with pre-configured secrets by the manufcature like YubiKeys. In case of most HST but also YubiKeys these secrets can be changed by the user whenever they want.

*Secret fixation* Using hardware or software modifications or both an attacker can pre-load a key to the HST, which makes the key computation deterministic.

*Predictable RNG modifcation* The Random Number Generator used for the secure creation of keys can be manipluliate to only create predicatable by using hardware and software modifications. In case of unintentionally weak randomness the attacker does not need to modify the token and can abuse it.

*Ransom attack* Like other ransom attacks, this attacks targets a denail of service. The HST is manipluliate in such a way that it stops operating after some time, demanding a ransom to resume working or release the secrets. This is attack has limited valabilty for FIDO2, since in most cases recovery codes are generated. This attack is more feasible for hardware walltes (out side of the scope of this paper).

*USB pivoting* On a nother node the HST can not only be used to attack logins, but also to attack the whole client via the USB interface. If the HST is equipped with malware it can act like USB Rubber Ducky (emulate a keyboard) or trigger a buffer overflow.

Pfeffer at al. present (already existing) methods to detect tempered HST. Modifcation on hardware or firmware level can be detected with tamper-evident package using holographic stickers. But this is only a low level of protection since holographic sticker are easily replaceable and the attacker can be a manufacturer or re-seller. A HST token can be single-piece cast, like the Yubikeys, or can be opened. Single-piece cast can be easily inspected but can be breakable with household chemicales when not using more chemical resistant plastic. Openable HST can be inspected by users increasing security by visually comparing manufacturer pictures with the HST. This process has its downsides as it is error-prone and cumbersome. Siginals on the printed circuit board (PCB) are intercebteable or maniplulatable needing shielding with can be done with a secure CPU or a secure element (external co-processor). The key never leaves this secure element and therefore cannot be intercept. To prevent firmware manipluliation automatic and manual software verification can be used. In case of automatic verification it will distinguished between local and remote validation. The local validation only validates the integrity by conducting a signature check. The remote validation is a more sophisticated where the internal status is validated by a third party. Both methods needs to visible to the user otherwise the user cannot make related trust decisions. The manual verification can only be done with some HST and the corresponding software has to be searched as it is not easily findable. Moreover, this method is neither explained nor advertised by vendors or service provides, letting uneducated users in the dark. A way to

prevent attacks on pre-configured secrets is to not ship them at all and let the user generate their secrets themselve. But manual verification and dispense of pre-configured secrets reduce the userfriendliness of HST, which can result in lower markt shares. [PMD+21]

### 3.3 Misconceptions

The study by Lassak et al.[LHGU21] about misconceptions in FIDO2 Biometric WebAuthn shows that users are not yet educated enough to understand the basic functionality of FIDO2 HSTs. There are among others misconceptions about storage location, recovery and usage of different devices. The study was held online with 42 participants from the UK and US. All of them were older than 18. The used HSTs were the participant andriod smart phone.

*Storage Location* The majority of the participants thought that there biometrics were sent (in an encrypted fashion) to the corresponding service provider. Only 14 participants recognized that the biometrics are stored locally and only 2 figured out that the service provider could not get there biometric data, because he is not in possession of the phone (HST).

Likewise, only 24 participants know / guessed that their biometric data are not affected in the event of a database breach on the services provider site.

Overall, only 4 participants were confident that their biometric data did not leave the phone when they used it for authentication.

*Lost HST* Because the private key used for authentication is stored on a phone, losing it can provide an attacker with access to the accounts. From the 42 participants 39 thought that an attacker need their biometrics for the authentication while in fact the private key can be unlocked with fallback mecanism likes PIN, pattern or password.

*Availability* If the unlocking of the phone via biometric fails only five participants were aware that they can unlock the HST with fallback options like PIN, pattern or password. The other participants stated that they have not setup a backup method at the service provider or they have to contact the service provider to recover their account.

*Multiple Devices / Delegating Access* Also, there are missconecptions concerning device sharing. One missconecption is that is possible to use another device (after registring ones biometric data). Only six participants ware aware that the login is tied to the authenticating device and that the biometric is only used to decrypt and unlock the private key for the authentication process. Transfering the private key from one authenticator to another is not intended in the current WebAuthn specifaction. If this kind of behavior is needed a roaming authenticator is needed. Furthermore, it is possible on some services to register more than one HST for a single account.

To grant a trusted person access 39 participants ansewered that it is not possible since the person would not have the required biometic data. The other participants argued that it would be possible using fallback methods or registering the biometric data of the trusted person on the HST (phone).

*Idea of resolving this problem* To prevent this type of misconceptions, service providers and HST manufacturers should launch an information campaign about HSTs and how they work. If one or both of them fail to do their job or additional advertising is needed, consideration should be given to whether government institutions, such as the BSI (German Federal Office for Information Security), should launch joint / own campaigns to inform the population about better methods of securing their accounts online.

### 3.4 More Problems

Besides the mention problems the FIDO2 standard has some other problems which are not bound to the implementation. This is highlighted in a study by Lyastani et al. [GLSN+20]

*Account Recovery* A fear the participants of the study had, was the losing of the HST, which means that they wont be able to access their accounts. Currently other 2FA can be used to allow account access even when a HST fails. Some service provides allow to setup additional HST, which is recommended by the FIDO Alliance, or other 2FA schemes, which are as we showed earlier vulnerable to e.g. phishing. Only Google's Advanced Protection Program forces the user to register two HSTs.

This issues is not yet challenged by the FIDO Alliance. The current recommendation is to use additional authenticators. The authors recommend guiding the users in the task of scalable account recovery. But this won't change the fact, that account recovery is a serious issues which should be handled by the FIDO instead of single service provides to create a single way of account recovery. This problem needs attention quickly especailly when FIDO2 should become the only 2FA or even a 1FA.

*Account Suspension* The authors themselve raise the concern how to revoke the HST from a account if the HST of the user is stolen. The FIDO Alliance agures that the risk of such thefts are lower than being the victim of a phishing campaign or server breach. The authors are concerend about the effect of FIDO (especially OFA) abusers in (intimate) partner violence. They are unsure if HST ease or hamper such target attacks. Also, the authors state that such behavior is not enough considered yet and that users need the possibility to lock their account down without having the HST. An insperation can be the key revocation in PKI or GPG.

## 4 Adoption Barries

Farke et al. explored the willingness of users to use HST in a study. Since they only had 9 participants a general statement cannot be made. But the study can show which prevent a widespreading of HST / FIDO2. The users indicated that they feared losing the HST, the additional effort to plug in the HST and the habit of using passwords.

*Convenience* Three participants stated that the login with a HST requires additional efforet and time, making it less convenient than using credentials like username and password. Also was perceived negatively that the setup of the HST took additional time (5 - 10 minutes). This implies that even a short duration of time can reduce the willingness to use HST.

In the study the workflow was negatively impacted, because using a HST meant that additional steps were required for each login attempt. The click count and the time to login were listed. Also it was mentioned that Windows Hello can be a hurdle for adoption, because the account selection when using multiple accounts for a single service was not clear. This may be implementation dependet but if a user has to search the right account in the HST / authentication device it can devinetly prevent spreading of FIDO2.

The study also indicated that thought must be given to overcoming old habits and that measures must be taken to disseminate HST.

The authors mention that the participants understand the authentication process using HST and that the authentication time was more of a limiting factor. The difference in speed could be one of the primary adoption barries for FIDO2 / HST. Also, participants ansewered that unconscious decisions to use passwords instead of the HST which makes it even more difficult to change the mental mindeset of the users in general to use HST over passwords. Since only recent version of OS and clients were equipped with all WebAuthn features, WebAuthn should spread better in the future. Because it is no longer required only special software or certain knowledge and is thus better available to the general public. To overcome the mention adoption barriers the authors have the following suggestions:

- Support for multiple HST (platform and roaming)
- Requirement of adoption of HST
- Implementation of FIDO2 in as many as possible services
- Allow to "remember" the unlock state of HST when using PIN or biometrics.

[FLS+20]

## 5 Conclusion

educate the user / population, support FIDO2 on more websites

## References


[Aut21] Auth0. Does my browser support webauthn. https://webauthn.me/browser-support, 2021. Accessed: 2021-12-22.

[AWAC20] Fatima Alqubaisi, Ahmad Samer Wazan, Liza Ahmad, and David W Chadwick. Should we rush to implement password-less single factor fido2 based authentication? In *2020 12th Annual Undergraduate Research Conference on Applied Computing (URC)*, pages 1–6, April 2020.

[BHJ+21] John Bradley, Jeff Hodges, Michael B. Jones, Akshay Kumar, Rolf Lindemann, and Johan Verrept. Client to authenticator protocol (ctap). https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-20210615.html, 2021. Accessed: 2022-01-16.

[FLS+20] Florian M. Farke, Lennart Lorenz, Theodor Schnitzler, Philipp Markert, and Markus Dürmuth. "You still use the password after all" – exploring FIDO2 security keys in a small company. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 19–35. USENIX Association, August 2020.

[GLSN+20] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. Is fido2 the kingslayer of user authentication? a comparative usability study of fido2 passwordless authentication. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 268–285, May 2020.

[HJJ+21] Jeff Hodges, J.C. Jones, Michael B. Jones, Akshay Kumar, and Emil Lundberg. Web authentication: An api for accessing public key credentials level 2. https://www.w3.org/TR/2021/REC-webauthn-2-20210408/, 2021. Accessed: 2022-01-16.

[LHGU21] Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. "it's stored, hopefully, on an encrypted server'': Mitigating users' misconceptions about FIDO2 biometric WebAuthn. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 91–108. USENIX Association, August 2021.

[PMD+21] Katharina Pfeffer, Alexandra Mai, Adrian Dabrowski, Matthias Gusenbauer, Philipp Schindler, Edgar Weippl, Michael Franz, and Katharina Krombholz. On the usability of authenticity checks for hardware security tokens. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 37–54. USENIX Association, August 2021.

[UAA+21] Enis Ulqinaku, Hala Assal, AbdelRahman Abdou, Sonia Chiasson, and Srdjan Capkun. Is real-time phishing eliminated with FIDO? social engineering downgrade attacks against FIDO protocols. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3811–3828. USENIX Association, August 2021.