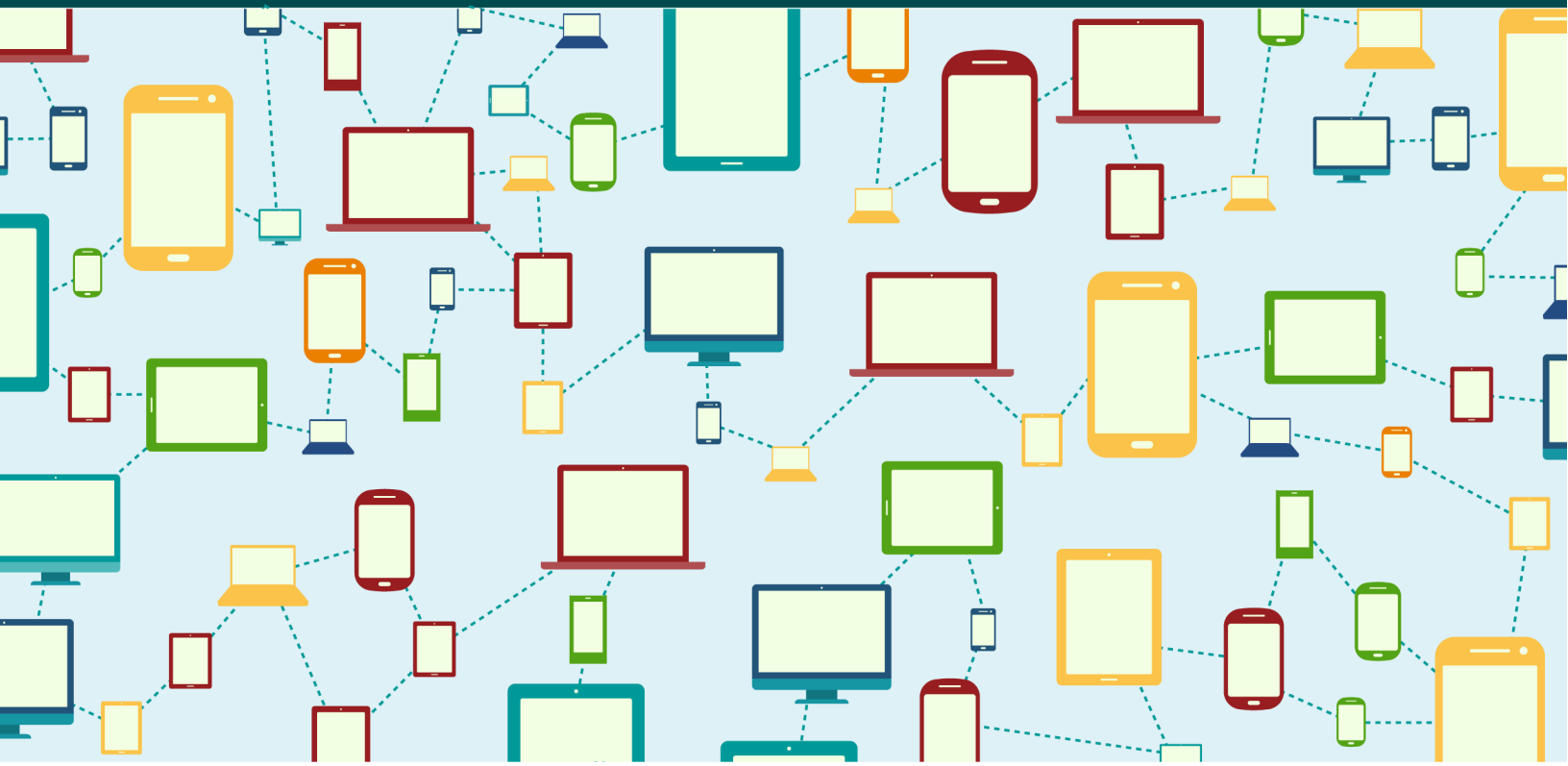
fido™ ALLIANCE | simpler stronger authentication



FIDO Enterprise Adoption Best Practices

FIDO and PKI Integration in the Enterprise

30 April 2019

Editors

Salah Machani, RSA
Arshad Noor, Strongkey

Audience

This white paper is aimed at enterprises looking to expand their authentication systems to FIDO technology and to work in conjunction with other authentication systems such as Public Key Infrastructure (PKI), Kerberos, and Lightweight Directory Access Protocol (LDAP). This document discusses specifically the use of FIDO with a PKI.

It is assumed the reader has a good understanding of PKI and FIDO architecture and protocols. To learn about FIDO, visit https://fidoalliance.org/category/intro-fido/.

Table of Content

Contents

Audience 2
Table of Content 3
Contents 3
Scope 5
Introduction 5
Use of PKI in the Enterprise 7
Device Logon 10
Pre-boot Authentication 10
Web Client Authentication 10
Thick Client Authentication 10
Email Encryption and Signing 10
VPN-IPSec 10
Transport Layer Security (TLS) 11
EAP (Extensible Authentication Protocol)-TLS for Wireless Access 11
Transaction Authorization 11
Document Signing 11
Code-Signing 12
Disk Encryption 12
Single Sign-On 12
Trust Establishment 12
Combined PKI and FIDO Authentication Solution for the Enterprise 12
Enrollment of FIDO credentials in a mixed PKI and FIDO environment 12
Existing Users with a PKI Credential 12
New Users without a PKI Credential 14
Enrolling Secondary FIDO Authenticators 14
Revocation/Expiration of credentials in a mixed PKI and FIDO environment 14
Account Recovery 15
Use PKI Credential for Account Recovery 15
Operations in a mixed PKI and FIDO Environment 16
Benefits of the Combined PKI and FIDO Approach 16

Acknowledgements 17

Appendix A: PKI2FIDO Open Source Project 18

Scope

For organizations that invested in a variety of authentication systems – such as public key infrastructure (PKI), Kerberos, Lightweight Directory Access Protocol (LDAP), etc. - and who want to deploy solutions based on FIDO protocols, this whitepaper answers the following questions:

- How can FIDO protocols deliver new and/or enhanced business benefits to the enterprise?
- Which enterprise applications (and application layer protocols) can use PKI?
- Can FIDO be used to provide similar services as PKI for applications that use or can use public key cryptography?
- Which enterprise security needs and security threats are best addressed using FIDO?
- How can an expanded public-key cryptographic system incorporating PKI and FIDO benefit an enterprise?
- What are the business implications for adding FIDO technology within an enterprise that already operates other authentication systems?

This document covers enterprise and government use cases. Consumer use cases are not in the scope of the whitepaper.

Introduction

Public-key infrastructure (PKI) is a mature technology, natively embedded in multiple platforms, and delivers many security benefits to businesses:

- Encryption of network sessions between requesting client applications and servers using the Transport Layer Security (TLS) protocol;
- Use of digital certificates to identify clients and servers in various PKI based protocols;
- Encryption of electronic mail and files to protect the exchange of sensitive data using the Secure Multipurpose Internet Mail Extensions (S/MIME) protocol;
- Authentication of humans, devices and applications without the need to use *shared secrets* between requesting clients and servers where resources are accessed.
- Transaction confirmation of business and legal transactions using digital signatures; and
- Preservation of the integrity of data and transactions using digital signatures.

These benefits have enabled many enterprises, government organizations and entire countries to deploy PKIs to issue digital certificates. Indeed, without PKI, the secure transport of data between applications and their servers would be extremely complex, since every application would have had to devise its own mechanisms for transporting data securely between computers.

As useful and persuasive as the business benefits of PKI are, PKI has evolved and has become increasingly complex to architect, build and operate. Application developers responsible for building applications that use public-key cryptography as defined by digital certificates, are faced with significant challenges to address complex issues in integrating technology components into the application; they also are tasked with keeping the environment stable as different components are updated periodically by their respective manufacturers - which tends to break the integrated solution. Among the challenges are the need to manage certificate revocation status, certificate renewal and root CA certificate rollover. While this is truer in some proprietary and in-house built applications, these problems exist even when commercial PKI-enabled applications must work with third-party devices and their

respective device drivers. Enterprises are frequently forced to make trade-offs between business functionality and security benefits as time goes by; this challenge is particularly true with PKI due to its complexity.

FIDO technology came into existence due to a confluence of events:

- The extraordinary success of the worldwide web and consequent dominance of the web-application architecture;
- The growth of smartphones integrated with biometric capabilities for local authentication to the device: fingerprint, iris, voice and facial recognition;
- The failure of PKI to address authentication problems in the consumer market. Even within enterprise and government markets, PKI resulted in creating more challenges than their adopters anticipated;
- The rise of online commerce and media platforms whose business models depended on acquiring and managing users as cheaply, quickly and simply as possible, thus eliminating every other authentication scheme but usernames and passwords;
- The explosion of password-based web-applications on the internet, leading to unprecedented password-proliferation and consequential password-management headaches for users; and
- The steep increase in attacks and compromises of sites due to many factors: password-reuse, unpatched systems, zero-day vulnerabilities, implementation flaws, increased sophistication of attackers, etc.

Why would an enterprise that has deployed a PKI choose to use FIDO? What benefits can be gained with FIDO, if the enterprise has invested significant amounts of money on enabling TLS ClientAuth to its web-applications, has enabled digital signatures in its applications, has enabled certificate-based logon to desktops and virtual private networks (VPN)? While every enterprise's decision to invest in FIDO must be made on a case-by-case basis, in general, there are business and technical advantages to using FIDO even within organizations that have a PKI.

When using FIDO, the benefits accruing to the enterprise include:

- Faster integration within applications using the World Wide Web Consortium (W3C) web-browser JavaScript API, Web Authentication or WebAuthn for short, for the use of strong web authentication based on the FIDO2 protocol's public key cryptography. The WebAuthn specification[1] has broad support and is currently a W3C Proposed Recommendation[2] and implemented in multiple platforms.
- Lower costs to the enterprise because there are hundreds of FIDO component suppliers within the FIDO ecosystem, while the number of PKI suppliers has been steadily decreasing, leaving enterprises with fewer choices for PKI related components;
- Compliance with privacy laws because FIDO was designed with privacy in mind. Devices that store digital certificates can enumerate certificates, thus leaking identities, even if they do not permit access to private keys of unauthenticated users. FIDO protocols ensure that devices storing FIDO credentials will not leak such information to anyone, apart from providing assertions to the authorized Relying Party (RP) when used by an authorized user;
- Reduced complexity for end-users, developers and administrators because FIDO does not use digital certificates for users and is designed to minimize the number of choices presented to users. FIDO application developers and administrators, similarly, do not have to deal with certificate-hierarchies, cross-certification, certificate bridges, choices of revocation protocols, etc.;

[1] https://www.w3.org/TR/webauthn/

[2] https://www.w3.org/blog/news/archives/7538

- Simplified key management for RPs and end-users because FIDO credentials do not expire. RPs may still choose to manage FIDO credentials' lifecycle at their discretion.
- Increased availability to end-users because FIDO not only allows multiple keys from different authenticators to be registered to the same account, but also because FIDO is enabled on newer platforms such as Android and Windows 10 devices. These provide at least two FIDO authenticators to end-users, ensuring that the loss of a single authentication device does not prevent them from using an alternate device to authenticate to resources.

The benefits of FIDO are beginning to be understood by end-user companies aka Relying Parties (RPs). As the FIDO Alliance evolves from defining protocols and encouraging implementations of FIDO components, to promoting the adoption of FIDO within RPs, many issues unrelated to FIDO protocols or implementations require guidance. This guidance can assist RPs to deploy FIDO-based solutions with knowledge of deployment challenges and how to navigate them.

One such challenge is when an enterprise currently has an operational PKI and chooses to adopt FIDO to address business requirements. PKI and FIDO both use public-key cryptography to deliver strong-authentication and transaction digital signatures. Yet, the protocols, tools and workflows for using each technology are markedly different. More importantly, the impact on end-users will be sufficiently different from their use of digital certificates – especially if end-users are required to use smartcard or similar devices to use their digital certificates – that RPs will surely benefit from guidance on how to navigate these challenges.

This paper aims to address these challenges. What this paper is not, is a discussion about whether FIDO should replace PKI. Determinations about whether to replace existing forms of authentication within an enterprise, how long multiple authentication systems should coexist within enterprises, etc. must be made by each enterprise individually. These determinations will be based on factors that are out-of-scope for discussion within this document.

Use of PKI in the Enterprise

The use of digital certificates is supported by many applications. Some common use-cases within an enterprise are:

- Server authentication
- Client authentication
- Desktop logon
- Pre-boot authentication to devices
- Remote desktop access
- Email encryption and authentication
- Internet Protocol Security (IPSec) virtual private networks (VPN)
- Wireless access
- Document signing
- Transaction authorization
- Code-signing
- Disk encryption
- Single-Sign On (SSO)

- Federation and trust establishment

Given that FIDO protocols use public-private key-pairs to enable their functionality, it is feasible that FIDO can be used to address many use-cases where X.509 digital certificates are traditionally used.

In which case, this begs the question: Why use FIDO at all?

As useful as PKIs are, FIDO protocols have a few advantages over PKI in some use cases; those use-cases will be highlighted below. Broadly, FIDO is simpler for end-users to use, and for application developers to integrate into web and mobile applications. FIDO infrastructures can also be less expensive to operate and manage given that they do not require many artifacts generally associated with PKIs:

- The need to issue digital certificates to every end-entity that must participate in the strong-authentication infrastructure;
- The need to issue Certificate Revocation Lists (CRL) or use Online Certificate Status Protocol (OCSP) Responders to verify the status of a digital certificate;
- The need to create Certification Authorities (CA) to issue digital certificates;
- The need to renew certificates when they expire;
- The need to use Hardware Security Modules (HSM) for the generation and storage of CA or OCSP Responder signing keys;
- The need to establish a detailed Certificate Policy (CP) and/or a Certification Practices Statement (CPS) for the use and operation of a PKI;
- The need to resolve technical integration issues with every platform and how digital certificates are used on that platform;
- The need to chain an internal PKI with an external Trusted Third Party (TTP) CA to have digital certificates from the internal PKI be trusted on the Internet[3]; and
- The need to create complex technical relationships – such as Bridge Certification Authorities – between different PKIs, because distinct companies/agencies with their own PKIs need to collaborate and use strong-authentication when interacting with each others' applications.

Table 1 lists the type of applications and enterprise use cases that can be addressed using PKI and those that can be supported using FIDO.

Use Case	PKI	FIDO
Device Logon	Yes	Yes
Pre-boot Authentication	Yes	Yes

[3] Unlike PKI credentials/certificates, FIDO credentials are bound to a specific relying party and cannot be used across multiple security domains. With a private CA in a closed environment, a certificate does not need to be trusted on the internet.

Web Client Authentication	Yes[4]	Yes[5]
Thick Client Authentication[6]	Yes[7]	Yes
Email Encryption and Signing – S/MIME	Yes	No
VPN-IPSec	Yes	No
TLS	Yes	No
EAP-TLS for wireless access	Yes	No
Transaction Authorization	Yes	Yes
Document signing	Yes	Yes[8]
Code signing	Yes	Yes[9]
Disk Encryption	Yes	No
Single Sign-On	Yes	Yes
Trust Establishment (E.g. for federation)	Yes	No

Table 1: PKI and FIDO Uses Cases in the Enterprise

In all use cases where X.509 certificates are required by the underlying protocols, such as SSL/TLS for server-side authentication, S/MIME for email encryption and signing, IPsec for VPN, etc. FIDO cannot be used. FIDO can be used in the following cases:

[4] This is achieved using TLS client-side certificate-based authentication

[5] Typically, FIDO protocol runs over TLS (with server-only authentication mode) and provides additional security benefits such as web origin verification

[6] A "thick client application" is a rich client application that leverages native APIs from the platform on which the thick client application executes, to authenticate users to a remote server. It may be a mobile or desktop application.

[7] In PKI, it is possible to enable certificate-based authentication without user interactions. In FIDO, some form of user gesture is always required to access and use the FIDO private-key

[8] Technically, Transaction Authorization can digitally sign the hash of a document. However, the verification of that signature is easiest for the relying party where the signer's FIDO public-key is registered. Verifying that signature outside the domain of the RP becomes a complex issue which FIDO protocols are not designed to solve, and which are better addressed with PKI.

[9] Code signing can be achieved with FIDO as long as the Verifier (typically the RP) has access to the matching FIDO public-key.

Device Logon

While UAF and U2F do not natively support authenticating to a device operating system such as Windows, macOS or Linux, several third-party vendors have created proprietary FIDO based authentication for device logon.

As part of the continuing evolution of FIDO protocols, FIDO Alliance announced the standardization of the FIDO2 protocol in early 2019, which includes a specification for the Client to Authenticator Protocol (CTAP2). This protocol enables communication between an operating system platform and a "roaming" (external to the platform device) Authenticator over multiple transport protocols: Universal Serial Bus (USB), Near Field Communications (NFC) or Bluetooth Low Energy (BLE) to enable device logon. CTAP2 is implemented in multiple commercial products today.

Pre-boot Authentication

User authentication before the computer boots up could be implemented using PKI based authentication or FIDO based authentication in similar fashions. The user will have to present and/or unlock a smartcard or a FIDO Authenticator the moment the computer is turned on. Although such process is not explicitly called out or defined in FIDO specifications, pre-boot authentication solution vendors can use the FIDO2 protocol to enable this capability.

Web Client Authentication

This use-case is supported by all FIDO protocols – Universal Authentication Framework (UAF), Universal 2nd Factor (U2F) and FIDO2. Most major browser vendors have committed to implement the W3C WebAuthn application programming interface (API). At the time of writing, WebAuthn API is either supported or under development in Microsoft Edge, Google Chrome, Mozilla Firefox, Apple Safari and the Opera browsers. Readers of this paper should confirm support of specific browser releases from their respective manufacturers before making any deployment decisions.

Thick Client Authentication

A thick client such as mobile app or desktop app such as desktop software for virtual private networks can use FIDO protocols for authentication to a remote server. The thick client might require a FIDO module if it is not built into the platform itself[10], leverage a platform embedded FIDO Authenticator if APIs are available or interact with a roaming FIDO Authenticator using CTAP2. Note that FIDO protocols could also be used for strong-authentication to text-oriented applications such as Secure Shell (SSH).

Email Encryption and Signing

This use-case cannot be supported by FIDO protocols. Historically, e-Mail Security has been used in the context of the S/MIME protocol, which mandates the use of an X.509 digital certificate. Since FIDO protocols do not use digital certificates to identify cryptographic keys, this use-case cannot be supported by FIDO protocols.

VPN-IPSec

This use-case cannot be supported by FIDO protocols. Historically, virtual private networking without the use of the TLS protocol has been used only with the IPSec protocol, which mandates the use of an X.509 digital

[10]Check with your product vendor to understand their support for FIDO protocols.

certificate. Since FIDO protocols do not use digital certificates to identify cryptographic keys, this use-case cannot be supported by FIDO protocols.

Transport Layer Security (TLS)

TLS and its deprecated predecessor, Secure Sockets Layer (SSL), are public-key based cryptographic protocols designed to provide privacy, authentication and data integrity between two or more communicating applications. When secured by TLS, connections between a client (e.g., a web browser) and a server (e.g., mybank.com site), the server authenticates to the client using an X.509 server certificate, and the client optionally authenticates to the server using an X.509 client certificate. Client and server certificates are issued by trusted Certificate Authorities (CAs). Applications using TLS may use other methods of client or user authentication to the server besides X.509 client certificates - such as (HyperText Transfer Protocol) HTTP Basic or HTTP Digest authentication. While FIDO protocols do not support certificate-based authentication and are not designed for client or server authentication in TLS, they may be used for client authentication at the application layer in lieu of or in addition to X.509 client certificate authentication.

EAP (Extensible Authentication Protocol)-TLS for Wireless Access

This use-case cannot be supported by FIDO protocols. Historically, device-to-device authentication within enterprises have been used with the EAP-TLS protocol, which mandates the use of an X.509 digital certificate. Since FIDO protocols do not use digital certificates to identify cryptographic keys, this use-case cannot be supported by FIDO protocols.

Transaction Authorization

This use-case is supported by two FIDO protocols – UAF and FIDO2. UAF and FIDO2 support two features necessary for this capability to be achieved:

- A Secure Display which is not under the control of the application developer – since it can enable attackers to subvert content being signed by users; and
- Support in the protocol to request a digital signature from the user using the Secure Display.

While Transaction Authorization is intended for small transactions that can fit within a small Secure Display of a mobile device, it is possible to build an application capable of displaying a *message digest* (aka *hash*) of the document which can be verified outside the application. A message digest is sufficiently small that it can be displayed within the Secure Display of a mobile device.

Document Signing

Technically, Transaction Authorization can digitally sign the hash of a data-object and/or document. As long as an application can be written to sign such a data-object or document, it can be accomplished with FIDO. However, the verification of that signature is easiest for a relying party (RP) where the signer's FIDO public-key is registered. Verifying that signature outside the domain of the RP becomes a complex issue which FIDO protocols are not designed to solve, and which are better addressed with PKI. But, this does not prevent an RP from building an application that would allow them to have users with registered FIDO Authenticators to sign documents, which they (and/or their affiliates that share the same FIDO server's database) can verify through their application.

Code-Signing

Like document signing, if an RP has built an infrastructure that allows them to verify signed code within their domain, the FIDO protocols can support that. Verifying signed code outside the domain of the FIDO RP will be challenging and possible only if the verifier has access to the FIDO server's public database. For the moment, the challenges of code signing, and verification are better addressed using PKI.

Disk Encryption

This use-case cannot be supported by FIDO protocols. Disk encryption can be implemented using different techniques including the use of smart cards. FIDO protocol is designed and is more suitable for user authentication and transaction signing.

Single Sign-On

When combined with federation protocols such as Security Assertion Markup Language (SAML) and OpenID Connect (OIDC), FIDO protocols can be used to enable a single-sign on experience. In this case the FIDO RP acts as an Identity Provider (IdP) and issues authentication, attribute and authorization assertions upon a user successfully authenticating using FIDO. The IdP has the flexibility to request the user re-authenticate based on business and security policy.

Trust Establishment

In federation protocols such as SAML and OIDC, a trust relationship is established between two parties: the identity provider (IdP) and the RP. The trust is typically established through the exchange of certificates. A signed RP request to authenticate a user is verified and authenticated by the IdP using the RP's digital certificate, while a signed assertion or identity token created by the IdP is verified and authenticated by the RP using the IdP's digital certificate. FIDO protocols are not designed to establish trust between the IdP and the RP.

Combined PKI and FIDO Authentication Solution for the Enterprise

Enrollment of FIDO credentials in a mixed PKI and FIDO environment

Enterprises and government agencies that currently have a PKI might have an elaborate identity assurance process to issue digital certificates to their users. This may involve a range of actions such as:

- Showing up physically at a Credentialing Facility
- Providing one or more government-issued identification credentials to establish their real-world identity
- Providing an enterprise-issued identification credentials to establish their enterprise identity

While not every organization may use these steps, it is safe to assume that a PKI-issued credential to their users is predicated on verifying the identity of the user in some manner that is deemed acceptable to the business.

Existing Users with a PKI Credential

Where organizations have a PKI, they may use the TLS ClientAuth protocol inherent in their existing users' smartcards and digital certificates – such as in the US Department of Defense's Common Access Card (CAC), or the US Federal Government's Personal Identity Verification (PIV) card - to enroll authorized users with a FIDO

credential. By leveraging the trust inherent in the PKI credential issued by the enterprise or agency, the RP can save significant resources by not having to repeat the identity assurance process when establishing a FIDO credential for such users.

Figure 1 illustrates the key components and the main steps a PKI-user goes through to register a FIDO Authenticator with a RP without repeating the identity assurance process.

The following elements are necessary:

- The user must have a valid digital certificate trusted by the RP – this may be a soft certificate or hardware-based certificate such as a smartcard;
- The RP must have a web-application that enables the user to strongly-authenticate to the application using, for example, the TLS ClientAuth protocol. To ensure that only authorized users can register FIDO Authenticators with this RP, the digital certificate-based strong-authentication must include the standard PKI Extensions (PKIX) Validation process: checking a CRL or an OCSP Responder to validate the full certificate chain of the end-entity;
- Upon successfully completing the TLS ClientAuth and PKIX Validation process, the web-application must enable the registration of a FIDO Authenticator and store the user's *key-handle* within a persistent store where it may be used with other FIDO-enabled web-applications;
- While this is optional, upon successfully registering a new FIDO Authenticator with the RP, it will be extremely beneficial to have a second web-application which is FIDO-enabled to test the newly minted FIDO credential. This will confirm to the user that their FIDO credential works as intended.

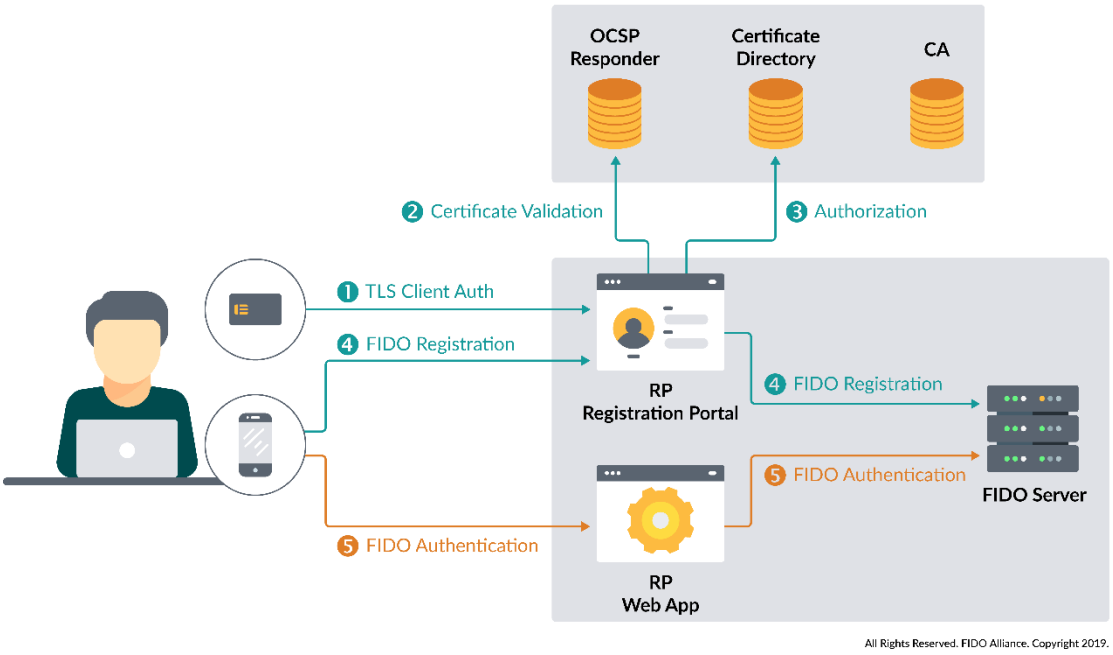


Figure 1: PKI to FIDO Process and Subsequent Authentication Flow

While there are many ways this paradigm may be implemented, readers of this paper may benefit from reviewing the design and implementation of the open-source web-application in Appendix A.

New Users without a PKI Credential

When a new user is enrolled within an organization that has a PKI, the question arises: should the user be enrolled in the PKI first and leverage the trust of the PKI credential to enroll their FIDO credentials? Or, should the user be enrolled with a FIDO credential first and leverage the trust of the FIDO credential to acquire a PKI credential?

The answer depends on the organizations' policy. It is technically feasible to enroll a new user from either direction depending on several factors:

- The initial credentialing process and the trust enabled within that process
- The degree of trust established in the initial credential and the security/capabilities of the device used for generating and storing cryptographic keys for the credential
- Other factors, such as whether the user is enrolling at a "local" versus a "remote" facility of the organization

The recently released National Institute of Standards and Technology (NIST) Digital Identity Guidelines Special Publication 800-63-3[11] provides guidance for the separation of the assurance of a credentialing process from the security capabilities of the authenticator used for the credential. The Guidelines can be used to define processes and controls that enable organizations to enroll PKI or FIDO credentials in a manner that complies with the organizations' policies.

Enrolling Secondary FIDO Authenticators

Most PKIs do not choose to support a policy that allows the same user to enroll for multiple digital certificates that are active at the same time for the same purpose with the same Subject Distinguished Name. There may be extenuating circumstances, but the generally accepted practice is to issue a single, unique, valid digital certificate per user. However, if a user's digital certificate has expired or revoked for a reason, the user, at the discretion of the organization, may allow the user to renew or issue a new digital certificate to the user (see next section for more detail).

FIDO protocols are different. They permit the use of multiple cryptographic keys to be registered to the same user at the same time, with the only caveat being that the key-pair not be generated and stored in the same Authenticator for the same user to be used against the same RP web-origin of the application – i.e. the triplet: username, RP web-origin and Authenticator must be unique to be successfully registered.

While this might seem strange to people from the PKI world, the use of multiple Authenticators for a user with unique registered keys to the same RP web-origin is encouraged to prevent the loss of a single Authenticator that triggers complex "account recovery" processes. With multiple keys on unique Authenticators registered to the same web-origin, when a user loses an Authenticator, RPs may allow users to authenticate to the applications with a secondary Authenticator, delete the key of the lost Authenticator and, optionally, register a new key with a new Authenticator, and continue with normal business. Even if the lost Authenticator is later found, it cannot be used to authenticate to the RP's application since the key registered from that Authenticator was deleted from the FIDO data-store at the RP.

Revocation/Expiration of credentials in a mixed PKI and FIDO environment

Typically, digital certificates have a limited lifetime. The duration is intended to balance several risks against the inefficiency of reissuing certificates too frequently and varies from organization to organization; but a general

[11] https://pages.nist.gov/800-63-3/

practice is a one or two-year client certificate. When the certificate is due to expire, users are, typically, reminded to renew their certificates using a predefined process.

FIDO protocols do not specify duration for registered keys with an RP. They are raw cryptographic keys and metadata associated with such a key is only stored in the FIDO data-store, contingent upon the implementation of the FIDO data-store. From a protocol point-of-view, FIDO credentials never expire.

When a digital certificate is revoked for any reason, it is typically done before the natural lifetime of the certificate has expired. The revocation is published through a Certificate Revocation List (CRL) or through an Online Certificate Status Protocol (OCSP) responder. This is intended to notify relying parties that a digital certificate, upon which the RP may be assuming trust, has been revoked as of a specific date and time, and any RP should not rely on that certificate for any reason after the timestamp in the CRL or the OCSP response.

Since FIDO credentials do not have durations, no such revocation action exists to publish or notify RPs. However, much as shared-secret based credentials are deleted or deactivated from a credential data-store, based on an RP's policy using an implementation-specific process[12], FIDO credentials can be similarly deleted or deactivated using implementation-specific processes. If the user whose key is deleted/deactivated has a secondary FIDO Authenticator registered to their account, they may continue using the Authenticator associated with that key if the RP has permitted them to do so. In use-cases where the RP chooses to disallow the user to authenticate using any of their registered keys, the RP may delete or deactivate all their registered keys.

In the event a previously registered user has all their keys deleted or deactivated by the RP and is now permitted to register a new FIDO credential, it is expected that the user will go through the same process as any user enrolling a new key with the RP. Whether the previously registered user must repeat a credentialing process to register a new FIDO credential is up to the RP's implementation for registration. Whether the revocation of a PKI credential should cascade to deleting/deactivating all registered FIDO credentials of a user or vice-versa, depends on a specific implementation of the PKI and FIDO. Organizations are likely to want to enable this capability and FIDO protocols neither support nor prohibit such capability.

Account Recovery

Use PKI Credential for Account Recovery

Organizations that have implemented a PKI, typically, have established processes for recovering a user's account due to a lost or stolen PKI credential. If a FIDO credential is enrolled based on the trust of a PKI credential in such organizations, what happens when the user loses the FIDO Authenticator with their registered key?

The answer to this depends on the policies established by the organization with respect to the use of FIDO technology within the organization. FIDO Alliance published a whitepaper that addresses the question and provides recommendations to both, enterprises and consumer service providers[13].

If the organization requires users to register multiple FIDO Authenticators – and consequently, multiple FIDO Authenticators – losing any one FIDO Authenticator permits the user to use another FIDO Authenticator to

[12] Unlike U2F or FIDO2, UAF protocol supports a delete operation on the authenticator in conjunction with the application. However, this is not a major issue considering all FIDO protocols are designed to be used only in a "scoped manner" – implying that even if a FIDO key-pair exists on a specific authenticator for a user, if the RP has deleted the FIDO credential of that user on the FIDO server, the authenticator will never use that particular key-pair since it will never be referenced in any FIDO protocol message from the RP site. While it may consume some amount of storage space on the authenticator, the market is expected to respond to this need by providing tools to help the user perform such credential maintenance activities independent of the RP application.

[13] https://fidoalliance.org/recommended-account-recovery-practices/

authenticate to their application. The user would then delete the key associated with the lost Authenticator, and optionally, register a new Authenticator (and key) with the account. However, when an RP permits users to register multiple keys from different Authenticators to its site, it must ensure that all Authenticators used by users meet its minimum required security policy. This prevents the user from registering a key generated on an Authenticator with a lower security certification level to replace a lost Authenticator that was certified at a higher security level – such a lapse in security can lead to potential compromises by unauthorized users despite the use of FIDO.

If the organization chooses to allow users to use just a single FIDO Authenticator and the user has lost that Authenticator, or the user has lost all Authenticators, it is feasible for the organization to enable users to use their valid PKI credential to enroll a new FIDO Authenticator. When registering a new FIDO Authenticator after being authenticated by a PKI credential, it is naturally assumed that the organization will have implemented a process to determine whether the user is registering an additional FIDO Authenticator for convenience or backup, or replacing a lost FIDO Authenticator; in the case of the latter, the organization should delete the lost FIDO Authenticator to prevent the lost Authenticator from being used to authenticate to application(s) should it be found again.

Operations in a mixed PKI and FIDO Environment

Organizations with PKIs, typically, have one or more individuals responsible for PKI operations. When FIDO is introduced into such organizations, it is logical for the PKI Operations group to be responsible for FIDO Operations. While the details of managing a PKI and FIDO infrastructure are likely to be specific to individual implementations, the knowledge and experience accumulated in the PKI Operations team can be leveraged to manage FIDO credentials.

In contrast to PKI, FIDO operations are much simpler. Once the registration and authentication policies are set in a FIDO system including which type of Authenticators and assurance levels are accepted, the Operations team does not need to worry about generating, binding, certifying and provisioning credentials for users or about the expiry and the renewal of those credentials. However, given that FIDO is relatively new and FIDO credentials life-cycle management aspects are kept outside the scope of FIDO protocols, it would be safe to assume that organizations with PKI might want to consolidate FIDO operations with PKI operations to leverage experience, knowledge and conserve costs.

Benefits of the Combined PKI and FIDO Approach

The implementation of FIDO can benefit organizations in many ways. It offers strong public-key based authentication that is equivalent to certificate-based authentication but without the overhead of maintaining complex and expensive public-key infrastructure. FIDO can be used to address the needs of an enterprise for a variety of applications and use cases, including web and SaaS applications access, mobile applications, offline and online desktop logon, access to shared workstations, strong authentication for remote and non-employees workforce (contractors and temporary workers, vendors, partners, and guests).

Organizations that invested heavily in PKI and want to modernize their authentication solution should take a phased approach to enable FIDO based authentication. FIDO could be implemented initially with minimal investments to protect access to cloud-based web and mobile applications for employees and non-employees. Existing user certificates could be used to bootstrap FIDO credentials for these applications. Overtime, they can be used to enable FIDO based authentication for other applications and use cases including for desktop logon and remote logon to legacy, on-premises applications.

Organizations who have not invested in PKI and are looking for strong authentication solutions should consider investing in FIDO if FIDO meets their security and business objectives.

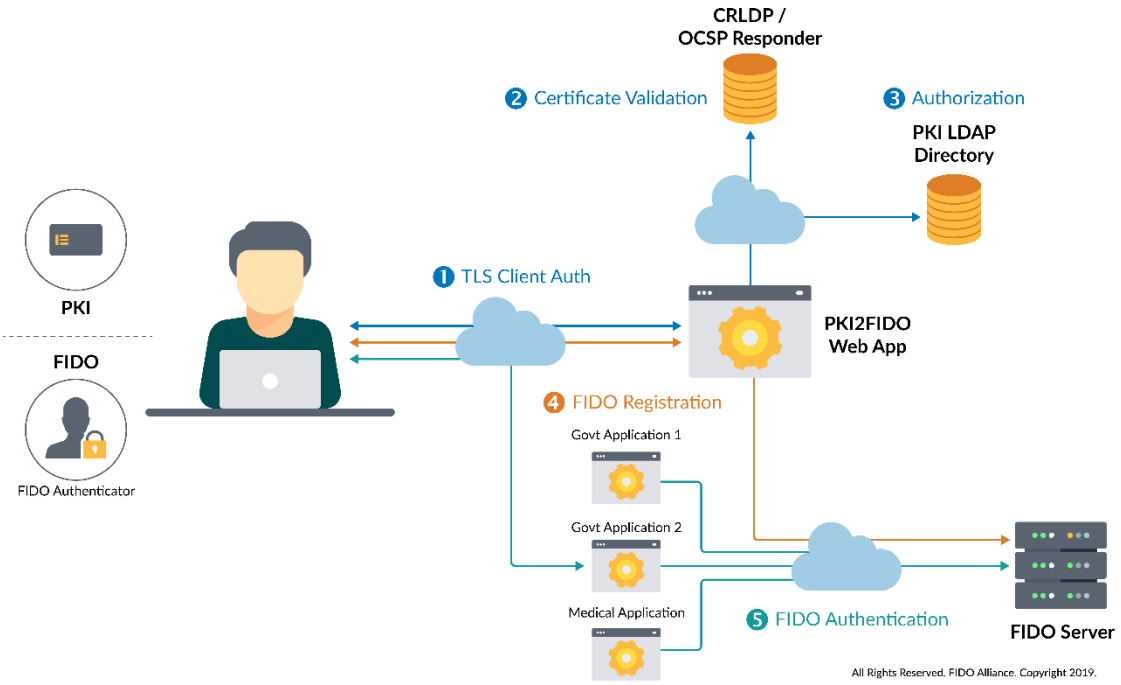
Acknowledgements

The editors would like to thank all FIDO members who reviewed or contributed to this paper, namely

- David Treece, Yubico
- Andrew Regenscheid, NIST
- Ingo Schubert, RSA
- John Fontana, Yubico
- Giridhar Mandyam, QUALCOMM
- Rae Barton, Gemalto
- Mingliang Pei, Symantec
- Karen Chang, Egistec
- Bill Leddy, Visa
- Bre McGahey, FIDO Alliance
- Andrew Shikiar, FIDO Alliance

Appendix A: PKI2FIDO Open Source Project

While there are many ways this paradigm may be implemented, readers of this paper may benefit from reviewing the design and implementation of an open-source web-application. The web-application, , available to anyone in the world to download and use without a licensing fee uses the following architecture to enable users with a digital certificate to register a U2F key with an open-source FIDO Certified U2F server:



In this picture, the user with a smartcard and digital certificate is strongly-authenticated using TLS ClientAuth in the flow depicted with the blue arrows. Once strongly-authenticated, the user registers a new U2F key with the FIDO Server[14] in the flow depicted with the red arrows. Finally, the user is shown in green using the newly generated FIDO Authenticator to strongly-authenticate to a few different FIDO-enabled web applications that are connected to the same FIDO Server used by the PKI2FIDO web-application.

Readers of this paper are cautioned that this is an illustration. A "real-world PKI2FIDO" type web-application must be configured for a RP's specific environment.

[14] It should be noted that the FIDO U2F server depicted in the flow can either be deployed on-premises or be available as a service to RPs.