Practical Implications of Public Key Infrastructure for Identity Professionals

By Robert Sherwood

To comment on this article, please visit our GitHub repository and submit an issue.

Table of Contents

ABSTRACT **2**

INTRODUCTION **2**

Terminology 2

BASICS OF PKI FOR IDENTITY PRACTITIONERS **6**

What is PKI 6
How do a 'Private Key' and a 'Public Key Certificate' Provide Authentication Assurance 7
Public and Private Keys *7*
Public Key Certificates *8*
Who Can Get a Certificate 9
How Are PKI Certificates Like Other Credentials, and How Are They Different? 10
Factors and Problems Limiting PKI Adoption 12

UNIQUE CONSIDERATIONS FOR IDENTITY PRACTITIONERS **14**

Ensure that PKI is the Right Fit for Your Requirements 14
The Importance of Planning 14
IGA and PKI 14
Lifecycle Management of PKI Certificates Compared to Other Credentials 15
Options for Identifiers in Public Key Certificates 16
Machine Identities and Certificate Management Systems 17
Federated Authentication and PKI 17

CONCLUSION **18**

AUTHOR BIO **19**

Abstract

Public Key Infrastructure, or “PKI,” is a technology that enables authentication via asymmetric cryptography. It is widely deployed for some vital security use cases on the Internet, especially for the authentication of servers via Transport Layer Security (TLS).

Despite its wide use in some scenarios, there are significant challenges in deploying PKI for more widespread use among smaller organizations or consumers.

Identity Professionals who need to deploy a PKI or have inherited a deployed PKI from someone else have several important considerations, including lifecycle management of keys and certificates, choosing the appropriate way to encode user identifiers, and understanding cross-PKI trust.

Introduction

In high-risk environments containing extremely sensitive data, every participant must have high confidence in the identity of every other participant. Public Key Infrastructure (PKI) is one of the most long-lived and widely deployed authentication technologies in these high-security environments. Despite the difficulty in deploying PKI for end users, which we discuss below, PKI was the only high-assurance credential available in the commercial market for many years.[i] It is still considered the gold standard of credential assurance by many experts. Military and government environments have used PKI to provide secure authentication in sensitive environments since the late 90s.

Despite the widespread adoption of PKI in government environments, PKI has yet to see the same success in commercial settings. Later in this article, we will discuss some of the reasons for the lack of widespread adoption.

Despite the difficulties, PKI can be a feasible alternative to passwords for some enterprises, thanks to the implementation of smartcard-based authentication in many operating systems and browsers. Enterprises have renewed interest in smartcard login to eliminate passwords for privileged users in high-risk environments and scenarios.

This article includes analysis and guidance for the deployment of PKI for both human users and machines.

Terminology

- Asymmetric Cryptography: Any cryptographic algorithm which depends on pairs of keys for encryption and decryption. The entity that generates the keys shares one (see Public Key) and holds and protects the other (see Private Key). They are referred to as asymmetric because one key encrypts, and the other decrypts.

- Automatic Certificate Management Environment (ACME): A communication protocol for automating lifecycle management of PKI certificates. Significant providers like Let's Encrypt leverage ACME to support issuing TLS certificates for web servers.
- Certificate Authority Trust List (CTL): A client maintains a list of trusted Certificate Authorities created and managed by the software provider or local administrators. The client will only trust certificates issued under one of the CAs in the CTL, so the CTL serves as a "safe list."
- Certificate Management System (CMS): A system that provides management and reporting layers for certificate issuance and revocation. A CMS integrates CA products with Identity Governance and Administration (IGA) systems as well as Service Desk systems.
- Certificate Policy (CP): A document that defines the high-level policy requirement for a PKI. RFC 3647 identifies a PKI's policy framework and describes a CP's contents and outline. An enterprise operating a CA will often publish its certificate policy to external parties so they can determine whether to trust certificates issued by the CA.
- Certification Practices Statement (CPS): A CP identifies the requirements for managing a CA and issuing PKI certificates. A CPS describes how a CA implements those requirements. The CPS uses the same outline as the CP, defined in RFC 3647. Unlike the CP, enterprises rarely publish their CPS in unredacted form.
- Certificate Revocation List (CRL): A certificate authority will publish a list of revoked certificates, called a CRL so that clients can verify that a certificate is still good.
- Certificate Signing Request (CSR): When requesting a certificate, the requesting entity provides a copy of the public key, their identifiers, and other information in a specially formatted binary object called a CSR.
- Classical Computer: A computer that uses binary encoding and Boolean logic to make calculations in a deterministic way. We use the term Classical Computers in contrast with Quantum Computers.
- Cryptographic Module: A hardware or software component that securely performs cryptographic operations within a logical boundary. Cryptographic Modules store private keys within this boundary and use them for cryptographic functions at the request of an authorized user or process.
- Cryptographic Module Validation Program (CMVP): A program allowing cryptographic module developers to test their modules against the requirements defined in FIPS-140. The computer security resource center under the United States National Institute of Standards and Technology (NIST) maintains a publicly available list of validated modules.
- Electronic Identification, Authentication, and Trust Services (eIDAS): European legislation gives legal standing to electronic signatures under eIDAS. This legislation also documents providing legally binding digital signatures with X.509 certificates to comply with Qualified Signature requirements.
- Elliptic Curve Cryptography (ECC): An asymmetric cryptosystem based on calculating points along elliptic curves.

- Encryption: Processing data using a cryptographic algorithm to provide confidentiality assurance.
- Federal Agency Smart Credential Number (FASC-N): A unique identifier associated with a smart card. FASC-N is used in the US Federal Government PIV standard to support Physical Access.
- Federal Information Processing Standard (FIPS) 140: A NIST standard defining "Security Requirements for Cryptographic Modules."
- Groups: A set of identities with defined permissions. In this specific context, a group contains many individuals, but the group identity is opaque, and no information is available regarding which group member took an individual action.
- Hardware Security Modules (HSMs): A hardware cryptographic module that generates and protects cryptographic keys.
- Identifier: The way a system refers to digital identity. PKI Certificates support both internal and external identifiers. See Ian Glazer's article, "Identifiers and Usernames."[ii]
- Internet Key Exchange (IKE): A subordinate standard under IPsec specifying how to use X.509 certificates to establish symmetric keys for an IPsec tunnel.
- Internet Protocol Security (IPsec): A standard for communication between two machines providing confidentiality and integrity over the Internet Protocol.
- Key: In a cryptosystem, a Key is a piece of information used to encrypt or decrypt data in a cryptographic algorithm.
- National Institute of Standards and Technology (NIST): A US Government agency that defines and publishes various standards. One department within NIST, the Computer Security Resource Center (CSRC), publishes the Federal Information Processing Standards (FIPS) series. While these standards are only mandatory for US Government Agencies, many countries recognize them as de-facto global standards.
- Non-person entities (NPE): Any unique combination of hardware and software firmware (e.g., device) that utilizes the capabilities of other programs, devices, or services to perform a function. Non-person entities may act independently or on behalf of an authenticated individual or another NPE.[iii]
- Online Certificate Status Protocol (OCSP): A protocol that allows a client to query the Certificate Authority or a Validation Authority for the status of an individual certificate rather than downloading a CRL.
- Path Discovery and Validation (PDVal): The process to determine whether a certificate is valid and trusted by the validator.
- Personal Identification Number (PIN): A numeric secret commonly used to unlock a private key container in software or hardware.
- Personal Identity Verification (PIV): A US Government program designed to enable strong authentication for all government employees and contractors, based on Public Key Infrastructure.
- Private key: A key that a single entity exclusively and privately controls. It corresponds to a public key that the entity may share for data encryption or signature verification.

- Public key: A key that an entity publicly distributes. It corresponds to a private key that the entity exclusively and privately controls.
- Public Key Certificate: A certificate containing a public key, one or more identifiers for the private key holder, an identifier for the Certificate Authority, and additional metadata to support security requirements.
- Public Key Infrastructure: A set of tools, standards, and related policies designed to manage trust based on public/private key pairs and certificates.
- Registration Authority (RA): An individual, system, or business function which provides registration and identity proofing for entities receiving certificates and manages the certificate issuance and renewal process. The most important responsibilities of an RA include identity proofing and binding the private key to the identity.
- Revoke: Revocation is the announcement that clients should no longer trust an individual certificate.
- Roles: A set of permissions. A role must be associated with an individual user, and the user gains the associated authorization when they are associated with the role.
- RSA: An asymmetric cryptosystem based on large prime numbers. The acronym RSA stands for the three principal inventors, Ron Rivest, Adi Shamir, and Len Adleman.
- S/MIME: A standard for constructing and sending digitally signed or encrypted messages using asymmetric cryptography.
- Secure Socket Layer (SSL): A deprecated standard for encrypting data in transit; TLS has superseded it.
- Server-based Certificate Validation Protocol (SCVP): A protocol that allows a client to query a server to determine whether a certificate is valid and trusted. The server does not need to be associated with the issuing CA. SCVP does two things; (1) it determines the path between the end entity and the trusted root, whereby the client doesn't need to trust any intermediate CAs. (2) it also performs delegated path validation according to policy.
- Signature: Processing data using a cryptographic algorithm to provide integrity assurance.
- Subject Alternative Name: One or more identifiers for a certificate subject that certificate issuers can use to carry application-specific identifiers such as email address or User Principal Name (UPN).
- Subject Distinguished Name (Subject DN): A unique identifier for the subject within the scope of the Certificate Authority. Issuers structure the subject DN like an LDAP entry name.
- Transport Layer Security (TLS): A cryptographic protocol designed to provide confidentiality and integrity of communications between two endpoints.
- X.509: An ISO standard from the X.500 series that defines the basic rules for encoding public key certificates.
- Validator: An entity that verifies a certificate and confirms that the other party controls the private key in the transaction.

Basics of PKI for Identity Practitioners

What is PKI

PKI stands for "*Public Key Infrastructure,*" a set of interlocking standards and technologies that support the secure exchange of public keys for *asymmetric cryptography* use cases.

Originally developed as part of the X.500 series of specifications for electronic directory services, the *X.509* standard proposed a way to link a public key into a universal, hierarchical directory designed to support OSI networks.

The OSI protocol is, for all intents and purposes, dead. However, the X.500 specification lives on in simplified form as LDAP, and X.509 has found a second life in the modern Internet.

PKI is woven deeply into the fabric of the Internet, and it supports the following critical Internet capabilities:

- *TLS* as a general encryption layer for application protocols
- *S/MIME* as a standard for secure email
- *IPsec* as a standard for virtual private networking (VPN), which depends upon PKI via the Internet Key Exchange (IKE) extension
- Some commercial software or services, such as Adobe Acrobat, Microsoft Word, or DocuSign, support digital signatures for non-repudiation or integrity protection. In Europe, qualified signatures and time stamps have official legal standing, recognized in the Electronic Identification, Authentication, and Trust Services (eIDAS) framework.

This article is not a general primer on PKI; it provides a minimal overview of PKI as it relates to identity Management and identifies critical issues relevant to identity Practitioners. Interested readers are referred to the references section at the end for more detail.

Here are some excellent resources to learn more about PKI in general:

Books:

- Applied Cryptography, by Bruce Schneier, is a classic guide to the cryptographic technology underlying PKI and its applications. For those who want to know everything about this subject, this is the place to start.

Online Resources

- The US Federal government has deployed PKI widely for both logical and physical access. IDManagement.gov maintains information about the Federal PKI here: https://playbooks.idmanagement.gov/fpki/

- Bruce Schneier, the author of Applied Cryptography, maintains a fascinating and helpful blog here: https://www.schneier.com/

Standards:

- X.509: The original specification for PKI certificates. This document must be purchased.
- RFC 5280: The Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile standard specifies a subset of the X.509 standard for use on the Internet.

How do a 'Private Key' and a 'Public Key Certificate' Provide Authentication Assurance

Public and Private Keys

Private and *public keys* are random numbers, but not just any random number.

- In the *RSA* specification, keys are derived from a large prime number.

- In *ECC,* keys are related to points along a particular elliptical curve.

By taking some data, such as text or an image, and plugging the data into a specific equation with one of the numbers (keys), you create a scrambled version of the data that only the other number (key) can unscramble. This concept is the basis of asymmetric cryptography.

The *private key's* owner must retain and closely guard it since a foundational assumption in PKI is that only the authorized user controls the private key. The public key, by contrast, can be widely shared.

A sender can scramble a message using the public key to send a message only the private key's owner can read. Because the private key is the only key that can unscramble and read the message, the sender knows that the message can only be read by the private key owner.[iv]

The owner of a private key can use it to scramble a message, and a recipient can only unscramble the message with the public key. The recipient can be sure that the private key owner sent the message and that it has not been modified in transit.[v]

In asymmetric cryptography, “*encryption”* refers to scrambling data with the public key, and “*signature”* refers to scrambling data with the private key.

In practice, signature and encryption are much more complicated, involving cryptographic hashes or intermediate symmetric keys. For our purposes, it is sufficient to understand that private keys sign and public keys encrypt.

![Illustration of Encryption and Signature path for a document](PKI.jpg)

Despite the widespread use of PKI for highly secure credentials, asymmetric cryptography does not directly provide authentication! Authentication protocols that leverage PKI credentials depend on signature or encryption.

In public-key authentication schemes, the user is whoever has control of the private key. When a system wishes to authenticate a private key owner, it requires them to use the private key they own. The user can sign something with the private key that the system can verify with the public key or decrypt something with the private key that the system encrypts with the public key.

The user can provide a signed message for the authenticating system to verify, or the authenticating system can generate and encrypt data that the user can only decrypt with their private key. In both scenarios, possession of the private key, demonstrated by the ability to use the private key to decrypt or sign data, proves the user's identity.

Public Key Certificates

So far, we have seen how to authenticate a user if you have their public key. The challenge that remains is finding a reliable way to exchange public keys.

We have solved this problem in several ways with different protocols and systems. Many modern authentication protocols, including FIDO, Verifiable Credentials, and passkeys, leverage public/private keys and asymmetric cryptography. Every protocol requires an out-of-band process that links the public key with a public identifier. In some contexts, notably in the SSH public-key authentication protocol or "Web of Trust" based systems like PGP, pre-existing relationships or pre-provisioned authorizations are sufficient.

In the context of complex modern business processes where you are unlikely ever to meet the majority of people you interact with, users cannot simply exchange keys directly. After all, in the absence of some way to verify the identity of the individual providing you with a public key, you have no way to distinguish between your intended counterparty and an imposter.

PKI solves this issue by relying on the concept of a "trusted third party." In a PKI, a central trusted authority vouches for identities according to a documented process. This centralized authority introduces scalable trust by allowing users to verify the identity of previously unknown users or systems. The users rely on the centralized authority to enforce an identity registration and lifecycle management process. The mechanism used in PKI to convey this assurance is the public-key certificate.

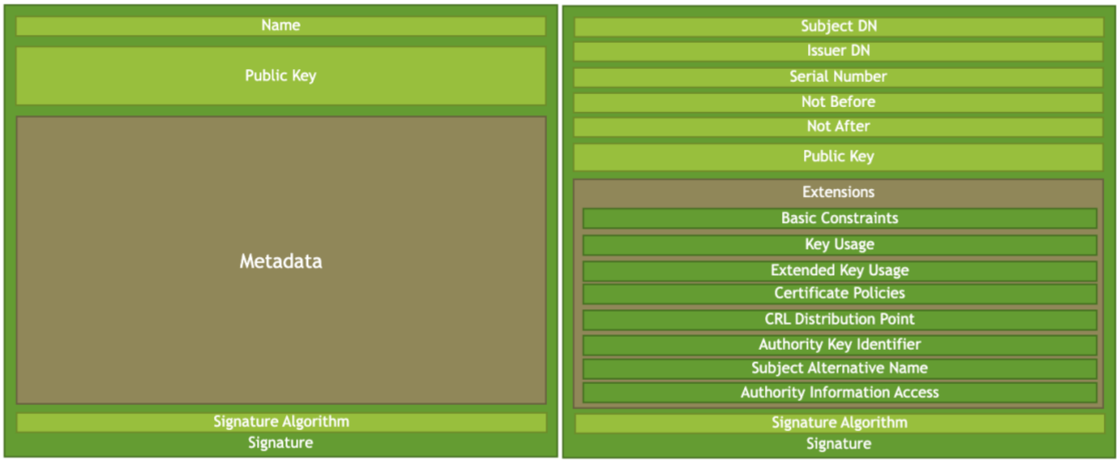
For every participant to have confidence in distributed business transactions, each participant must have confidence in the identity of every other participant. For asymmetric encryption to support business applications, the public key must be connected, or "bound," to the participant's identifier. In PKI, public key certificates are the artifacts that connect a public key and an identifier.

A *public key certificate* contains several critical pieces of information. For authentication purposes, the following three fields are the most important:

- The public key

- One or more identifiers associated with a user

- Information about the “trusted third party” that vouches for the association between the key and the identifier.



Figures 1 and 2: Key components of a PKI certificate that support identity including Name, key, metadata, signature algorithm, and signature. Additionally, a detailed listing of several possible elements of a PKI certificate.

A public key certificate is a file with a prescribed structure defined by the X.509 v3 standard and refined by RFC 5280. It contains the user's public key, their identifiers, and important metadata about the certificate itself.[vi] The file is digitally signed using the private key of a trusted third party, called a "Certificate Authority."

Who Can Get a Certificate

Any business process participant who can generate and store a private key and associated public key may receive a certificate. The most common recipients of certificates are listed here:

- Humans: A human being can receive a public key certificate that names them individually.

- *Non-person entities:* Examples of non-person entities include devices like routers, software services like web or email servers, IoT devices, and other non-human entities like software providers who digitally sign software packages.

- *Roles:* Sometimes, a person may act in a role, such as "Software Release Manager" or "Doctor on call." A certificate authority can issue a certificate that identifies the user's role, allowing them to authenticate in the persona of that role. Role certificates are issued to individuals and contain a personal identifier for the person holding the private key to maintain individual accountability. Everyone with a role certificate has a unique private key.

- *Groups:* In some exceptional cases, several people share a private key. In this case, a certificate authority can issue a certificate to a group. The certificate will identify the group, and the group members will take additional security precautions to ensure that only authorized members use the private key.

How Are PKI Certificates Like Other Credentials, and How Are They Different?

Users can authenticate themselves with a private key and corresponding PKI Certificate, like other credentials.

- The trustworthiness of the credential depends on the identity proofing and issuance process as much as it depends on cryptographic math. As with other credentials, the identity assurance level for authenticated users is low if the proofing or issuance processes are insecure or the user does not protect the private key.
- Like other credentials, a private key and certificate are a single authentication factor that enterprises can supplement with additional factors. Typically, we consider a key and certificate "something you have" and often supplement it with a PIN, Password, or biometric.

PKI credentials have many unique properties not shared by most other authentication credentials.

A public-key certificate file contains all the information necessary to authenticate the subject:

For most other credential types, each authentication challenge requires the involvement of the credential issuer. When a user enters a password, the authenticating system must check it against the directory or database where the user created their account. By contrast, PKI authentication can occur without directly interacting with the issuing Certificate Authority. The user generally activates the private key with a secret, such as a

PIN or a password, but his secret is entered directly into the software or device containing the private key; the user does not provide it to the Certificate Authority.

Public key certificates are long-life credentials:
Certificates may be valid for a much longer-term than is typical for other credential types. It is common for a certificate authority to issue a public key certificate to a user with a three-year lifetime. This extended lifetime is acceptable because the private key credential is not user-selected and is too long to be easily memorized or copied by humans.

Key protection affects the overall security of the PKI credential:
Like any other authentication secret, the user must protect a private key from third parties to prevent the third party from impersonating the user. Recall that in public-key cryptography, the user is whoever controls the private key. For this reason, it is essential to ensure that private keys cannot be copied or taken without a user's awareness and permission. Because private keys are usually very long and appear random, they cannot be memorized and must be stored.

Several technologies are available to protect private keys, including *Hardware Security Modules (HSMs)* or personal tokens such as the YubiKey Security Key or SafeNet eToken Smart Card. The United States *National Institute of Standards and Technology (NIST)* has published a standard, *Federal Information Processing Standard (FIPS) 140,* and has implemented the *Cryptographic Module Validation Program (CMVP)* to ensure that HSMs implement proper cryptographic algorithms and key protections for private keys.

The security properties of PKI credentials mean they can provide a higher level of identity assurance than other kinds of credentials. Governments reserve the highest levels of assurance defined by governments for PKI certificates stored on smart cards. This security comes at a price in terms of direct costs and additional complexity.

PKI credentials can support additional use cases beyond interactive authentication:
While passwords, OTP, and other credentials are limited to interactive authentication, PKI credentials are suitable for transactions that are not immediate and interactive. One example is a digital signature, where the recipient of a signed message must know the signer's identity, but the signer may not know who will verify the signature. Encryption is another case where the encryptor of the sensitive data must ensure that the intended recipient is the only one who will have access even when data is exchanged out-of-band and asynchronously. PKI can enable capabilities such as S/MIME, Qualified Signatures, and others that cannot be supported by credentials that only provide authentication.

Factors and Problems Limiting PKI Adoption

The roots of PKI extend back to the 1970s, and the earliest versions of the Secure Sockets Layer (SSL) standard cemented its use as the basis for secure communication in the mid-1990s. However, despite its maturity and widespread use for some specific use cases, it has yet to see broad adoption for authentication of individuals, either for business-to-consumer or business-to-employee use cases. There are many reasons why PKI has yet to see widespread adoption outside these narrow use cases, though the technology and vendor support has improved. The following are some of the most significant challenges hindering adoption:

Enterprise key management is challenging:
For PKI to be a trustworthy and secure authentication approach, the private key must be controlled exclusively by the authentication subject. As we said earlier, the user is whoever controls the private key. There are two ways to ensure that the intended user is the only one with access to the private key. The authentication subject must generate the private key within a protected software environment, or the CA must generate the private key on the subject's behalf and then pass it to the subject using a secure transfer mechanism. Both processes are complex and challenging to automate without extensive tooling.

Internet software providers have focused on providing automation for critical technical use cases, such as TLS for Web Servers. Protocols like *Automatic Certificate Management Environment (ACME)* and services like Let's Encrypt provide zero-touch key management and certificate rotation for web servers. These services do not support the management of certificates issued to humans.

Vendors, meanwhile, have implemented sophisticated, proprietary solutions for the automation of key management. Microsoft Active Directory Certificate Services can provide key management and certificate services for machines and human users in an Active Directory environment. The Entrust Certificate Authority provides a client-side tool to manage the lifecycle of keys and certificates for clients. However, these tools and others like them are tied to a specific product and are part of a closed, proprietary system.

Other providers, like KeyFactor or Venafi, can provide certificate lifecycle services for a mix of CA products. However, these tools are proprietary and may require significant integration efforts.

PKI has poor usability:
As discussed above, key management is a complex organizational and technical issue with its share of challenges. Unfortunately, many PKI implementations require end-users to manage much of that complexity. Notably, users must initiate the key generation and request process. Once a user generates a private key and the CA issues a certificate, the user must configure all of their tools (operating system, web browser, mail client, etc.) to use the private key generated by the user and manage the list of trusted certificate issuers.

Sophisticated enterprises with dedicated engineering teams should be able to handle this complexity on behalf of the user community. Still, this complexity is difficult to manage even in highly controlled environments. This complexity is unmanageable for most small businesses and home users.

One way to address this user challenge is to have a designated administrator or security officer who assists users in generating their private keys and initializing their tokens. This approach is widespread in large enterprises and can also be feasible for smaller companies.

In high-security environments, users and administrators generate private keys on a hardware security module. This hardware requirement adds device driver installation and management issues to the other problems confronting users attempting to use PKI for authentication. Some platform vendors have implemented platform-level API (e.g., Microsoft CAPI). Still, support for this API is not universal, with some applications implementing proprietary or platform-neutral key storage systems that do not integrate with the host OS.

As with many IDM technologies, enterprises should observe the 80/20 rule. IDM professionals should ensure that critical or widespread user applications support your PKI implementation and accept alternative credentials for essential legacy applications.

Public key enablement of applications is hard:
We have discussed the difficulty of using PKI for authentication from the perspective of Authentication Subjects. Enabling applications to consume PKI credentials is even more challenging in some ways:

1. The list of trusted certificate issuers must be maintained and synchronized across all applications where the user may need to authenticate.
2. Applications must validate certificates, which requires the applications to access a public HTTP site or LDAP directory to obtain Certificate Revocation information.
3. A local user profile must be created in the application based on an identifier present in the certificate or entered by the user during a manual registration process.

There is no concept of provisioning or de-provisioning built into PKI by default, so applications must implement this capability through a separate integration with the Registration Authority (RA). Since it is common for users to authenticate with a site directly, CAs rarely offer this capability. Identity professionals should leverage existing directory technologies, such as Active Directory, to support user profiles for multiple applications.

For internally-facing enterprise applications, an IGA system may manage these aspects. Across enterprise boundaries or in a B2C context, this additional complexity makes PKI credentials difficult and expensive compared to other authentication technologies.

Certificate trust path discovery and validation are complex and existing implementations have inconsistent behavior:
In the previous section, we discussed applications needing to validate the certificates. This validation is complicated, even when administrators configure applications to use a static Trust List of known good issuers.[vii] To complicate things further, PKI supports a form of federation through cross-certification, discussed below in more detail. In this section, we will note that determining whether a trusted partner issued a certificate in a federated or cross-certified environment is very challenging.

Path Discovery and Validation (PDVal) is complex. Different vendors implement it inconsistently. One application may treat a certificate as valid, while another application may reject the same certificate, depending on the underlying certificate validation library. Some third-party solutions support consistent PDVal across products, but they must be implemented and integrated with each endpoint. This burden has made enterprises leery of implementing PKI on the server side.

Unique Considerations for Identity Practitioners

Ensure that PKI is the Right Fit for Your Requirements

Deployment of PKI involves several complexities and difficulties outlined in this document. However, PKI is a powerful tool that can offer strong authentication and support other use cases, such as email signing/encryption, that are impossible with other strong authentication credentials. When considering the deployment of PKI, ensure that the use cases you can support justify the added complexity for your environment and your users.

For TLS and link encryption, PKI may be the best or only choice, but that does not necessarily mean that you should implement your own local PKI. A third-party PKI service provider is an excellent alternative for most organizations.

The Importance of Planning

If you determine that an internally managed PKI is the correct choice for your organization, planning is critical for a successful PKI deployment. While the need for planning is not unique to PKI, the complexity of a PKI environment can make retroactive cleanup much more complex than careful up-front planning and deployment. As with any Identity Management technology, planning is critical to success.

IGA and PKI

Enterprises that leverage Identity Governance and Administration tools may need to expand their toolkit to accommodate PKI credentials. Existing IGA tools can manage accounts and privileges but may not track the PKI credentials associated with the managed accounts. It is important to recall that a PKI certificate and private key represent self-contained credentials that are still valid even if the underlying account has been deactivated or deleted. Unless the certificate is revoked or has expired, external applications may still accept a PKI credential as valid.

The challenges of managing non-human accounts such as machines, IoT devices, Bots, or other entities also apply to certificates issued to non-person entities. Refer to "Non-human Account Management (v3)" by Graham Williamson, André Koot, and Gloria Lee for information about unique issues related to these entities.[viii] The section below, 'Machine Identities and Certificate Management Systems,' discusses machine identities in more detail.

Many CAs include management capabilities to address these challenges. Some third-party (Certificate Management System) CMS products interact with multiple CA products to provide a single pane of glass for certificate management in a multi-vendor multi-CA environment. A later section discusses these products in more detail.

Lifecycle Management of PKI Certificates Compared to Other Credentials

Modern cryptographic algorithms ensure that private keys cannot be easily guessed. For example, a *classical (non-quantum) computer* would need about 300 trillion years to break a 2048-bit RSA key, while the same computer would require an average of five sextillion seven hundred eighty-three quintillion + years to guess a 128-bit ECC key.

However, the security of an overall system rarely depends exclusively on math.

The overall security of a PKI system includes several variables, including unreliable humans. CAs issue end entity certificates for a relatively short time, such as 90 days for public SSL certs or up to years for human subscriber certificates. CA certificates may be valid for as long as 20 years. This lifetime is much longer than a typical password or other credentials because the private key is never directly presented during authentication. The CA must store its private keys in a Hardware Security Module to ensure that an attacker cannot copy them.

Because CAs issue certificates with a fixed lifetime, key management can become a significant challenge. Enterprises should deploy a Certificate Management System (CMS) to monitor certificates and automate the renewal process or provide notification when a renewal is required. Most CA products include a rudimentary management console, but CMS products can offer a single pane of glass to manage multiple CAs from different vendors. CMS systems can also provide Service Desk support tools for assisting in smartcard registration and forgotten/locked PIN issues.

As with any other type of credential, a certificate may become invalid before it expires for various reasons. A user may leave the organization, change roles, or lose access to the private key. PKI provides for the revocation of public-key certificates in this case. The list of "no longer trusted" certificates is called the Certificate Revocation List (CRL). In every certificate, the CA publishes a URL where the CRL can be found. Alternative protocols, such as the Online Certificate Status Protocol (OCSP), offer other means of checking the validity of a certificate. Most web browsers have implemented proprietary revocation-checking techniques.

A third technology, called Server-based Certificate Validation Protocol (SCVP), has been developed and documented in a standard but has not been widely implemented. It is mentioned here for completeness, but most enterprises can disregard SCVP.

Because a certificate passes between the subject and a third party without involving the original issuer of the certificate, it is imperative that applications correctly validate certificates and check the revocation information.

Options for Identifiers in Public Key Certificates

The primary purpose of the certificate, as described above, is to link a public key with a user identifier. Of course, a user may have several identifiers for different use cases. Rather than issuing separate certificates for each user identifier, the PKI specification supports including multiple identifiers in a single certificate.

The primary user identifier in a certificate is the Subject Distinguished Name (Subject DN). The Subject DN must be structured like an LDAP Distinguished Name. Typically, there will be a "Base DN" shared by all certificates issued from a Certificate Authority and one or more "Relative DNs" which differentiate certificate subjects. Common relative DNs include "Organization" and "Organizational Unit." Finally, the certificate lists a subject's unique identifier. This identifier can take several forms, such as "Common Name," usually a user's Legal Name. In large PKI deployments, users with frequently seen names may have other identifiers embedded or appended to their names to distinguish between users with the same legal name. The common name is not the only possible identifier for a user in a Subject DN. Certificates can also use UID or email address to identify a certificate subject.

Because the Subject DN must mimic an LDAP Distinguished Name, it is very restrictive. For this reason, certificates often use an additional field instead. The "*Subject Alternative Name*" field is a much more flexible option to encode different user identifiers. It allows multiple names to be encoded and does not mandate a particular structure. Common uses for the subject alternative name field include:

- Email address to support S/MIME digital signature and encryption

- UPN to support smartcard login on the Windows platform
- Hostname to support TLS connections

The Subject Alternative Name does not impose any constraints on the type of identifiers that can be encoded. So, in addition to all of the previously listed identifiers, private communities of interest may insert identifiers that have strictly local meaning into this field. An example is the *Federal Agency Smart Credential Number (FASC-N),* which is part of the US Federal Government's Personal Identity Verification (PIV) standard.

Generally, Enterprises should use the subject alternative name for the user or machine identifiers. The Subject DN must be unique but should not contain multiple identifiers or non-standard ID types.

Machine Identities and Certificate Management Systems

While PKI has not seen widespread adoption as a credential for people,
it is dominant as a credential for machines, thanks to its use in TLS. TLS is not only used to provide secure access to web servers in end-user browsers; it is also widely used as a tunneling technology in machine-to-machine or site-to-site communication.

Virtualization and containerization technologies and the use of cloud providers have exploded in recent years. For this reason, the number of PKI-based machine identities is increasing exponentially. Managing and tracking the keys and associated certificates is becoming a significant challenge.

A Certificate Management System is an increasingly critical tool for enterprises to deploy to avoid service outages due to expired certificates, especially for enterprises with hybrid-cloud-based infrastructure or multi-vendor server environments.

Federated Authentication and PKI

We have already seen a critical difference between PKI and other credentials - a user can authenticate to an external application without involving the issuing authority in every transaction. This property can simplify authentication flows but places a greater burden on external applications since they validate the certificate themselves.

For an external application to consume certificates issued by your certificate authority, the application must trust your certificate authority. There are two basic ways for an application to trust an external certificate authority: explicit trust using certificate trust lists or implicit trust based on cross-certification.

Explicit trust is the most commonly used approach. In this model, applications explicitly managed trusted issuers within static Certificate Authority Trust Lists (CTL). The location of the trust lists will vary from product to product and system to system. A Java virtual machine, the Windows Operating System, and most web server software all maintain individual trust lists. Synchronizing them all in an Enterprise environment can be a very complex challenge. If you leverage an internal PKI, a CMS product can automate much of the management of disparate trust stores. Vendors typically configure standard software packages to trust the more prominent commercial Certificate Authorities. This is another good reason to acquire certificates from these sources.

Implicit trust relies on a technique known as cross-certification. In cross-certification, a CA will issue a certificate to another CA, typically operated by a different organization. From the perspective of an application validating certificates, the external CA appears to be another internal CA connected to the CA issuing the cross-certificate. The promise of cross-certification is that it dynamically allows applications to discover trust relationships between independently operated certificate authorities. In practice, however, the inconsistent behavior of PDVal implementations in software validating trust relationships between CA has prevented the promise of cross-certification from being fully realized. For most enterprises, cross-certification is not a helpful tool for federated authentication.

Finally, Identity Federation technologies can simplify the implementation of cross-domain trust by providing assertions across enterprise boundaries rather than relying directly on crossPDVal. The certificate can be validated within enterprise boundaries using relatively straightforward reliance on trust lists and local revocation publication. An SSO product can provide a federated token to external applications. This will address the interactive authentication use case but will not solve the challenges associated with other use cases that PKI can support, such as secure email encryption and signature or digital signatures for documents.

Conclusion

PKI is a powerful but complex tool for highly-secure authentication. It is likely already used within your environment for NPE or machine identities. Identity professionals should investigate the tools and processes used by individual programs to minimize redundancy of effort and cost.

Carefully weigh the benefits of the use cases within your environment before committing to deploying the technology to end users. If you choose to deploy PKI, avoid the temptation to introduce local or proprietary extensions, and stick to widely supported standards.

If an enterprise identity management environment is needlessly complex, it significantly complicates PKI deployment. Before deploying PKI, or any other complex authentication technology, ensure that identity management tools and practices are rationalized and streamlined within the enterprise environment.

If you introduce PKI for end-users, consider deploying a Certificate Management System to track the lifecycle of keys and certificates across your entire domain.

Author Bio

Robert Sherwood is the Principal Consultant at Credentive Security, a boutique consulting firm focused on Identity Strategy and Architecture.

Change Log

Date	Change
2021-09-30	V1 published
2022-12-15	V2 published; content significantly updated

[i] Note that credential assurance is distinct from identity assurance. Identity assurance measures how well you verified the identity of the account holder and how securely you connected the identity to the credential at the time of issuance. Credential assurance measures how confident you can be that the credential subject has maintained control over the credential, and that the credential has not been compromised.

[ii] Glazer, I., (2020) “Identifiers and Usernames”, *IDPro Body of Knowledge* 1(1). doi: https://doi.org/10.55621/idpro.16.

[iii] Williamson, G. & Koot, A. & Lee, G., (2022) “Non-human Account Management (v3)”, *IDPro Body of Knowledge* 1(7). doi: https://doi.org/10.55621/idpro.52

[iv] Technically, the sender generates a symmetric key, encrypts the message with the symmetric key, and then encrypts the symmetric key with the intended recipient’s public key.

[v] Technically, digital signing appends a 'hash' to the document that can be deciphered by the sender's public key - ensuring the sender's identity.

[vi] International Telecommunications Union – Technology (ITU-T), *X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*, October 2019, https://www.itu.int/rec/T-REC-X.509.

[vii] For example, see this article on how browsers handle revocation checks: https://www.ssl.com/blogs/how-do-browsers-handle-revoked-ssl-tls-certificates/.

[viii] Williamson, Koot, and Lee, “Non-Human Account Management (v3).”