PUBLIC KEY INFRASTRUCTURE (PKI) BASICS

Public Key Infrastructure (PKI) offers great potential for the way forward in information security. Its disciplines and methodologies have permeated a great number of commercial-off-the-shelf (COTS) security products. Protocols are becoming more robust as new and better algorithms are developed and adopted for general use or as key lengths for existing algorithms continue to increase. In the late 1960s, PKI cryptography concepts began to be referenced in classified documents in Great Britain. Prior to this time, PKI was held strictly in the purview of the military, secret services, and intelligence communities. By the mid-1970s, open literature and public discussion of the concepts of public key cryptography began to emerge in the United States and on a wider stage[i]. Since these times, PKI has become a central and ubiquitous part of enterprise security architectures, both large and small. PKI is now a generally accepted industry approach for providing universal trust and identity management capabilities in computer networks. By leveraging the X.509 standard[ii], PKI serves to provide interoperable vendor-agnostic security solutions to a wide range of commercial and government users world-wide.

Most standard protocols for secure email, Web access, virtual private networks, single sign-on, digital signatures, and data encryption use PKI for its cryptographic functions. Further, PKI services are standards-based and widely available in a range of existing and planned COTS hardware, software, and firmware products and applications. A representative set of modern security structures employing PKI concepts include:

- X.509 Certificate Standard (IETF/PKIX)[iii]
- Microsoft Public Key Infrastructure (MS PKI)
- Permission Management Infrastructure (OASIS/PMI)
- XML Key Management Specification (WWW Consortium/XKMS)
- Secure/Multi-purpose Internet Mail Extensions (S/MIME)
- Secure Socket Layer/Transport Layer Security (SSL/TLS)

This white paper examines PKI and the role it plays in protecting government and industry networks and facilities against unauthorized access.

What is PKI?

PKI is a set of standard computing services, commands, and application program interfaces (API) and a processing framework for tightly integrated digital certificate-based security controls. These controls are based on the concept of asymmetric key cryptography[iv]. In an asymmetric key encryption scheme, anyone can encrypt messages using the intended recipient's public key, but only the holder of the paired *private* key can decrypt the message. Likewise, PKI is used for authentication when the public key is used to verify that a holder of the paired private key digitally signed and sent an email message. In both cases, security and authentication assurance depends on the secrecy of the private key.

PKI is also a framework of generic, pervasive, application-independent security services. These services are standards-based and widely available blocks of program code built into a range of existing and planned network hardware, software, and firmware products and applications. Standard PKI services include, but are not limited to:

- Encryption/Decryption for Data Confidentiality
- Digital Signatures for Data Integrity, Authenticity, and Non-Repudiation
- Digital Identity Authentication (Person Entity[v] and Non-Person Entities[vi])

- VPN Secure Data Transmission and Tunneling (SSL/TLS)
- PKI Certificate Validation
- Key Generation, Publishing, Escrow, Exchange, and Recovery
- Certificate Revocation
- Trusted Time Stamping (TTS)
- Certificate Auto-Renewal at Expiration (MS and Some Commercial CAs, but not DOD PKI)

Access to PKI services occurs through use of standard APIs. This standards-based framework, interfaces, services, and command sets built into commercial-off-the-shelf (COTS) hardware and software offerings avoid the interoperability issues that in the past might have been evident with point-to-point, vendor-specific, or ad hoc proprietary application-based security solutions.

The PKI Concept of "Trust"

If cryptography is the foundation upon which PKI is built and X.509 Certificates the building blocks of the PKI framework, then "trust" is the mortar that holds it all together. Third-party trust[vii] is the general model for a PKI wherein a trusted Certification Authority (CA) signs and issues certificates (certs) to parties who rely on the digital identities asserted by the certs. In a typical PKI implementation, trust emanates from the Root CA, the top-level CA in the hierarchy of trust. Entities trust certs, public keys, and digital identities certified ("signed") by the Root CA or trusted CAs that are subordinate to the Root CA (Subord CA)[viii].

To prevent tampering or compromise and to ensure trust, the Root CA is maintained in a secure physical location in an *offline-only* mode. The Root CA is never hosted on an accessible network that would be subject to hacking, unauthorized access, and compromise[ix]. While Root CA certs are "self-signed" because there is no CA higher in the hierarchy to sign them, the Subord CA certs are signed with the private key of the Root CA certificate.

In a PKI hierarchy, the Root CA is the starting point for all trust. The hierarchy of CA's in a PKI domain is typically represented as an inverted tree structure with the root at the top, the branches extending downward, and the leaves at the bottom. In the inverted tree, the root represents a particular CA, commonly known as the Root CA, which acts as the "trust anchor" for the entire domain of PKI entities under it. Below the root CA are zero or more layers of subordinate CA's depending on the size, scope, and geographic expanse of the PKI hierarchy. And, the leaves correspond to non-CA PKI entities, often called end entities or simply end users[x].

Entities in the PKI hierarchy of trust hold the public key of the Root CA as their anchor of trust. Since the root CA is maintained completely offline, the public key is transported to the subordinate CA's in the hierarchy in a secure out-of-band fashion, that is, hand carried ("sneaker net") to the subordinate CA's on CDs or other removable media. (NOTE: All entities in the hierarchy except for the Root CA are on-network, so their public keys are transported electronically on the network to dependent entities). All subordinate CA certs are verified ("trusted") using the public key of their issuing CA, either the Root CA, or higher ranking Subord CA depending on the physical structure of the hierarchy. All end-entity certs, including those issued to person and non-person entities, are verified using the public key of their issuing CA.

The Purpose of PKI

PKI features, functions, and services establish the cryptographic framework for maintaining confidentiality, authentication, authorization, integrity, and non-repudiation when transacting online business digitally across private-, such as a corporate or agency networks, and/or public-

networks, such as the Internet. Cryptography concerns itself with the following five data processing objectives:

- **Confidentiality** – The data is encoded (encrypted) and cannot be deciphered (interpreted or understood) by anyone for whom it was not intended
- **Authentication** – The sender and recipient can confirm, with a very high degree of assurance, each other's digital identity, and the origin/destination of the information
- **Authorization** – The individual entity, group, or role has been assigned access privileges based on its digital identity to perform specific predetermined functions in the network environment (e.g., Read, Write, Modify, Delete, Create)
- **Integrity** – The information cannot be altered in storage or transit between sender and intended recipient without the alteration being detected or detectable
- **Non-repudiation** – The creator/sender who digitally signed the information cannot deny at a later stage the fact of or his or her intentions in the creation or transmission of the information

PKI is a methodology to bind end-entity digital identities to public and private asymmetric key pairs in the form of standards-based certificates. Asymmetric keys may be 1024, 2048, 4096, or more bits in length[xi]. Bit length determines the strength of the keys to resist cracking the cipher by brute force (e.g., systematically trying each and every value in the key space to break the cipher) or other forms of attack.

Certs may be version 1, 2, or 3, but both industry and Federal Government require use of the X.509 v3 standard.

- X.509 v1, introduced in 1988, did not support CA renewal.
- X.509 v2, improved in 1993, by adding two unique ID fields.
- X.509 v3, improved in 1996, added new fields and extensions.

Maintaining the trust dynamic in PKI requires air-tight procedural security and automated control procedures maintained under the framework of strong Certificate Policy (CP) and Certification Practice Statements (CPS)[xii]. Strong procedural and personnel roles are instituted to assess and confirm in person the identities of entities to be bound to PKI certificates and to control the issuance of PKI certificates.

- A CP is used to express policy only, defining *what* is performed within the PKI environment, not *how* it is performed.
- A CPS is a blueprint of the PKI environment, defining *how* the CP is enforced.

X.509 v3 certificates contain references to both CP statements in the form of Object Identifiers (OIDs) and the issuing CA's CPS.

RFC 3647[xiii], the Internet X.509 PKI Certificate Policy and Certification Practices Framework, recommends the same format for both CP and CPS even though they differ in focus and the audience served, (i.e., PKI stakeholders and Certification Authority, respectively). Depending on specific use cases, PKI may be engineered to support several different Levels of Assurance[xiv]:

- **Class 2 PKI (Basic):** Commercial; Not planned for Civilian Government or DOD Infrastructures
- **Class 3 PKI (Medium)**: Secure But Unclassified (SBU); Civilian Government and DOD business and operational PKI systems fielded using COTS products
- **Class 4 PKI (High):** National Security Systems (NSS) and Command & Control (CC) environments; Civilian Government and DOD classified PKI systems fielded using Class 4 Tokens

Certificate Policy (CP)

The PKI trust model assumes the issuing CA verifies/confirms identity of the certificate entity. The mechanisms the CA must use at a minimum to manage identity of entities are published in the Certificate Policy (CP).

- CP is enforced by PKI-enabled applications.
- CP sets requirements for certificate usage.
- CP often describes the protection level for certificate's private key, such as:
  - Private key stored in hardware token (more secure),
  - Private key stored in local filesystem (less secure), or
  - Private key escrowed/not escrowed policy.

The Extended Key Usage attribute in certificates contains a list of Object Identifiers (OID) that specify authorized usage of the keys hosted in certificates. OIDs exist in an inverted tree defined by ITU-T X.208 standard, and they reside inside each certificate.

Certification Practice Statement (CPS)

Certification Practice Statements (CPS) describe how the operating procedures and practices the CA uses work to uphold its security policies and CPs. It describes:

- How CA operations are secured,
- How private keys are protected,
- How CP entity validation procedures are enforced, and
- How certificate lifecycle services such as issuance, management, escrow, revocation, and renewal are provided.

In the context of a PKI, certification is the act of binding a subject name (the name of the Cert owner) with a public key. The binding occurs in the form of a signed data structure referred to as a public key certificate.

PKI Roles

A Certification Authority is responsible for issuing the public key certificates which are digitally signed with the private key of the issuing CA. Because the issuing CA digitally signs certificates, the combination of unique identities and digital keys in the certificate is self-protected for data integrity against inadvertent or malicious tampering and change after issuance. Manipulation of the certificate is readily detectable.

SAMPLE FIELDS IN A PUBLIC KEY CERTIFICATE
Version
Serial Number
Signature Algorithm
Issuer
Valid From
Valid To
Subject
Private Key (Hidden)
Public Key
Certificate Policies (OIDs)
Extensions (Type, Criticality, Value)
Issuer's Digital Signature (aka Thumbprint)

A Registration Authority (RA) may be created to off-load certain control functions from the Certification Authority to enhance scalability and decrease operational costs and inefficiencies. RAs may be distributed geographically (close to PKI subscribers) to establish and confirm the identities of individuals, who may be widely dispersed from the centralized CA. RAs confirm identities through a combination of physical presence and review of official identification documents, such as a birth certificate, social security card, driver's license, or passport.

PKI Use of Cryptographic Functions

There are three main cryptographic functions that serve as the underpinnings of real-world practical uses of cryptography in PKI for security applications:

- Symmetric Cryptography
- Asymmetric Cryptography
- Cryptographic Hash Functions

Symmetric Cryptography

Early methods of confidential communications employed symmetric secret key formulas (or ciphers)[xv] to transform plaintext to uninterpretable nonsensical data ("encrypt") and back to plaintext ("decrypt") for the intended reader. Alternative names for a secret key include "shared key" or "shared secret".

The secret keys were described as symmetric because they were identical; the same cipher was used on both sides of the transaction. It was used to transform data into unintelligible secret text by the sender and then back to intelligible text by the recipients of the encrypted message. Under symmetric key cryptography[xvi], the cipher had to be shared on a case-by-case basis between the transmitter of the secured data and all intended recipients who would need to decrypt the data. Symmetric keys could not be maintained in a publicly accessible online repository or directory for this would defeat the purpose of a secret key[xvii].

Examples of contemporary symmetric cipher types and key lengths include:

- 3DES: 128 and 168 Bits[xviii]
- AES: 128, 192, and 256 Bits[xix]

3DES and AES ciphers are extremely secure. With a key length of n bits, there are 2^n possible keys for use in the cipher (e.g., $2^3 = 2 \times 2 \times 2 = 8$). So, for AES-128, there are 2^{128} possible individual keys.

Some strengths to consider when using symmetric key cryptography:

- Result in a small implementation size for the resulting ciphertext.,
- Ability to handle bulk data encryption quite easily and efficiently, and
- Encryption/decryption speeds 1000X (or more) faster than asymmetric ciphers.

Symmetric ciphers require secret key exchange, one-to-one or one-to-many depending on the number of intended message recipients. The problem is how to distribute the secret key in a secure way against eavesdropping or sniffing attacks. It presents difficulties in initiating secure communication between previously unknown/untrusted parties.

Additional shortcomings with the use of symmetric key cryptography include:

- The encryption key and decryption key are identical or very easily derived from the other, rendering them less secure than public key cryptography methods.
- It does not effectively support authentication or non-repudiation.
- It is difficult to scale. N users require N*(N-1)/2 unique keys to communicate secretly[xx], such that 20 users wishing to communicate secretly would require 190 unique keys ((20x19)/2).

Asymmetric Cryptography

In the mid-1970s, Whitfield Diffie and Martin Hellman steered the public discussion of cryptography in the direction of asymmetric ciphers. The keys were described as asymmetric because the key used for encryption is different from the one used for decryption. The public and private keys in an asymmetric key pair are substantially different and the relationship computationally complex. This means that knowledge of one does not readily allow calculation or reverse engineering to derive the other, even by an adversary with substantial computing power at their disposal and specific knowledge of the cipher algorithm. Neither key can both encrypt and decrypt a message, and while the two keys are related mathematically, neither key can feasibly be derived from the other.

Because it is computationally infeasible to derive one key from the other, one of the keys can be revealed publicly without compromising communications security provided that the other non-public key remains secret and private. Theoretically, the private key can be derived because the keys in an asymmetric cipher are related, as one key must decrypt what the other encrypts and vice-versa. However, with current technology limitations and key strengths, the amount of time, memory, or computing power necessary to derive one key from the other is prohibitively large, making this challenge insurmountable for the next several years at least [xxi].

Examples of contemporary asymmetric cipher types and key lengths include:

- DSA 1024 Bits[xxii]
- RSA 1024, 2048, 3072, or 4096 Bits[xxiii]
- ECC 160, 224, 256, 384[xxiv]

DSA and RSA ciphers use the same model as their symmetric cousins, but the number of bits in an asymmetric cypher results in a key that is orders of magnitude larger than the keys associated with symmetric key cryptography. (Current federal standards require minimum use of RSA-2048.). ECC, on the other hand, employs extremely complicated calculations, but with smaller key sizes. As a result, current federal standards require minimum use of ECC 224 or 256.

Generally, computations involved in asymmetric cryptography are too slow for bulk data encryption due to the extreme complexity of the cipher algorithms and ultra-high key strengths, so in practice, asymmetric encryption involves a two-step process.

- Bulk data is first encrypted using a randomly generated symmetric key, also known as a “session key”. There is a different ciphertext derived during each symmetric encryption event.
- The very small symmetric key is then encrypted using the asymmetric public key(s) of the intended recipient(s) of the data.

Decryption involves a two-step process in reverse. The recipient(s) first decrypts the symmetric key, using his or her private key, and the symmetric key is then used to decrypt the actual data.

Even when the total amount of data to be encrypted is very small, convention holds that the two-step processes are used rather than direct data encryption/decryption using the asymmetric public/private-key pair. This convention keeps encryption/decryption processing clear, simple, and unambiguous, and eliminates uncertainty as to whether output of a private-key decryption operation is data or a symmetric key.

There are also several material shortcomings with the use of asymmetric key cryptography:

- A comparatively large implementation size,

- Significantly slower than symmetric cryptography when processing a large body of data because encryption may get slower as key lengths increase,
- Encryption/decryption speeds 1000X (or more) slower than symmetric ciphers, and
- The potential with some public key algorithms for ciphertext twice the size of the corresponding plaintext.

Elliptic Curve Cryptography (ECC), an emerging addition to the asymmetric protocol suite with smaller key sizes and higher cipher strength, is very appealing for devices with limited storage or processing power or for situations where increasing key lengths of RSA, DSA, and other traditional asymmetric methods might become prohibitive from a processing performance standpoint[xxvi].

ASYMMETRIC CRYPTOGRAPHY KEY LENGTH EQUIVALENCY RSA/DSA TO ECC	
RSA and DSA Key Size (bits)	ECC Key Size (bits)
1024	160
2048	224
3072	256
7680	384

NIST Approved Key Lengths (Source: SP 800-56)[xxv]

Cryptographic Hash Function

A hash function is a mathematical procedure or algorithm for compressing or abstracting data of any arbitrary size to data of a fixed size. A hash function takes a data string of any length as input and produces a fixed length string that acts as a kind of signature for the data provided. Such compressed data is generally referred to as a message digest, hash value, or simply a hash. The data compression process uses a one-way transformation function that makes it impossible to determine the original data from the resulting hash value.

Examples of hash functions include:

- Message Digest 5 (MD5) hash algorithm produces a 128-bit (or 16 alphanumeric character) message digest.
- Secure Hashing Algorithm 1 (SHA-1) produces a 160-bit message digest.
- SHA-256, SHA-384, and SHA-512 (SHA family collectively known as SHA-2) produce message digests of 256, 384, and 512 bits, respectively.

Hash functions are designed so that any change to the input data, no matter how slight, has a very high probability of significantly affecting the computed hash value. The same hash value is always produced by a given hash algorithm from the same input data, and the probability of two different pieces of input data producing the same hash value is extremely small. This quality means that hashing, used in conjunction with cryptographic algorithms for data encryption, provides easy verification that data hashed and signed by a cryptographic algorithm has not been changed after the digital signature was applied. This feature is referred to as data integrity.

Even though it is very difficult to reverse engineer a hash to derive the original data, the quality that makes cryptographic hashing so valued, the production of the same hash value by a given algorithm for the same input data, gives a slight opening to the determined brute force attacker. As an additional layer of security, some have appended a randomly-generated variable to the input data ahead of hashing ("salting the hash"), making each hash result unique, which in turn makes it virtually impossible to successfully reverse engineer the hash value to determine the original input data[xxvii].

Note: Hash functions are neither public or symmetric key algorithms. They are included in this discussion because digital signature algorithms are always used in conjunction with hash algorithms to provide the services of signing/verification and data integrity.

Practical Uses of Hash Functions and Cryptography

The three most common uses of hash functions and cryptography are secure login to enterprise networks, digital signing, and PKI certificate validation.

Secure Login to Enterprise Networks: Passwords have historically been used to confirm the identity of end users requesting network login and access to network resources, but passwords tend to be weak and cannot resist brute force attacks or eavesdropping attacks. Contemporary network operating systems (Windows, Macintosh, Unix) have integrated cryptographic capabilities and Kerberos to facilitate secure end user authentication to networks and network-hosted resources[xxviii]. The Kerberos protocol is used by operating systems to negotiate shared session key and service request transmissions between clients requesting authentication and servers hosting required services. Request and reply transactions provide mutual authentication between client and server resources.

Communications for mutual authentication of client and server services for smartcard (PIV or CAC) login include the following:

- Client initiates, signs. and sends the initial request for authentication to the login server. Use of public-key authentication is indicated by including a special pre-authenticator attribute in the request identifying type of credential used (e.g., password, PKI smartcard).
- Server tests the client's request against its authentication policy, trusted digital identities, and Certification Authorities (CA).
- If the request passes the server's verification tests, the server replies with a response encrypted with the client's public key and signed with the server's private key. The server's public key is returned to the client in the encrypted package for use in validating the server's digital signature, and a shared random number session key is returned for subsequent encryption of challenge-response transactions.
- Client validates the server's signature, obtains the encryption key, decrypts the server's reply, and proceeds to complete the mutual authentication and login activity using the shared session key for subsequent transaction activity.

The use of asymmetric cryptography in the form of PIV or CAC certificates, which are X.509 certificates, establishes initial authentication and secure communication using randomly generated session keys. Session keys are different for each request to authenticate, eliminating the hacker's ability to use brute force or eavesdropping attack methods. Because they are timestamped, they also tend to thwart re-play attempts.

One additional advantage of employing Kerberos in authentication sequences is that the client exposes personal cryptographic material from the PIV or CAC only once during the initial login request scenario. Requests for access to network resources after the initial login (aka "single sign-on" requests) are driven by independently derived random-value session keys, divorcing any subsequent accesses to resources in the network from the methods and keying material that were used in the initial authentication.

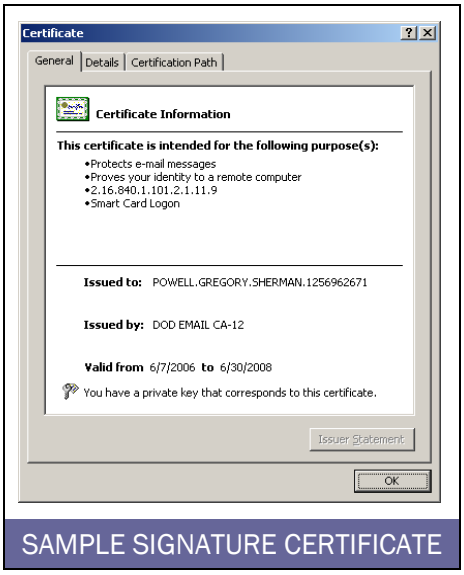
Digital Signature Function: A digital signature is a block of fixed-size data ("message digest") encrypted and affixed to a document by the sender, making it computationally infeasible for any

other entity to create, forge, or imitate. A digital signature may be inserted directly into the data or could be in a file associated with, but separate from, the actual data.

A digital signature relies on the concept of public-private key ciphers working in conjunction with a hash function to create the "signature". There must be a private key uniquely and explicitly bound to the digital identity of the signer and known only to the signer. There must also be a public key available to and known by a wide group of entities that is relied on by them to verify the authenticity of the digital signature. As the party relying on the public key, you can verify a signature ("encrypted hash value") created with the private key of the signer if you know the public key of the signer. This provides:

- **Data Origin Authentication** – authenticates the identity of the data originator
- **Data Integrity** – guarantees that the contents of the file or email are the same as when the data was signed
- **Non-Repudiation** – inextricably binds the identity of the signer and the origin so the signer cannot at a later time deny authorship, ownership, or origination of the e-communication, making it legally enforceable in a court of law[xxix]

The signatureAlgorithm field in a PKI certificate contains the identifiers for the public key algorithm and hash function that are used to sign data and used by the CA to sign the certificate. For security purposes, the PKI certificates are signed to maintain their own data integrity.



SAMPLE SIGNATURE CERTIFICATE

Digital signing a *plaintext* data, such as an email or a form, is a two-step cryptographic process. First, the message to be transmitted is passed through a hash algorithm (for example, SHA-1) to obtain a fixed-size hash value of the message (for example, 160 bits). Then, the hash value is encrypted using the private key of the sender, producing the digital signature. Both the encryption and signature certs are transmitted with an outgoing signed message for use by the relying party.

Digital signature verification is also a two-step cryptographic process. First, the message to be verified is hashed to obtain an independent determination of the fixed size hash value. Then, the encrypted digital signature hash value is decrypted using the public key of the signer to produce a copy of the original fixed size hash value. If the independently determined hash value and the transmitted digital signature hash value match based on the application of public key decryption, the received message text is confirmed as unaltered since the digital signature was applied, and the digital signature itself is confirmed as created by and only by the trusted entity fostering non-repudiation. Otherwise, signature verification fails.

Digital signing of *encrypted* data is a five-step cryptographic process:

1. The bulk data is encrypted using a randomly generated symmetric key, also known as a session key. Different ciphertext is derived during each encryption event with the random session keys even with identical data.
2. The symmetric key is then encrypted using the asymmetric public key(s) of the intended recipient(s) of the data.
3. The encrypted session key is appended to the ciphertext message.

4. The ciphertext/session key combination is hashed to obtain a fixed-size hash value.
5. The hash value is encrypted (for example, RSA) using the private key of the sender to produce the digital signature.

PKI Certificate Validation: To ensure validity of the PKI certificates forming the base of a cryptographic security environment, the whole certificate is "signed" by the issuing CA. The signature of the issuing CA is appended to the end of the certificate in a data field referred to as the "thumbprint", forming the basis of trust.

Basic PKI certificate validation is a five-step cryptographic process:

1. The certificate to be verified undergoes a hash algorithm to obtain an independent determination of the hash value of the certificate.
2. The digital signature "thumbprint" is decrypted using the public key of the issuing CA to produce the original hash value.
    a. If the independently determined hash value and the embedded digital signature hash value match based on the application of public key decryption to the digital signature, the digital signature and certificate are confirmed as created by the trusted CA (data origin authentication) and content unaltered since the certificate was created and signed (data integrity).
    b. If they do not match, certificate validation fails.
3. The certificate is confirmed to be within the "valid from/valid until" expiration period.
4. The certificate revocation status is checked employing the Certificate Revocation List (CRL)[xxx] or a variation known as Online Certificate Status Protocol (OCSP)[xxxi].

References

[i] Carlisle Adams and Steve Lloyd, Understanding PKI: Concepts, Standards, and Deployment Considerations (Addison-Wesley) 2003
[ii] In cryptography, X.509 (or PKIX "X.509-based PKI) is a standard that defines the format of public key certificates, the data attributes hosted therein and their uses, Wikipedia, Definition of X.509
[iii] X.509 is defined by the International Telecommunications Union's Standardization sector (ITU-T) and is based on ASN.1, another ITU-T standard
[iv] Asymmetric ("Public Key") Cryptography is any cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys which are secret and known only to the owner
[v] "Person Entity", Wikipedia, "In the computing environment 'person entities' include the total set of natural human beings operating on or within the network domains of an enterprise"
[vi] "Non-Person Entity", Wikipedia, "In the computing arena 'non-person entities' include the total set of digital identities on the network (excluding 'human' actors) such as hardware devices, software applications, and other information artifacts that constitute the computing environment"
[vii] In cryptography, a trusted third party (TTP) is an entity, e.g., Certification Authority (CA), which facilitates trusted electronic transactions between two or more relying party entities all of which "Trusts" the CA and the digital identity certificates it issues
[viii] Subordinate CAs in "Microsoft-speak" are referred to interchangeably as "Intermediate CAs"
[ix] Notable Network Breaches: Yahoo! 2014 breach of 500 million user accounts; the Justice Department, Internal Revenue Service, US Navy, and Snapchat 2016 breaches; Office of Personnel Management's (OPM) 2015 two separate breaches netting 22 million personnel accounts
[x] Adams and Lloyd, Understanding PKI
[xi] A "bit" (short for binary digit) is the smallest unit of data in a computer's memory. A bit has a single binary value, either 0 or 1. The value of a bit is usually stored as either above or below a designated level of electrical charge in a single capacitor within a computer's memory device, Whatis.com
[xii] Chokhani, S., and W. Ford. "Internet X.509 Public Key Infrastructure: Certificate Policy and Certification Practices Framework." Internet Request for Comments (RFC) 2527, March 1999
[xiii] RFC 3647, "Internet X.509 PKI Certificate Policy and Certification Practices Framework", https://www.ietf.org/rfc/rfc3647.txt
[xiv] "PKI Levels of Assurance", Federal Public Key Infrastructure (FPKI), ID Management.gov
[xv] A "cipher" in cryptography is a formula (or algorithm) for performing encryption or decryption – a series of well-defined steps that can be followed as a procedure to encode data to keep it secret or decode it in order to recover its original meaning, Wikipedia
[xvi] Symmetric ("Secret Key") Cryptography is any cryptographic system that uses identical encryption and decryption keys or the keys are easily derived from one another
[xvii] Carlisle Adams and Steve Lloyd, Understanding PKI: Concepts, Standards, and Deployment Considerations (Addison-Wesley) 2003
[xviii] "3DES" in cryptography, officially the "Triple Data Encryption Algorithm", is a symmetric key block cipher which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block, Open Web Application Security Project (OWASP)
[xix] "AES" in cryptography, the "Advanced Encryption Standard", is the successor of DES as a Federal Information Processing Standard symmetric encryption algorithm for US federal organizations, OWASP
[xx] Adams and Lloyd, Understanding PKI
[xxi] Adams and Lloyd, Understanding PKI
[xxii] "DSA" in cryptography, the "Digital Signature Algorithm", is a Federal Information Processing Standard asymmetric key cipher for digital signaturing, Wikipedia
[xxiii] "RSA" in cryptography, the "Rivest Shamir Adleman" algorithm, is a Federal Information Processing Standard asymmetric key cipher for data encryption and secure data transmission, Wikipedia
[xxiv] "ECC" in cryptography, the "Elliptic Curve Cryptography" algorithm, is an approach to public key cryptography based on the algebraic structure of elliptic curves; ECC requires smaller keys for comparable cipher strength to their non-ECC counterparts, e.g., RSA, DSA
[xxv] NIST SP 800-56, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography", http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf
[xxvi] "ECC", GlobalSign GMO Internet Group, "ECC 101: What is ECC and Why Would I Want to Use It", https://www.globalsign.com/en/blog/elliptic-curve-cryptography/
[xxvii] "Salting Hashes", AddedBytes, "Why You Should Always Salt Your Hashes", https://www.addedbytes.com
[xxviii] "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", Request for Comments (RFC) 4556, June 2006
[xxix] "Legal Enforceability of Digital Signatures", Wikipedia, Electronic Signatures in Global and National Commerce Act (ESIGN, Pub.L. 106–229, 114 Stat. 464, enacted June 30, 2000, 15 U.S.C. ch. 96) is a United States federal law passed by the U.S. Congress to facilitate the use of electronic records and electronic signatures in interstate and foreign commerce by ensuring the validity and legal effect of contracts entered into electronically

[xxx] Adams and Lloyd, Understanding PKI, "Certificate Revocation List 'CRL' is a signed data structure periodically issued by the Certification Authority 'CA' that contains a list of the revoked PKI certificates for that specific CA"
[xxxi] Adams and Lloyd, Understanding PKI, "Online Certificate Status Protocol 'OCSP' is a simple request-response protocol that obtains revocation status 'good' 'revoked' or 'unknown' of PKI certificates from a trusted entity referred to as an OCSP responder"