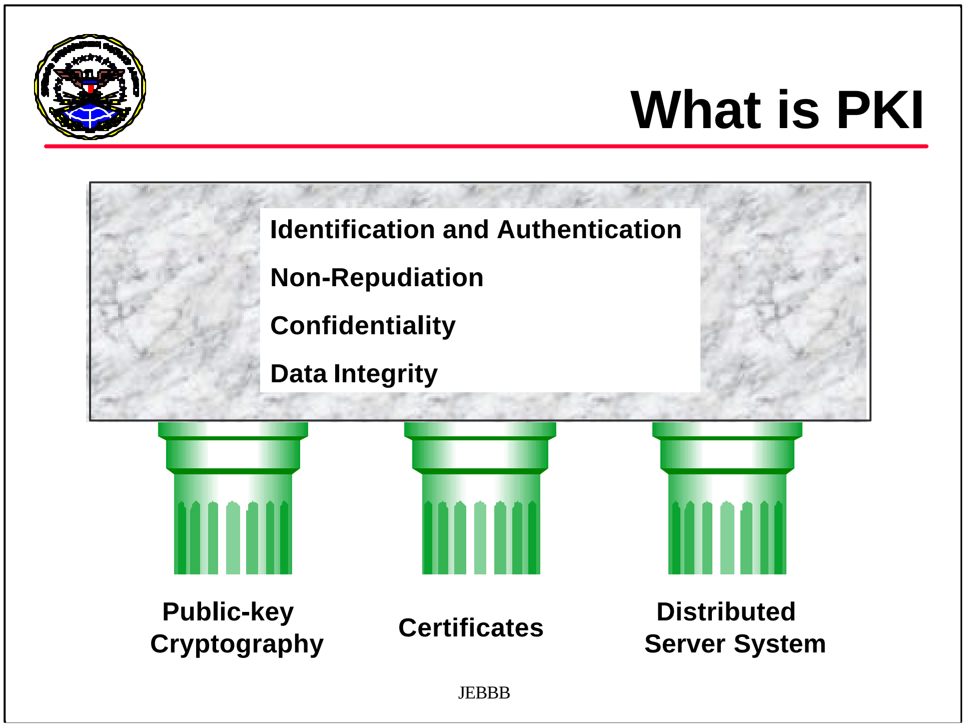


What is PKI

Identification and Authentication

Non-Repudiation

Confidentiality

Data Integrity

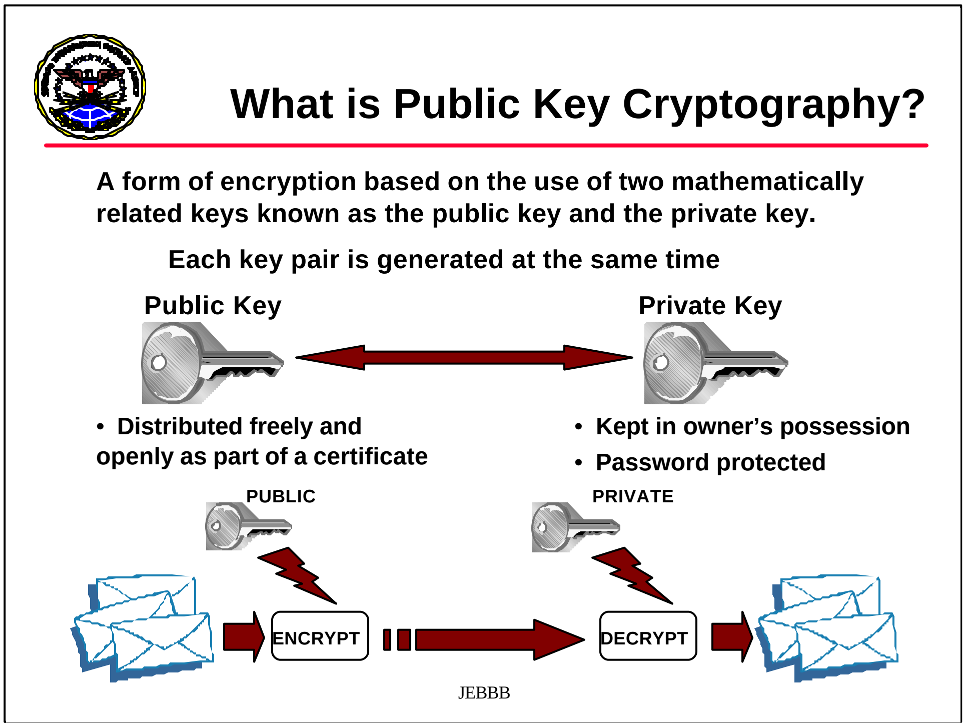
Public-key Cryptography

Certificates

Distributed Server System

JEBBB

The use of public-key cryptography and X.509 certificates in a distributed server system to establish secure domains and trusted relationships. This supports these assurance services that are necessary for the conduct of electronic business.



What is Public Key Cryptography?

A form of encryption based on the use of two mathematically related keys known as the public key and the private key.

Each key pair is generated at the same time

Public Key

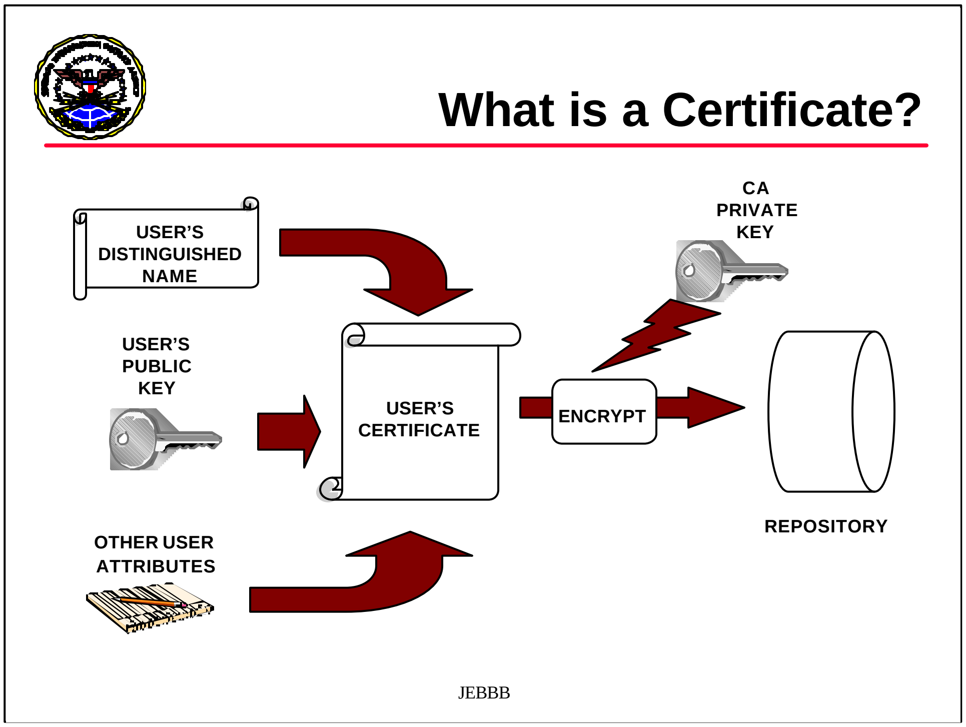
Private Key

- Distributed freely and openly as part of a certificate

- Kept in owner’s possession
- Password protected

JEBBB

<u>Public-key Cryptography </u>- Generate key pairs - one public, the other private (asymmetric). Owner of key pair uses private key to en/decrypt. Others uses public key to en/decrypt.



What is a Certificate?

USER'S DISTINGUISHED NAME

USER'S PUBLIC KEY

OTHER USER ATTRIBUTES

USER'S CERTIFICATE

CA PRIVATE KEY

ENCRYPT

REPOSITORY

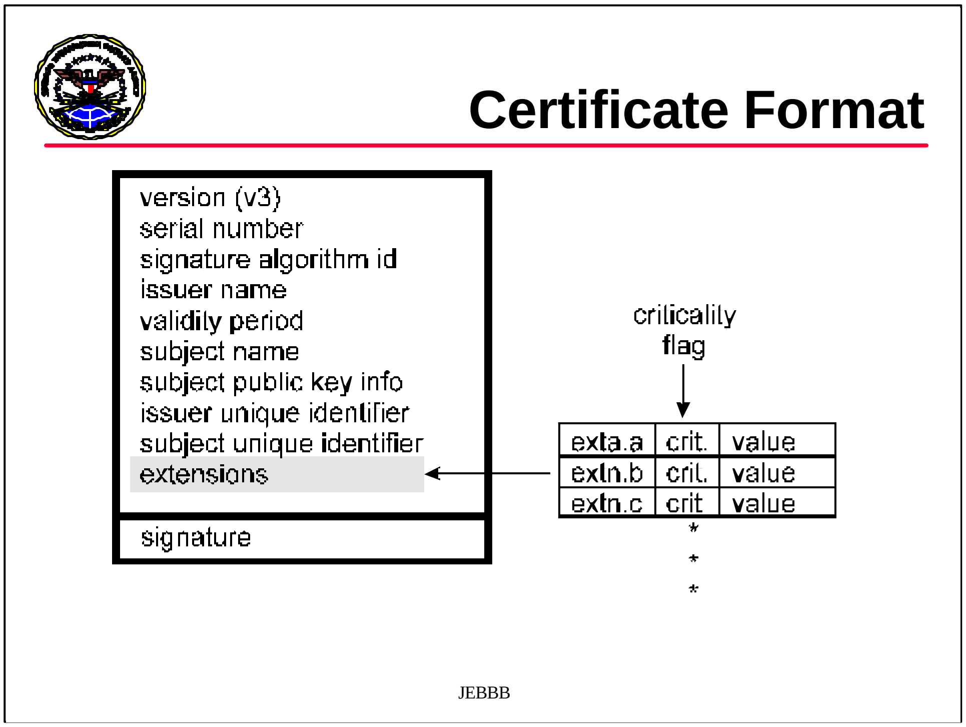
JEBBB

The public keys of a user, together with some other information rendered unforgeable by encipherment with the private key of the CA that issued it. [Ref. X.509] (AKA user certificate; public key certificate)

Its purpose is to reliably associate a user and their public key. It includes the user's distinguished name, which may include additional attributes that uniquely identify the user. The certificate specifies a validity period, and the specific operations for which the public key is to be used. Users can trust that that the public key recovered belongs to the entity specified by the distinguished name.

Public key is susceptible to spoofing and "man-in-middle" attacks – mismatch key and owner. Certificates binding user and public key mitigates that.

A repository is an on-line, publicly accessible system for storing and retrieving certificates and other information relevant to certificates, such as revocation information. In effect, putting a certificate in a repository publishes it.



Certificate Format

version (v3)
serial number
signature algorithm id
issuer name
validity period
subject name
subject public key info
issuer unique identifier
subject unique identifier
extensions

signature

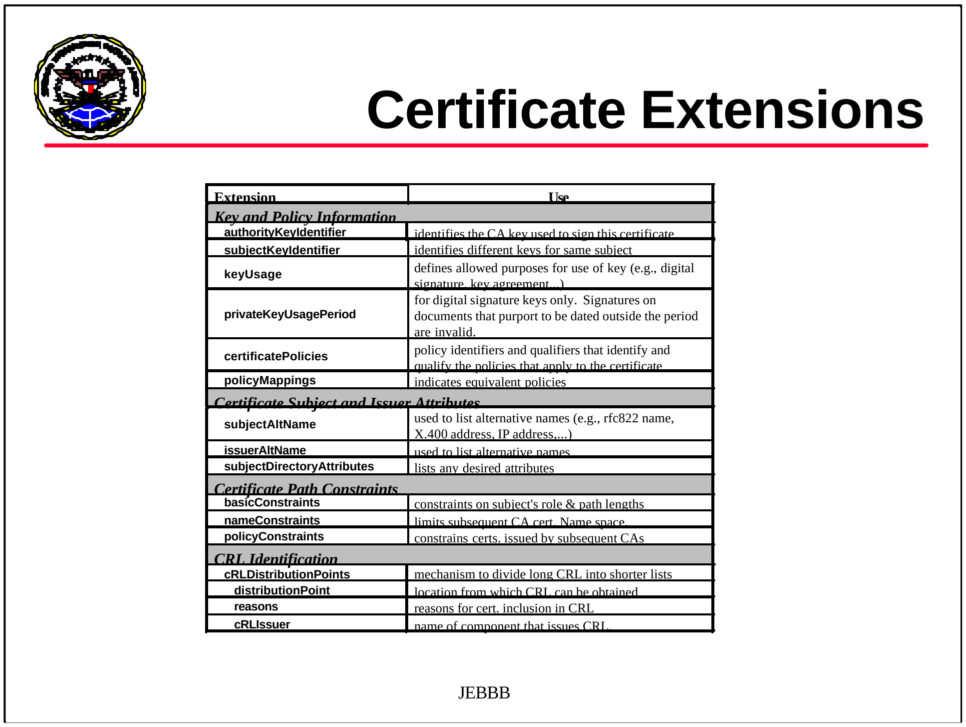
criticality
flag

exta.a	crit.	value
extn.b	crit.	value
extn.c	crit	value

*
*
*

JEBBB

version	version number; an integer, value is “2” for version 3	
serial number	unique identifier for each certificate generated by issuer; integer	
signature algorithm ID	algorithm identifier	algorithm used to sign certificate
	parameters	should not be used
issuer name	name of issuer (X.500 “distinguished name” that uniquely identifies a directory object).	
validity period	notBefore	Time
	notAfter	Time
subject name	name of subject (X.500 “distinguished name”)	
subject public key info	algorithm identifier	subject’s signature algorithm
	parameters	parameters applicable to subj. pub. key
	public key	subject’s public key
issuer unique identifier	(optional) contains additional information about the subject; certificate must be version 2 or higher - not used by the Federal PKI.	
subject unique identifier	(optional) contains additional information about the issuer; certificate must be version 2 or higher - not used by the Federal PKI.	
extensions	(optional)	
issuer’s signature	algorithm identifier	algorithm used for this signature
	parameters	should not be used
	ENCRYPTED (certificate hash)	



Certificate Extensions

Extension	Use
Key and Policy Information	
authorityKeyIdentifier	identifies the CA key used to sign this certificate
subjectKeyIdentifier	identifies different keys for same subject
keyUsage	defines allowed purposes for use of key (e.g., digital signature, key agreement...)
privateKeyUsagePeriod	for digital signature keys only. Signatures on documents that purport to be dated outside the period are invalid.
certificatePolicies	policy identifiers and qualifiers that identify and qualify the policies that apply to the certificate
policyMappings	indicates equivalent policies
Certificate Subject and Issuer Attributes	
subjectAltName	used to list alternative names (e.g., rfc822 name, X.400 address, IP address,...)
issuerAltName	used to list alternative names
subjectDirectoryAttributes	lists any desired attributes
Certificate Path Constraints	
basicConstraints	constraints on subject's role & path lengths
nameConstraints	limits subsequent CA cert. Name space.
policyConstraints	constrains certs. issued by subsequent CAs
CRL Identification	
cRLDistributionPoints	mechanism to divide long CRL into shorter lists
distributionPoint	location from which CRL can be obtained
reasons	reasons for cert. inclusion in CRL
cRLIssuer	name of component that issues CRL.

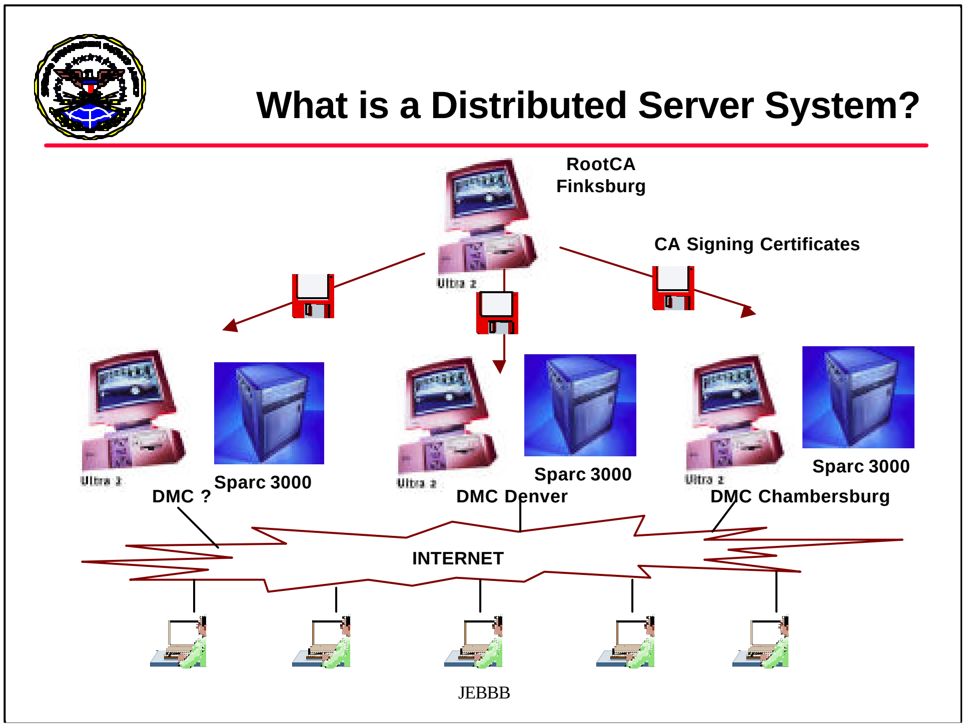
JEBBB

Key and Policy Information - These extensions provide information to identify a particular public key and certificate. They may restrict the purposes for which a key may be used.

Certificate Subject and Issuer Attributes - These extensions provide additional information about other names and attributes of the subject and issuer.

Certificate Path Constraints - These extensions apply to restrictions place on validation of certificate paths. (An ordered sequence of certificates in a hierarchy which, together with the public key of the initial certificate, can be walked back to the root certificate that established trust.)

CRL Identification - These extensions include information in a certificate about where to obtain the CRL that applies to a certificate.



What is a Distributed Server System?

RootCA
Finksburg

CA Signing Certificates

Ultra 2

Ultra 2

Sparc 3000

DMC ?

Ultra 2

Sparc 3000

DMC Denver

Ultra 2

Sparc 3000

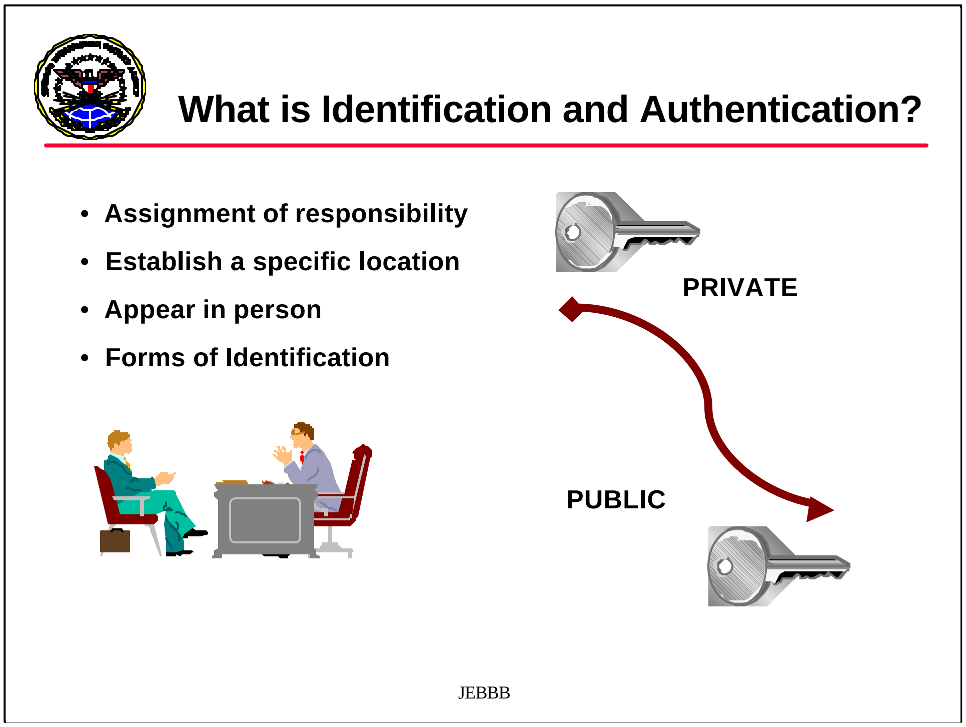
DMC Chambersburg

INTERNET

JEBBB

<u>Distributed Server System</u> - An architecture of servers (e.g., directory, messaging, certificate, etc.) for distributing applications transparently across networks of heterogeneous computers.

Server: A provider of resources (e.g. name server).



What is Identification and Authentication?

- Assignment of responsibility
- Establish a specific location
- Appear in person
- Forms of Identification

PRIVATE

PUBLIC

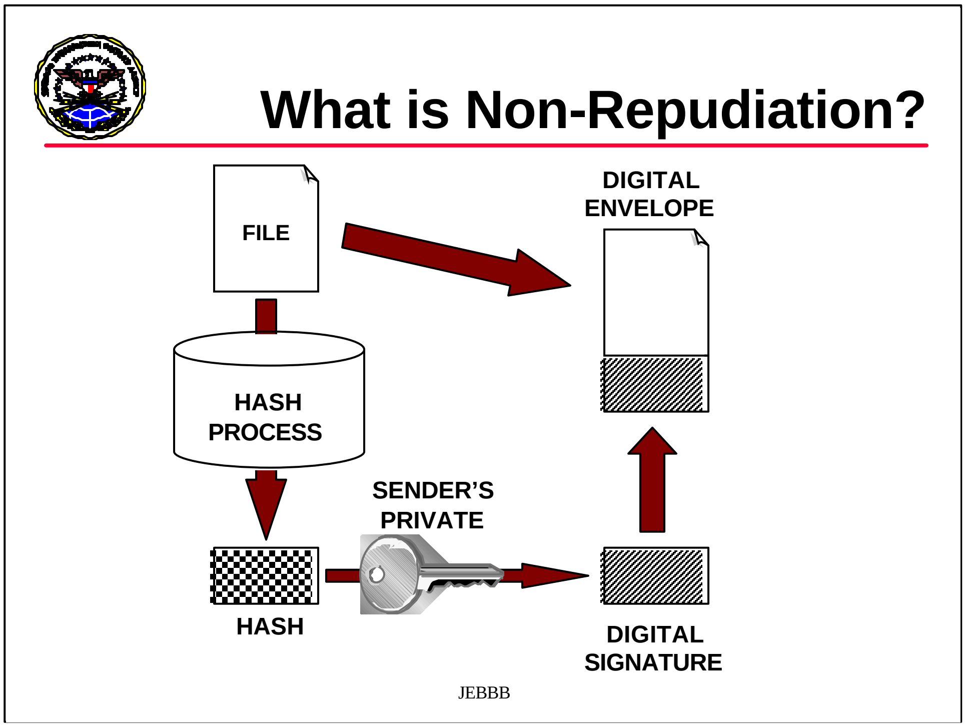
JEBBB

<u>Identification & Authentication</u> - The sender is who they claim to be. Also, <u>Authorization</u> - what they can do. And <u>Access Control</u> - with what system resources

An established procedure that requires those applying for certificates to appear in person and identify themselves to the Local Registration Authority.

The CA by issuing a certificate guarantees that you and your public key match. You know the CA issued the certificate because the CA’s public key decrypts it.

Authentication is done because you need the sender’s public key to decrypt the data he sent you.

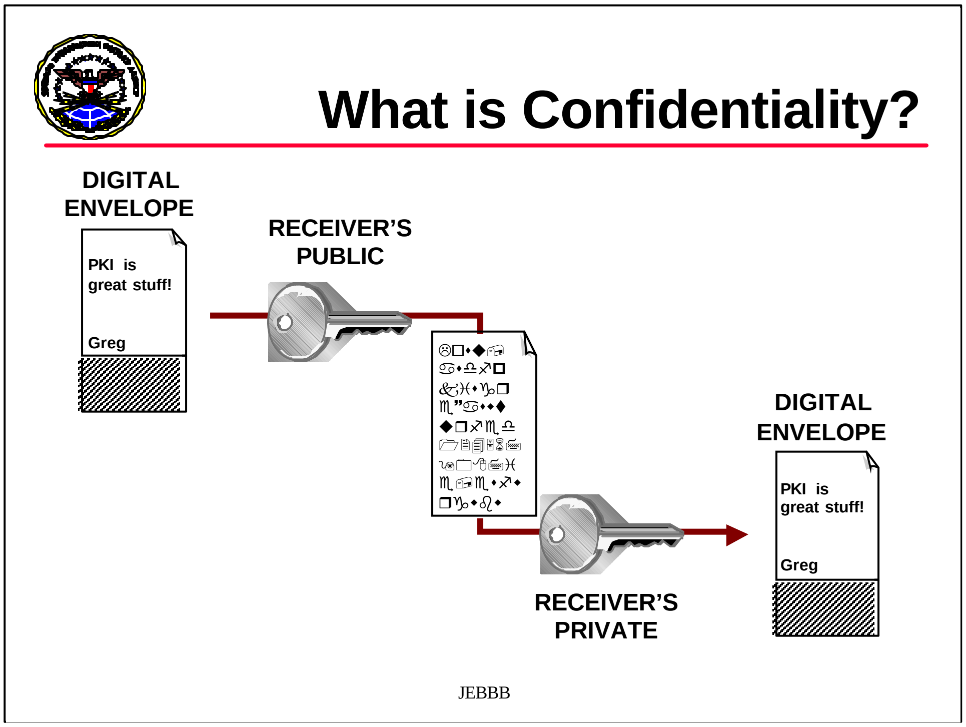


What is Non-Repudiation?

JEBBB

<u>Non-repudiation</u> - The sender cannot later deny the data.

The file is passed through a hashing algorithm which creates a unique numerical representation of the file. This is then encrypted with the sender's private key, and appended to the original file to create the Digital Envelope. The Digital Envelope is then encrypted with the receiver's public key.



What is Confidentiality?

DIGITAL ENVELOPE

PKI is great stuff!

Greg

RECEIVER'S PUBLIC

RECEIVER'S PRIVATE

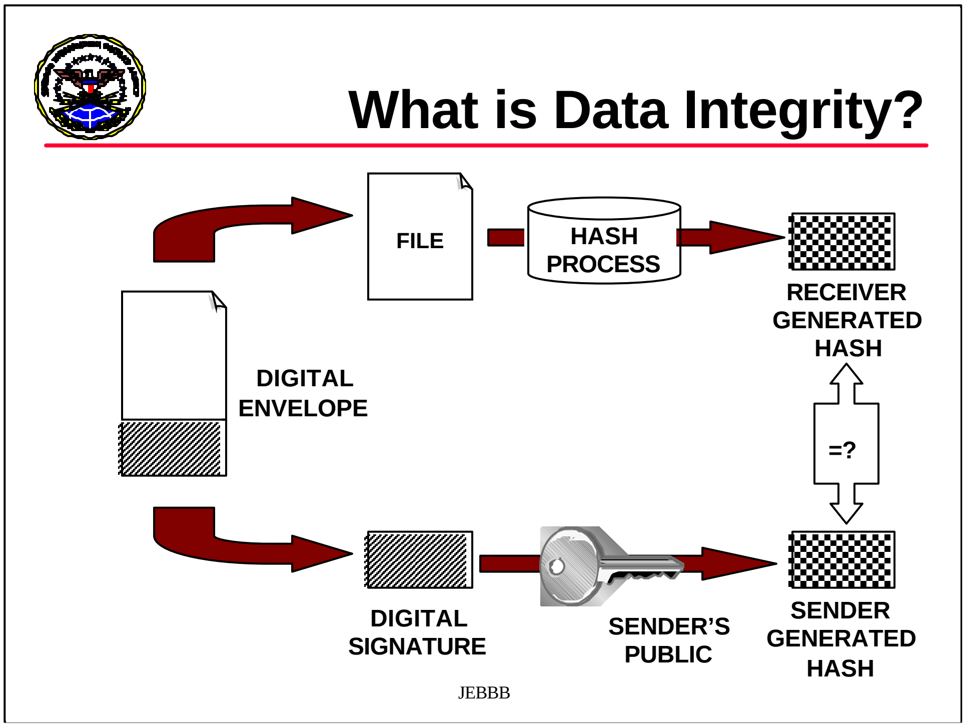
DIGITAL ENVELOPE

PKI is great stuff!

Greg

JEBBB

Once created, the Digital Envelope can be encrypted using the Receiver's readily available Public Key. This "sealed" Digital Envelope is then transmitted via the Internet. It can only be read after the intended Receiver has decrypted the message using his private key.



What is Data Integrity?

FILE

HASH PROCESS

RECEIVER GENERATED HASH

DIGITAL ENVELOPE

=?

DIGITAL SIGNATURE

SENDER'S PUBLIC

SENDER GENERATED HASH

JEBBB

To establish confidence that the received message has not been tampered with while in transit, the recipient does the following:

1. Using the same hashing algorithm as the sender the receiver creates a numerical representation of the file.

2. Using the sender's public key, decrypts the digital signature that accompanied the file to recover the hash output the sender created.

3. Compares the two numerical values to ensure they are the same.