INFS 5115 Security Principles

Cryptography

University of South Australia

School of
Information Technology
and Mathematical Sciences

Cryptography

- In this module, we will first review the topic of cryptography including symmetric and asymmetric encryption
- Then we will examine the concept of hashing and how it is used
- Finally, we will cover public key infrastructure and digital certificates

Concepts

Scytale

History

Enigma

Wikipedia 2019, ‘Scytale’, Wikipedia, https://en.wikipedia.org/wiki/Scytale

IWM 2018, ‘How Alan Turing Cracked the Enigma Code’, Imperial War Museums, https://www.iwm.org.uk/history/how-alan-turing-cracked-the-enigma-code

Concepts

Terminology

- **Cryptography** - the science of secret writing with the goal of hiding the meaning of a message
- **Cryptanalysis** - the science of breaking cryptosystems
- **Encryption** is the process of turning plaintext into ciphertext
- **Decryption** is the process of turning ciphertext into plaintext
- **Cryptology** – study of encryption & decryption, including cryptography & cryptanalysis

- Encryption / decryption requires: an algorithm and a key

Concepts

Example

The problem:

- I need to get a message to another person across the network but it needs to be confidential.

Solution:

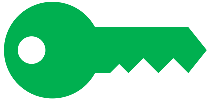
- The message can be encrypted using an encryption algorithm and a key and sent to the other person.

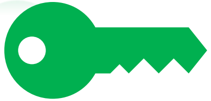
Concepts

If a message is to be encrypted by the sender the receiver will need the key used to encrypt so they can decrypt. There are two main types of encryption algorithms:

- **Symmetric algorithms** – two parties have an encryption and decryption method for which they **share** a secret key.
- **Asymmetric (or public key) algorithms** – a user possesses a secret key as in symmetric cryptography but also a public key.

Symmetric Cryptography

plaintext +  → *algorithm* → ciphertext

ciphertext +  → *algorithm* → plaintext

Symmetric Cryptography

The Key

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Plaintext: `SECRET (the message)`

Ciphertext: `vhfuhw (the resulting encrypted message)`

Symmetric Cryptography

Substitution Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
k	d	w	n	q	m	b	t	f	h	x	c	v	g	a	s	z	e	o	r	y	j	i	u	p	l

Plaintext: `THE QUESTION`

Ciphertext: `rtq zyqorfag`

Symmetric Cryptography

Substitution Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
k	d	w	n	q	m	b	t	f	h	x	c	v	g	a	s	z	e	o	r	y	j	i	u	p	l

Plaintext: `THE QUESTION FROM THE COMMITTEE`

Ciphertext: `rtq zyqorfag meav rtq wavvfrrqq`

Symmetric Cryptography

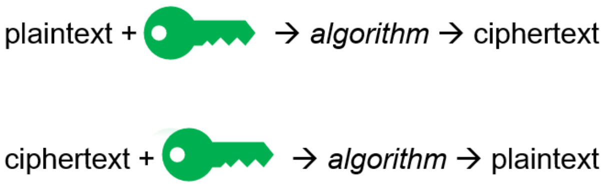
- DES (Data Encryption Standard)
  - published by NIST in 1977
  - 56bit key length, this short key length can be broken relatively easy nowadays
  - 3DES = encrypting DES 3 times in a row, much more robust but inefficient
- AES (Advanced Encryption Standard)
  - NIST replacement for DES published in 2001
  - 128-256bit key length
  - Intel micro-processors have incorporated this algorithm into their hardware to increase speed.

Paar, C & Pelzl, J 2010, 'Understanding Cryptography: A Textbook for Students and Practitioners', 2nd edn, Springer-Verlag, Berlin, pp. 55-89

https://www.intel.com.au/content/www/au/en/architecture-and-technology/advanced-encryption-standard-aes/data-protection-aes-general-technology.html

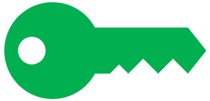
Symmetric Cryptography

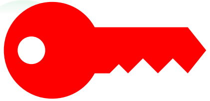
- Ideally suited to provide confidentiality
- Requires pre-shared secrets



- *How does the sender share the key with the recipient?*
- *If the key is sent across the network could it be intercepted by a third party?*
- *What would be the consequence if a third party had the key?*

Asymmetric Cryptography

plaintext +  → *algorithm* → ciphertext

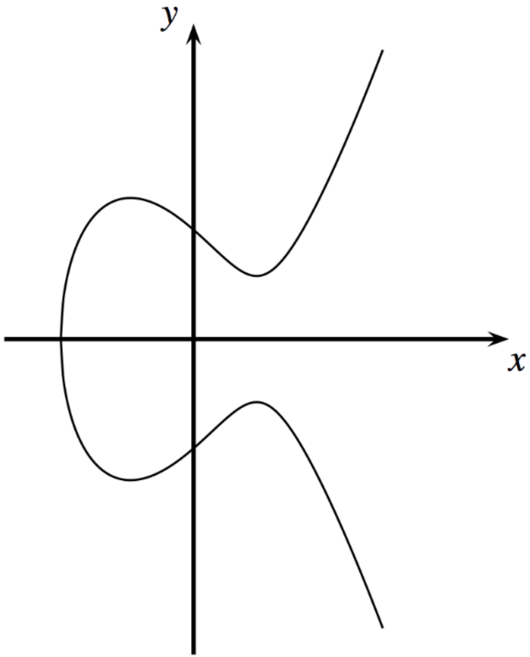
ciphertext +  → *algorithm* → plaintext

Asymmetric Cryptography

- Public vs private keys
  - ciphertext encrypted with public key can be decrypted with private key
  - ciphertext encrypted with private key can be decrypted with public key
- No need to possess a pre-shared secret key
- Much slower than symmetric cryptography (arithmetically intensive)
- Well suited to encrypting small amounts of data

Asymmetric Cryptography

- Examples:
  - RSA (Rivest-Sharmir-Adleman)
  - ECC (Elliptic Curve Cryptography)



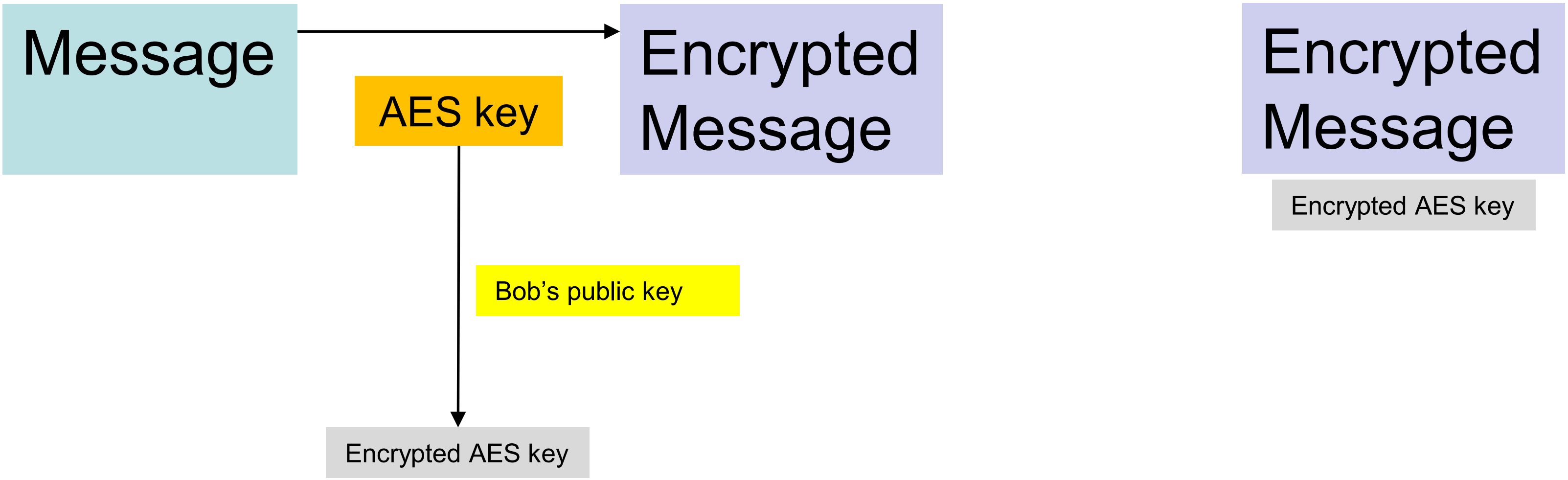
Asymmetric Cryptography

- In practice:
  - Anyone can encrypt a message for Bob with his public key, but only Bob can decrypt the messages with his private key
  - Bob can encrypt a message with his private key and anyone with the public key can decrypt it

Cryptography Example

- Alice does the following:
  - generates an AES key
  - encrypts the message using this key
  - encrypts the AES key using Bob’s public key
  - sends the encrypted message and the encrypted key to Bob
- Bob does the following;
  - decrypts the AES key using his private key
  - decrypts the message using the AES key

Cryptography Example



Scenario #1

- Prepare an encrypted message to send to your partner (receiver) using the example Caesar Cipher.
- Pass the key to your partner
- Is this an example of symmetric or asymmetric encryption?
- Which of the following does this scenario illustrate: confidentiality, integrity, availability?

For students enrolled externally, use the Caesar Cipher to decrypt the following message:
Li brx vwxgb kdug brx zloo kdyh d uhzduglqj fduhhu lq LW.

Diffie-Hellman Key Exchange

- Allows two parties to derive a common secret key over an insecure channel
- Modulo / modulus function:
  - $a \bmod b$ is the remainder when a is divided by b
    - $17 \bmod 3 = ?$
    - $2^3 \bmod 5 = ?$
    - $7^4 \bmod 13 = ?$
    - $7^8 \bmod 13 = ?$

https://www.calculators.org/math/modulo.php

Diffie-Hellman key exchange

Alice	Bob	Eve (Everyone)
$p = 13, g = 7$	$p = 13, g = 7$	$p = 13, g = 7$
$a = 8$		
$A = g^a \text{ mod } p = 3$	$A = 3$	$A = 3$
	$b = 4$	
$B = 9$	$B = g^b \text{ mod } p = 9$	$B = 9$
$s = B^a \text{ mod } p$	$s = A^b \text{ mod } p$	
$s = 3$	$s = 3$	

Diffie-Hellman Key exchange

Client	Server
Client gets a public number (P=23)	Server gets a public number (G=9)
Client chooses a private key value (a=4)	Server chooses a private key value (b=3)
Client knows Servers public number	Server knows Clients public number
Client computes public value of x by taking the public number of the server (9) to the power of its own private number (4) = 6561 then mod clients public number (23). This means divide 6561 by 23 to get remainder 6. So x = 6	Server computes public value of y by taking its own public number (9) to the power of its own private number (3) = 729 then mod clients public number (23). This means divide 729 by 23 to get remainder 16. So y = 16
Client sends calculated public number (6) to server	Server sends calculated public number (16) to client
Client computes symmetric key (KeyA) Y(16) to the power of a(4) = 65536 mod p(23) to get remainder 9. So KeyA = 9	Server computes symmetric key (KeyB) X(6) to the power of b(3) = 216 mod p(23) to get remainder of 9. So KeyB = 9
9 is the shared secret	9 is the shared secret

Source

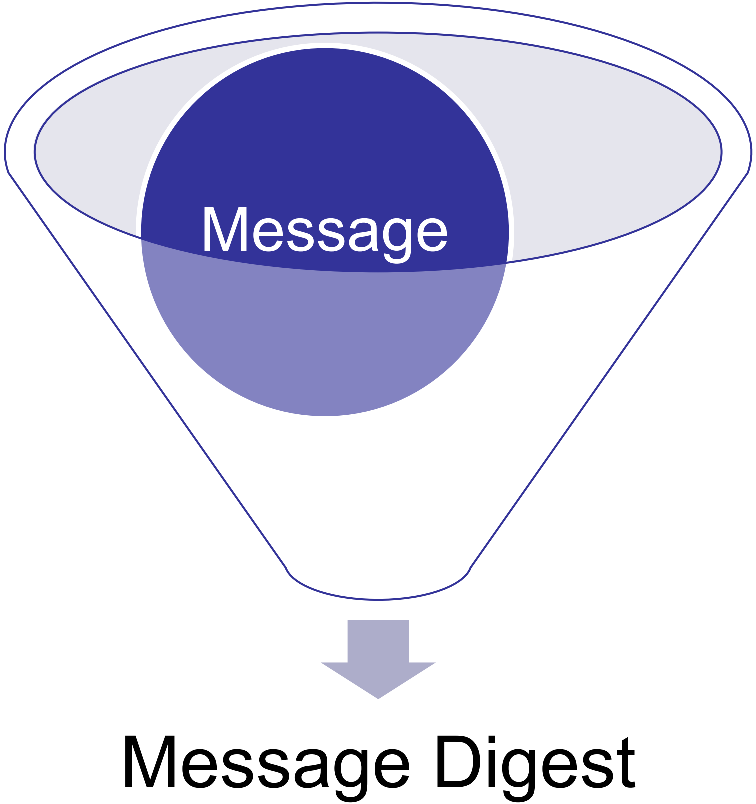
Scenario #2

- One person should play each of the following roles:
  - Client, Server
- Using the following values, step through the Diffie-Hellman key generation process:
  - Client p = 5, Server g = 7

What capability is provided by the Diffie-Hellman approach?

Diffie-Hellman calculator to check your answers:
https://www.irongeek.com/diffie-hellman.php?

Cryptographic Hash Functions



Cryptographic Hash Functions

- Uses:
  - Message integrity
  - Digital signatures (more on this later)
  - Storing passwords
- No key

Cryptographic Hash Functions

- One-way
- It is not feasible to modify a message without changing its hash value
- Strong collision resistance – highly unlikely that any two inputs will hash to the same output
- Compression – usually a fixed size output, smaller than the input
- Efficiency

Cryptographic Hash Functions

P@ssw0rd1

MD5: 8B8E9715D12E4CA12C4C3EB4865AAF6A
SHA1: F2A12F187EBB7080BD75AAC9160214E6B1E49F7D

Verify that this message has not been tampered with

MD5: 9170724B2BE3B30C169236F3D9EEB88D
SHA1: 4A35114486A27E7F126434F206703BD271D3120D

Cryptographic Hash Functions

- Alice does the following:
  - applies the hash function to the message to produce the message digest
  - sends Bob the message and the message digest
- Bob does the following:
  - applies the hash function to the message to produce the message digest
  - compares the output with the message digest sent by Alice

Cryptographic Hash Functions

- Plaintext password is hashed and the **result** is stored
- During authentication, a user provides the plaintext password, which is hashed and compared to the stored hash value

user_id	user	password
1	admin	5f4dcc3b5aa765d61d8327deb882cf99
2	gordonb	e99a18c428cb38d5f260853678922e03
3	1337	8d3533d75ae2c3966d7e0d4fcc69216b
4	pablo	0d107d09f5bbe40cade3de5c71e9e9b7
5	smithy	5f4dcc3b5aa765d61d8327deb882cf99

Graham, J, Olson, R, & Howard, R (eds) 2010, Cyber Security Essentials, CRC Press, Boca Raton.

Cryptographic Hash Functions

- An adversary can generate a ‘dictionary’ which relates passwords to the hash value (depends on the hashing algorithm used) – these ‘dictionaries’ are called rainbow tables
- To mitigate against rainbow tables, salt values can be added to the password before hashing
- Salts are generally random data of a given length stored in plaintext with the hashed password

Graham, J, Olson, R, & Howard, R (eds) 2010, Cyber Security Essentials, CRC Press, Boca Raton.

Cryptographic Hash Functions

Hash value created on a Cisco Router

2ac9cb7dc02b3c0083eb70898e549b63

- Browse to https://crackstation.net and paste the hash above into the field.
- Select **Crack Hashes**.
- What is the password?

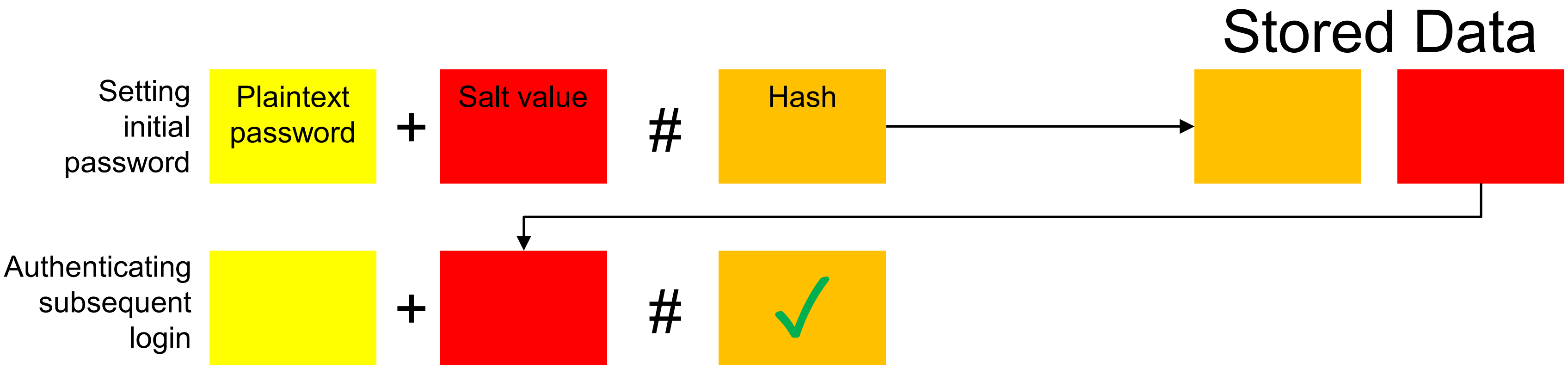
Cryptographic Hash Functions

- To mitigate the damage that a hash table or a dictionary attack could do, we salt the passwords. A salt makes a hash function look non-deterministic, which is good as we don't want to reveal duplicate passwords through our hashing.

- Let’s say that we have password ”password1” and the salt xyz. We can salt that password by either appending or prepending the salt to it. This will yield password1xyz or xyzpassword1.

Cryptographic Hash Functions

Password Storage



Scenario #3

- If a user has set their password to P@s5w0rd, and a salt value of 08423 is being used, what will the system store in order to validate this user's password?
- What steps will the system take to validate the password?
- What is the main advantage of using salting?
- What is the main advantage of storing hash values rather than plaintext passwords?

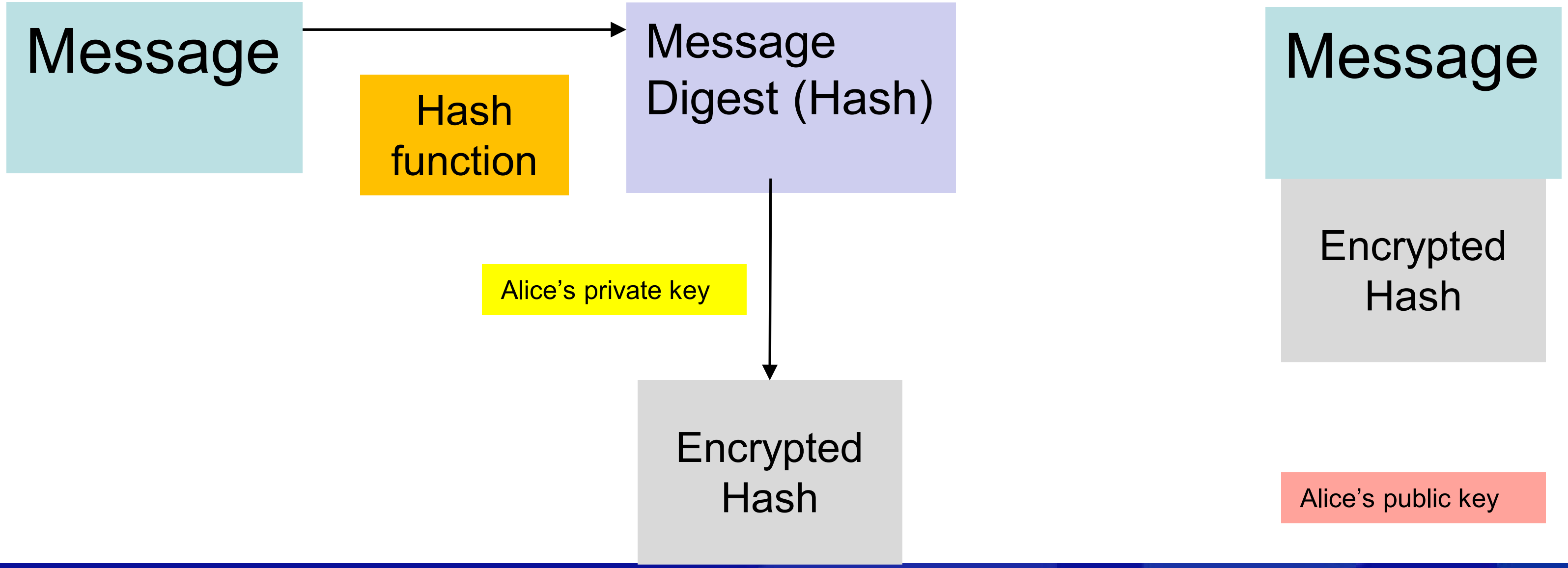
Digital Signatures

- Asymmetric encryption + hashing can be used to implement digital signatures
- Provides integrity assurance and non-repudiation
- Commonly achieved by hashing the message, encrypting the hash using the sender’s private key and ‘attaching’ the encrypted hash to the message.
- The sender’s public key may also be sent with the message.

Digital Signatures

- Alice does the following:
  - Applies hash function to message to create message digest (hash)
  - Uses her private key to encrypt the hash
  - Sends message + encrypted hash + her public key to Bob
- Bob does the following:
  - Decrypts the encrypted hash using Alice's public key
  - Applies hash function to message and compares this result to the decrypted hash
- Note that non-repudiation depends on Bob knowing that only Alice holds the private key – requires a trusted third party

Digital Signatures

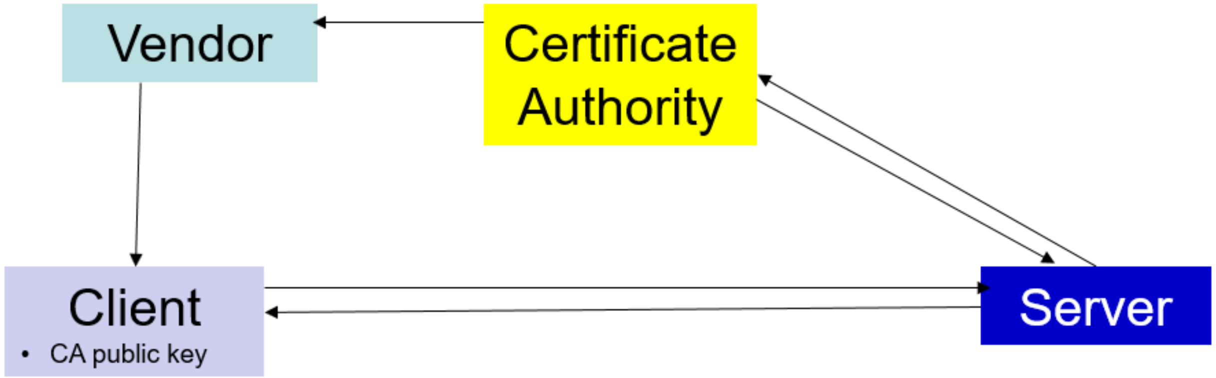


Digital Signatures

- How does Alice distribute her public key to Bob?
- How does Bob trust that the public key he received from Alice has not been tampered with by somebody else?

Public Key Infrastructure

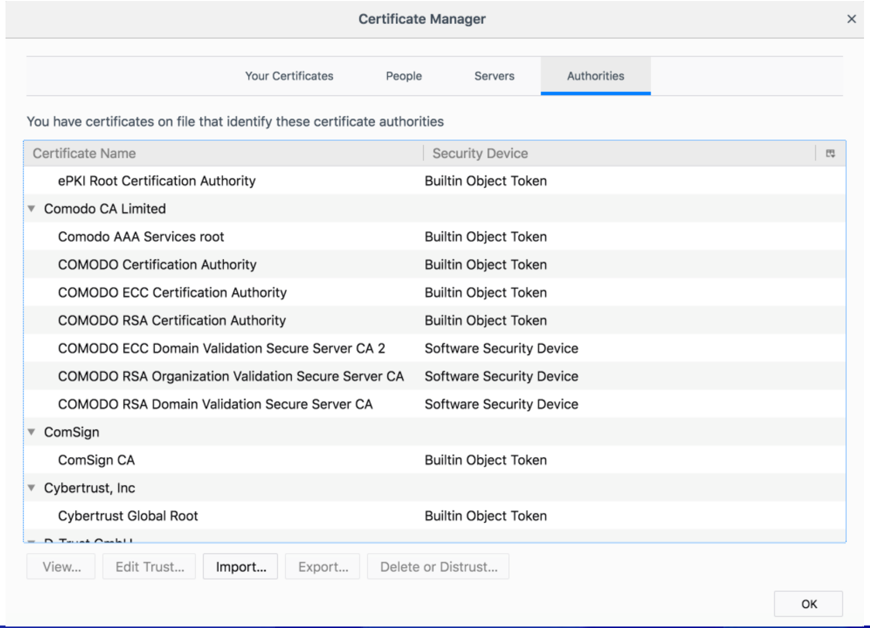
A public key infrastructure (PKI) binds public keys to entities, enables other entities to verify public key bindings, and provides the services needed for ongoing management of keys in a distributed system.



Kuhn, DR, Hu, VC, Polk, WT, Chang, S *2001 Introduction to Public Key Technology and the Federal PKI Infratructure, National* Institute of Standards and Technology, SP 800-32, p. 15

Public Key Infrastructure

- Vendors (e.g. operating system & software providers) trust the Certificate Authority and include the public key of the CA in their products



Public Key Infrastructure

- Server:
  - generates public and private keys
  - requests certificate from CA (certificate will contain server public key and will be encrypted using CA private key)
- Client:
  - decrypts certificate using CA public key
  - uses server’s public key to establish secure communications

Scenario #4

- One person should play each of the following roles:
  - Vendor, Certificate Authority, Client (User), Server (Provider)
- For each role, in the order specified below, outline the steps to be taken to engage in / support secure communications between the client and the server:
  - Vendor / Certificate Authority
  - Server / Certificate Authority
  - Client / Server