Western Norway University of Applied Sciences

Public Key Infrastructure and Cryptography in Blockchain

DAT159 – Basic Cryptography Module

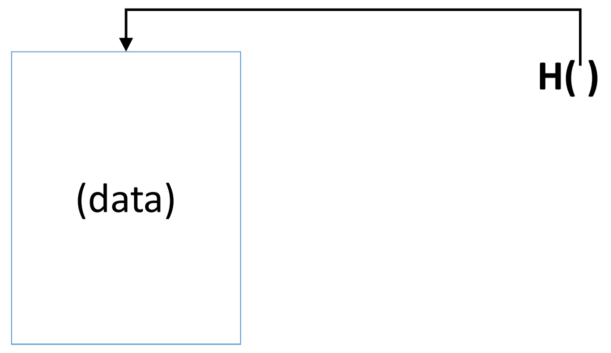
Tosin Daniel Oyetoyan

Cryptography in Blockchain

- Hashing
  - Preserves the integrity of transactions/records along the chain
- Digital Signature
  - Authenticity and non-repudiation
- Public Keys
  - Identity and anonymity

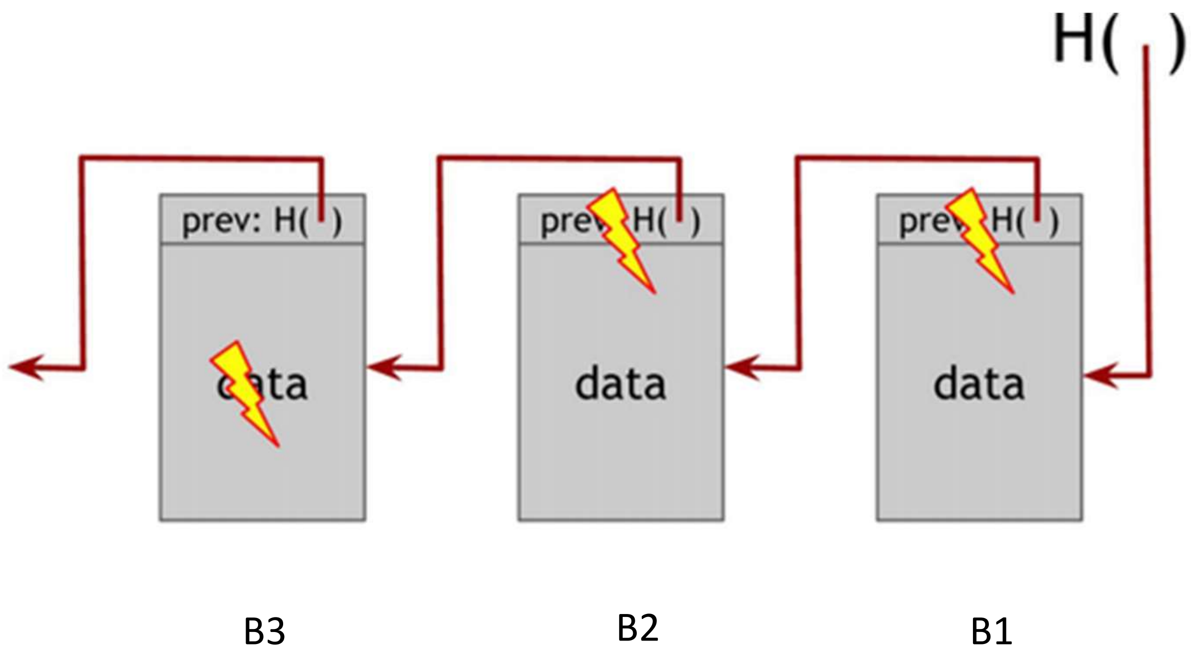
Hash Pointers and Data Structures

- A hash pointer is a pointer of where data is stored together with the hash value of this data at a given point in time
- Linked List hash pointers
- Merkle Tree hash pointers



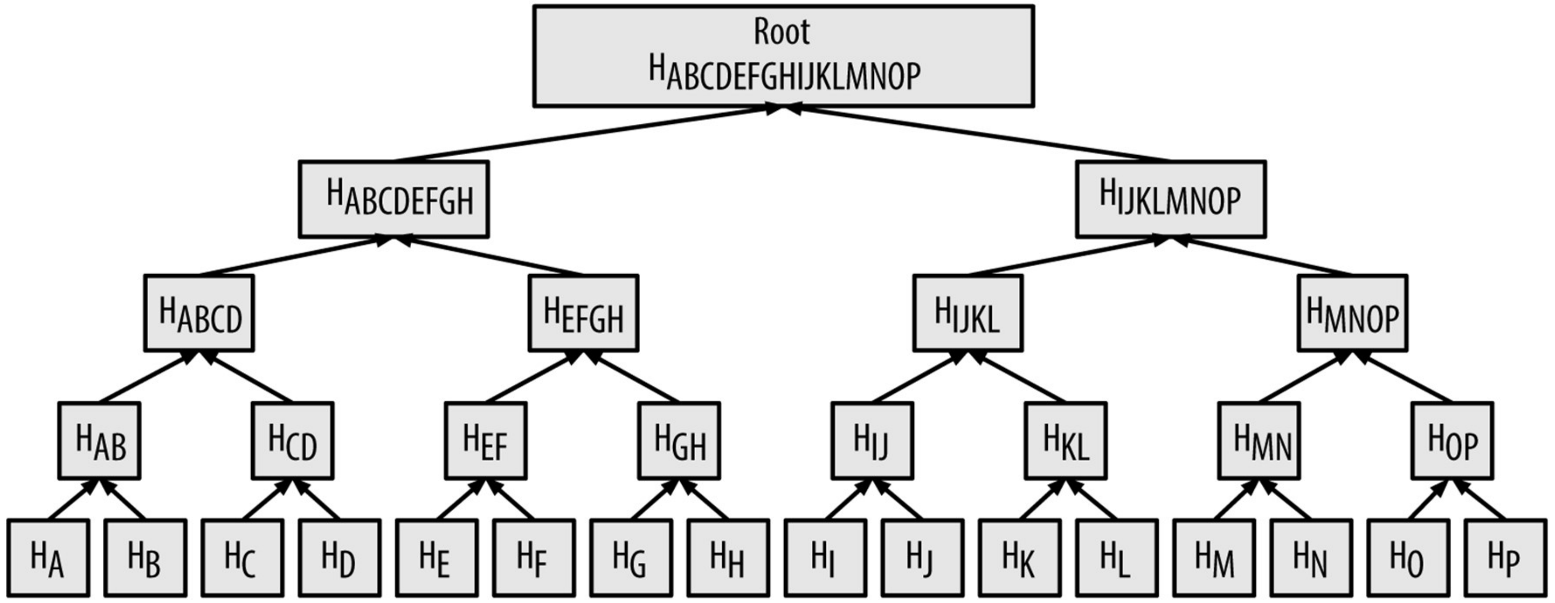
Linked list hash pointers

- Tamper-resistant feature
- Changes in one block in the chain affect all the subsequent blocks



Merkle Trees

- Binary Tree
- Acyclic graph
- Node is the **hash** of the leaves
- Any change in any of the leaf will change the value of the node

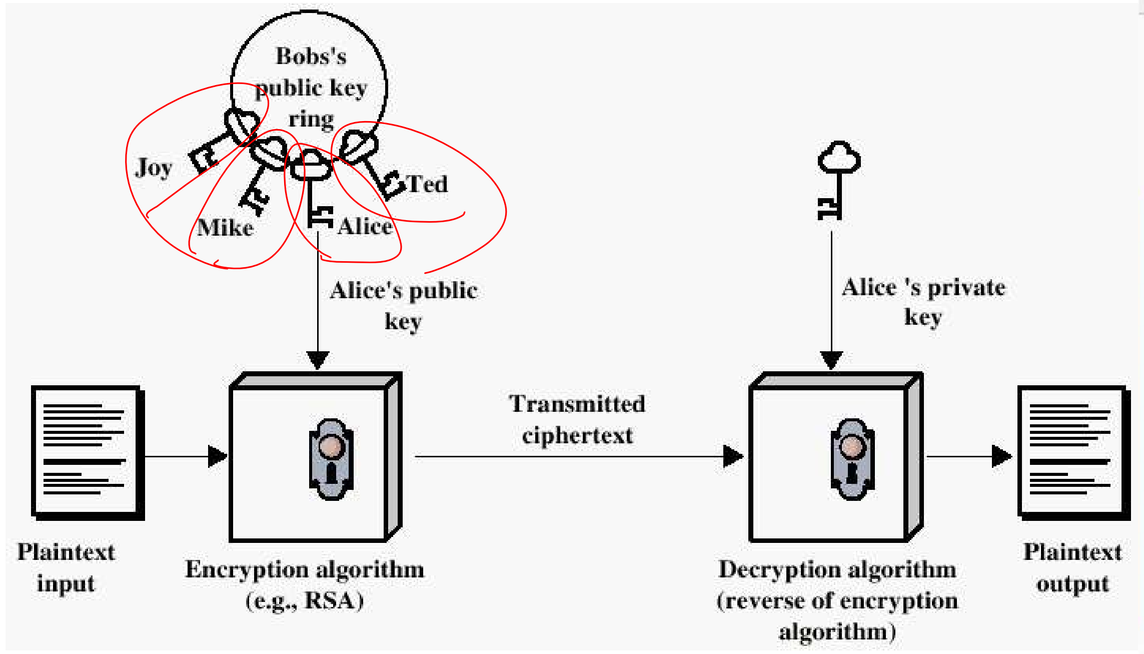


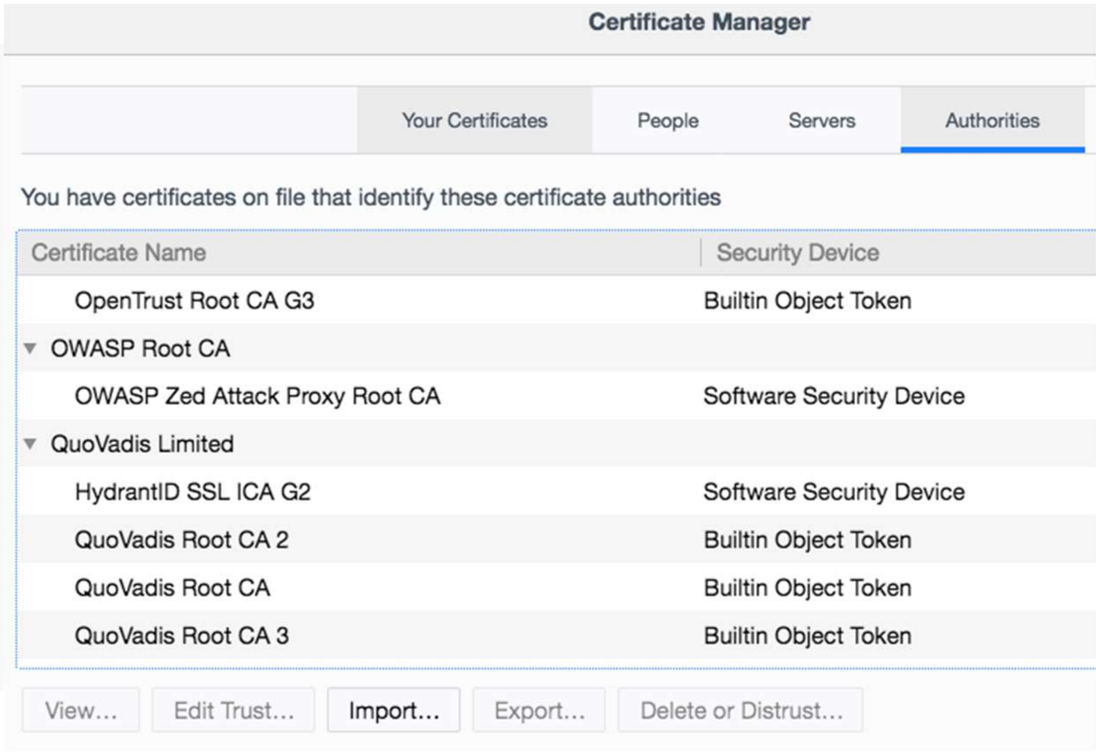
Digital Signature and Public Keys

- Sign transactions with private keys (**Digital signature**)
- Use your **public keys** as your identity to verify your transactions

Public Key Infrastructure (PKI)

Encryption with public keys




Certificate Manager

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
OpenTrust Root CA G3	Builtin Object Token
▼ OWASP Root CA	
OWASP Zed Attack Proxy Root CA	Software Security Device
▼ QuoVadis Limited	
HydrantID SSL ICA G2	Software Security Device
QuoVadis Root CA 2	Builtin Object Token
QuoVadis Root CA	Builtin Object Token
QuoVadis Root CA 3	Builtin Object Token

Bob’s Signature

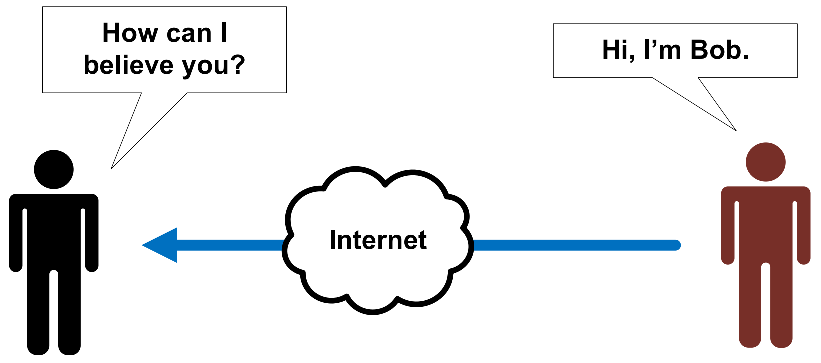
Message = "Hello Alice"
Bob => Sign(Message, Bob’s Private Key) = Signature

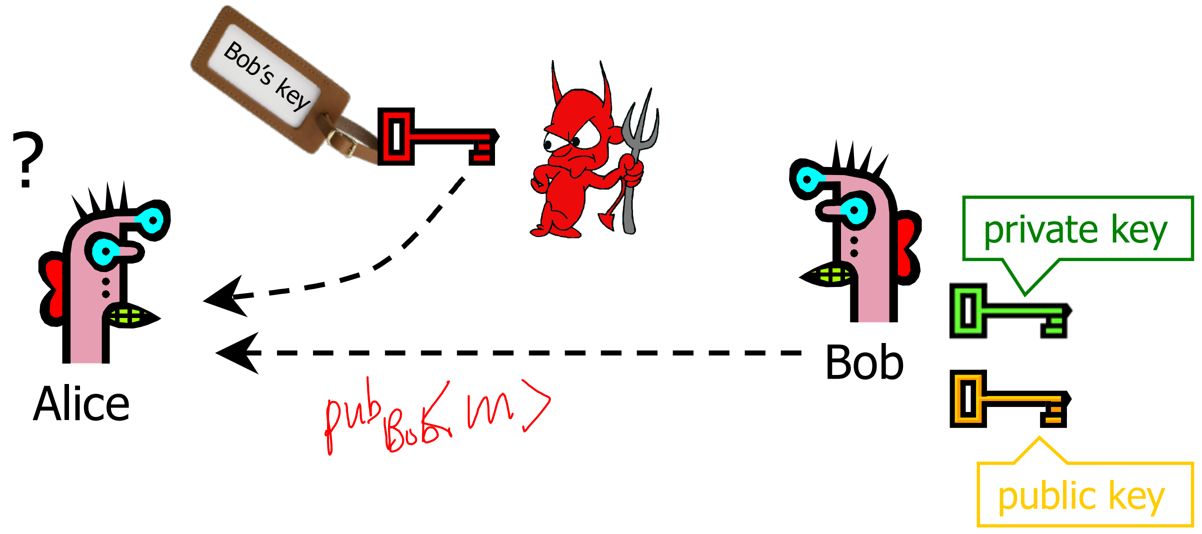
Alice => Verify(Message, Signature, Bob’s Public Key) = True
-> I believe

So, how do we obtain Bob’s Key?

- How do we know Bob’s public key?
- Ask him?
- Should there be a ‘trusted’ middle party for public key distribution?
    - Any difference with symmetric key?

Man-In-The-Middle



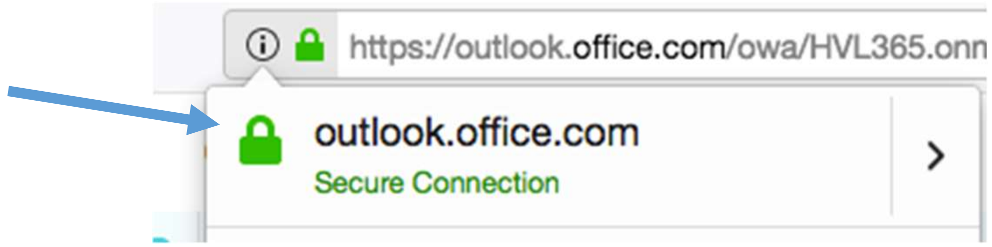


Distributing public keys

- Announce them publically or put them in a public directory
  - Threats: Forgery and Tampering
- Public-key certificate
  - Signed statement specifying the key and identity
    - $\text{sig}_{\text{Alice}}(\text{"Bob"}, K_{\text{pub}})$
- Common approach
  - Certificate authority (CA)
  - Offline proof of identify and knowledge of the private key to obtain CA’s certificate for the public key.

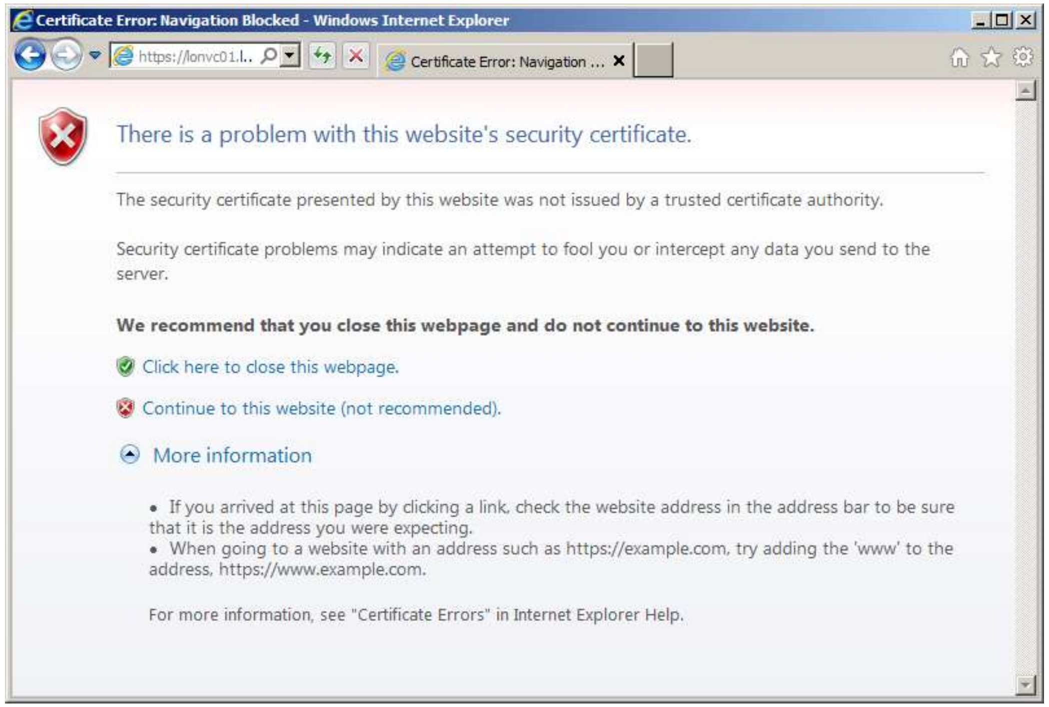
What is a Public Key Infrastructure

- PKI is a set of system, software and communication protocols to use, manage and control public keys
  - Publish public keys and certificates
  - Certify that a key is tied to an individual or entity
  - Provide verification of the validity of a public key
- PKIs bind an identity to a public key
- That a public key belongs to an individual that you know and trust
- Allows you to trusts servers you have never worked with before



Pitfalls

- Security breaches
  - Key compromises
- Revocation difficulties
- Negligence
  - Certificates are routinely not checked or some of the attributes ignored
  - Alarms and warnings ignored
    - *("Certificate not valid. Press OK to proceed")*



Certificate Error: Navigation Blocked - Windows Internet Explorer

https://lonvc01.l..

Certificate Error: Navigation ...

There is a problem with this website's security certificate.

The security certificate presented by this website was not issued by a trusted certificate authority.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

Continue to this website (not recommended).

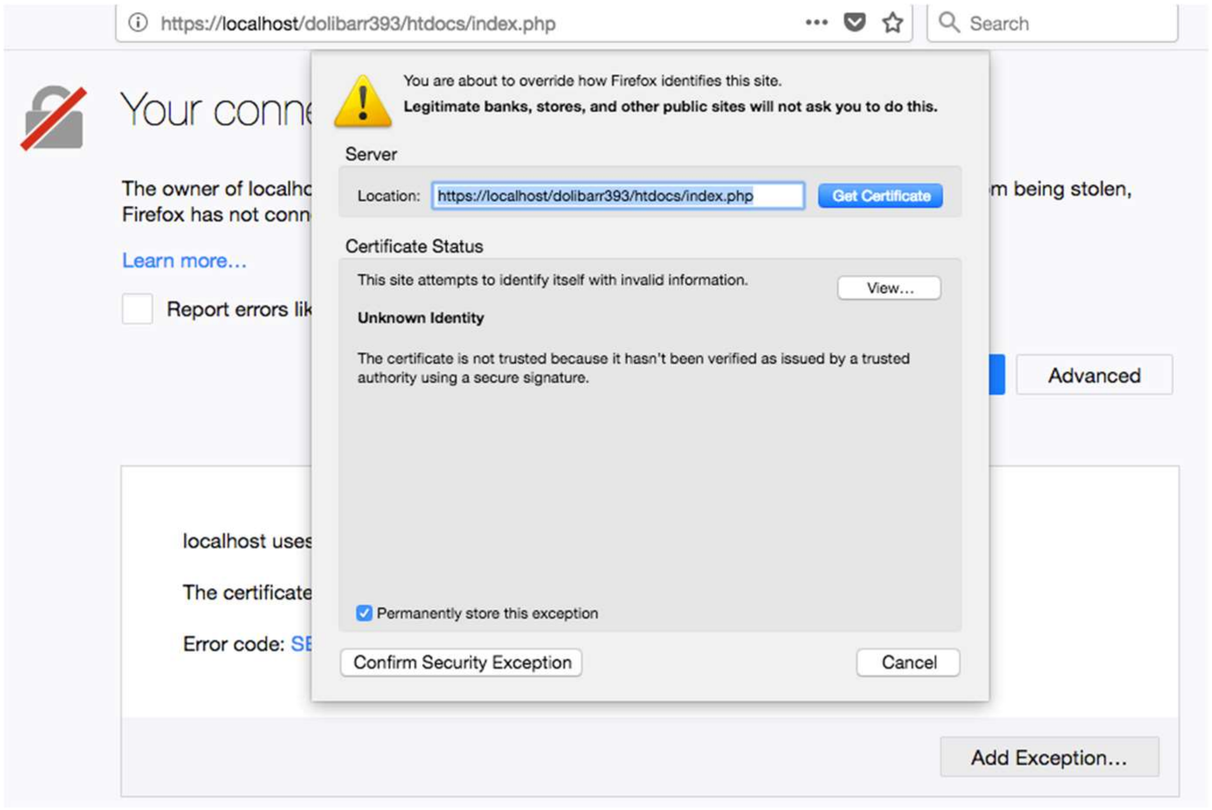
More information

- If you arrived at this page by clicking a link, check the website address in the address bar to be sure that it is the address you were expecting.
- When going to a website with an address such as https://example.com, try adding the 'www' to the address, https://www.example.com.

For more information, see "Certificate Errors" in Internet Explorer Help.

If someone breaches the security of CA?

Misuse



https://localhost/dolibarr393/htdocs/index.php

Search

Your conn

The owner of localho … m being stolen,
Firefox has not conn

Learn more…

Report errors lik

Advanced

localhost uses

The certificate

Error code: SE

Add Exception…

You are about to override how Firefox identifies this site.
Legitimate banks, stores, and other public sites will not ask you to do this.

Server

Location: https://localhost/dolibarr393/htdocs/index.php Get Certificate

Certificate Status

This site attempts to identify itself with invalid information. View…

Unknown Identity

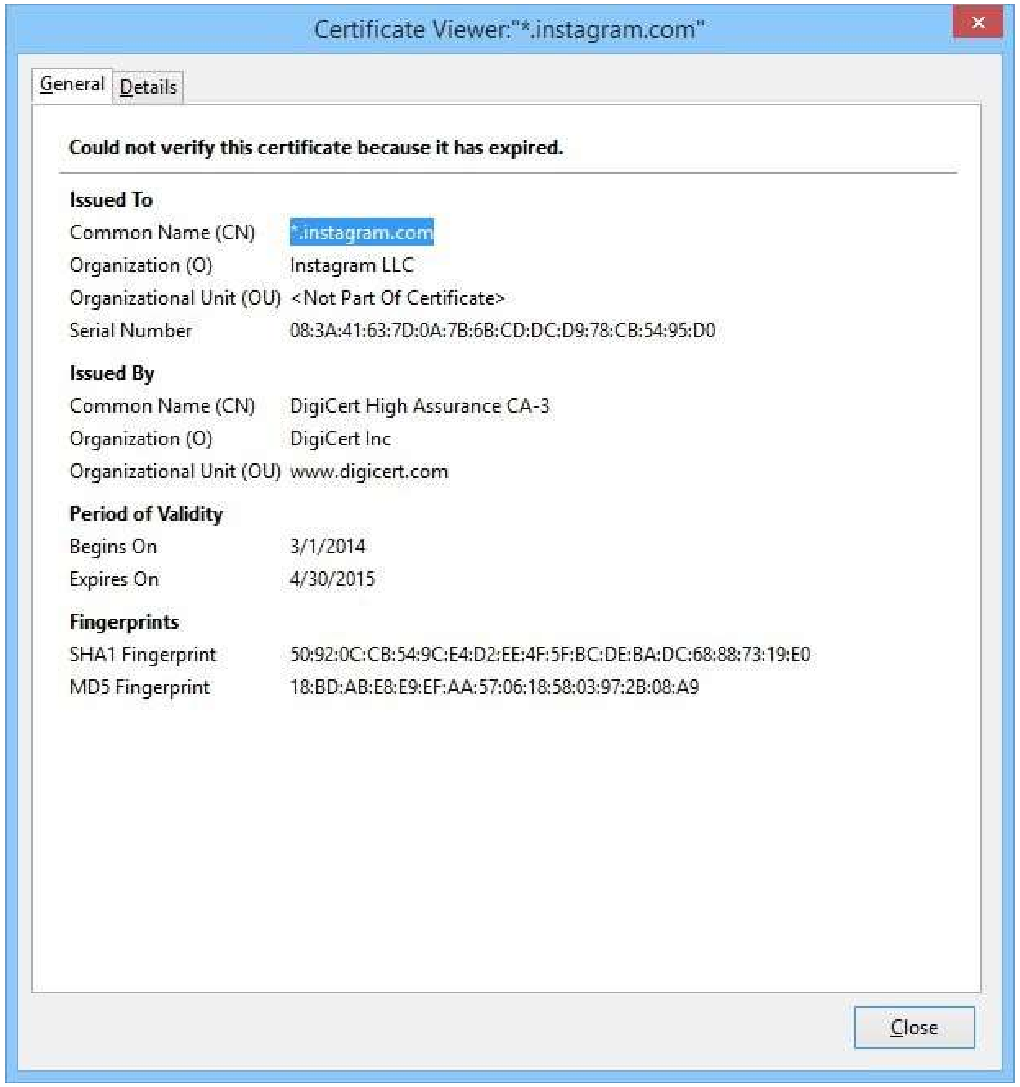
The certificate is not trusted because it hasn't been verified as issued by a trusted authority using a secure signature.

[✓] Permanently store this exception

Confirm Security Exception Cancel

Elements of PKI

- Digital Certificates
  - Proposed by Loren Kohfelder at MIT in 1978 (Bachelor’s thesis)
  - Authenticated identifier pairing the public key to a significant name
  - Such that any user can identify themselves



Certificate Viewer:"*.instagram.com"

Could not verify this certificate because it has expired.

Issued To

Common Name (CN)	*.instagram.com
Organization (O)	Instagram LLC
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	08:3A:41:63:7D:0A:7B:6B:CD:DC:D9:78:CB:54:95:D0

Issued By

Common Name (CN)	DigiCert High Assurance CA-3
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com

Period of Validity

Begins On	3/1/2014
Expires On	4/30/2015

Fingerprints

SHA1 Fingerprint	50:92:0C:CB:54:9C:E4:D2:EE:4F:5F:BC:DE:BA:DC:68:88:73:19:E0
MD5 Fingerprint	18:BD:AB:E8:E9:EF:AA:57:06:18:58:03:97:2B:08:A9

Elements of PKI

- Certificate Authorities (CA)
  - Symantec, Netscape, Verisign, Entrust, RSA Keon
- Public/Private Key Pairs - Key management
- PKIs - models
  - Certificate Authority based: x.509 Identity Certificates
  - Web of Trust based: PGP – Pretty Good Privacy (Email)
  - etc

Why should we trust CAs?

Certificates

- Certificates $\neq$ Signature
  - Certificates are *implemented using* Signatures

- Certificates $\neq$ Authentication
  - Authentication *can* be *implemented using* Certificates

- Certificates are *static*
  - Change => Re-Issue

Certificate Authority

- Issuer/Signer of the certificate
  - Binds public key with identity+attributes
- Enterprise CA
- Individual as CA (PGP)
  - Web of trust
- “Global” or “Universal” CAs
  - VeriSign, Equifax, Entrust, CyberTrust, Identrus, …
- Local CAs
- “Trust is the key word”

Read the article: Ten Risks of PKI: What You’re not Being Told about Public Key Infrastructure By Carl Ellison and Bruce Schneier

Certificate Authority

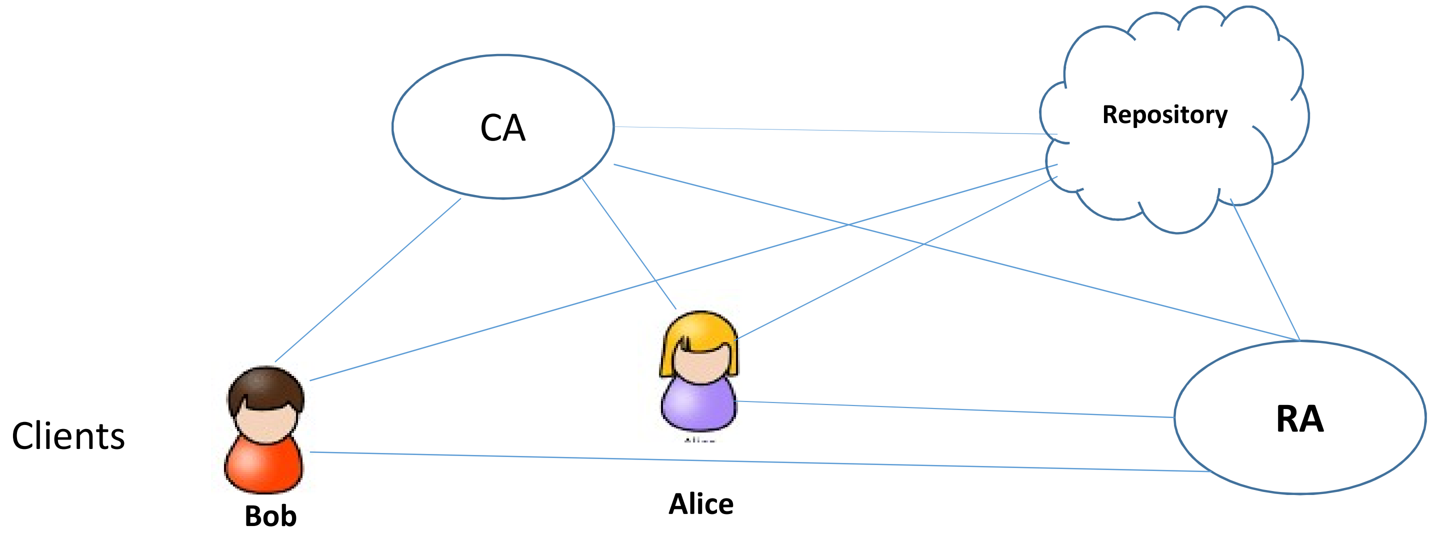
- A trusted third party - must be a secure server
- Signs and publishes X.509 Identity certificates
- Revokes certificates and publishes a Certification Revocation List (CRL)
- Many vendors
  - OpenSSL - open source, very simple
  - Netscape - free for limited number of certificates
  - Entrust - Can be run by enterprise or by Entrust
  - Verisign - Run by Verisign under contract to enterprise
  - RSA Security - Keon servers

RA – Registration Authority

- Also called LRA – Local RA
- Goal: Off-load some work of CA to LRAs
- Support all or some of:
  - Identification
  - User key generation/distribution
    - passwords/shared secrets and/or public/private keys
  - Interface to CA
  - Key/certificate management
    - Revocation initiation
    - Key recovery

PKI Elements

- CA issues and revokes certificates to a client
- RA checks the client’s identity
- Client have and use certificates
- Repository stores the certificates and status information

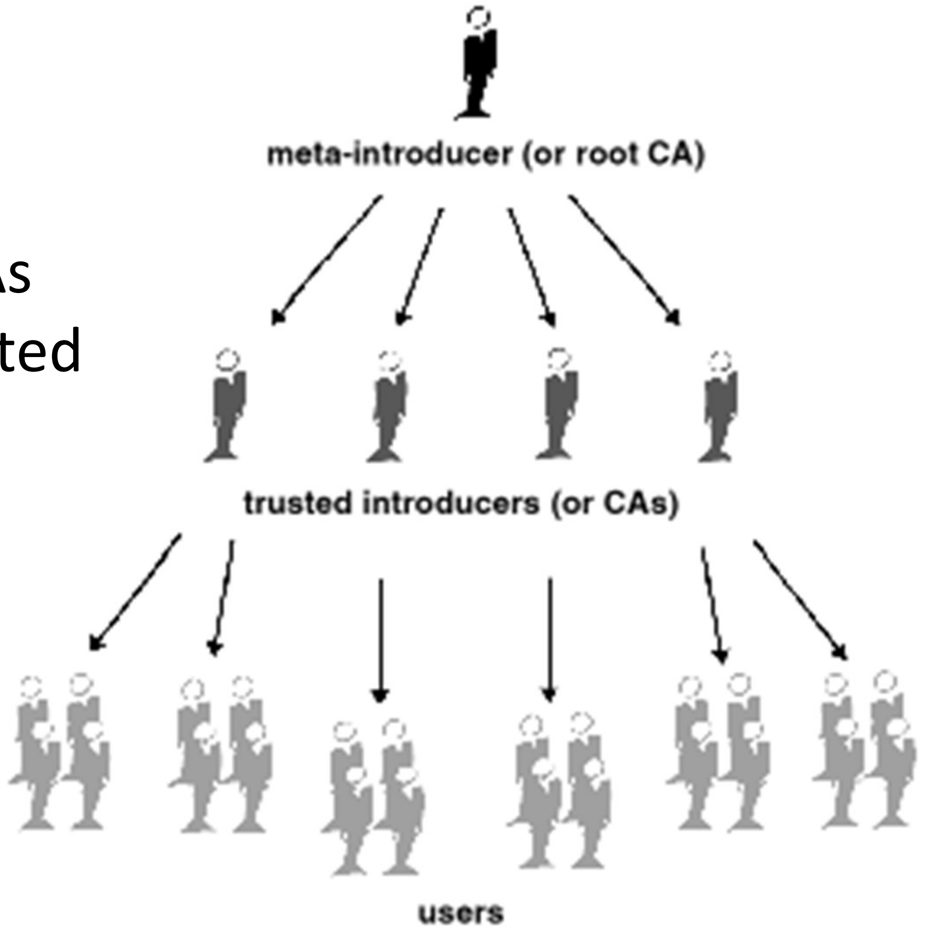


Certificate Chains

- Used to provide better scalability and control of certificates
- 2 types of certficate authorities
  - Root CAs
  - Intermediate CAs that have certificates signed by the root CA
    - Easier for local CAs to verify the identities of local users
- If the certificate used by a SSL/TLS server is signed by an intermediate CA, then client has to verify the chain
  - That server’s certificate is signed by the intermediate CA
  - That the intermediate CA’s certificate is signed by a root CA that the client trusts

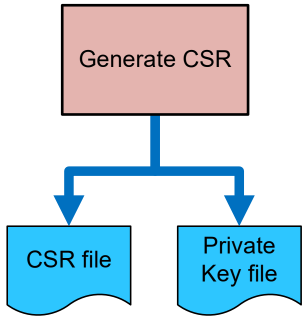
Root CAs and Intermediate CAs

- Root CAs are usually shipped by the manufacturers
- A root CA can authorize intermediate CAs
- The root CA uses the private key associated with a root CA’s certificate to sign certificates



Obtaining a certificate

- Client generates a certificate signing request (CSR)
- A CSR/PEM file and a Private Key
- Private key is used to sign the CSR file

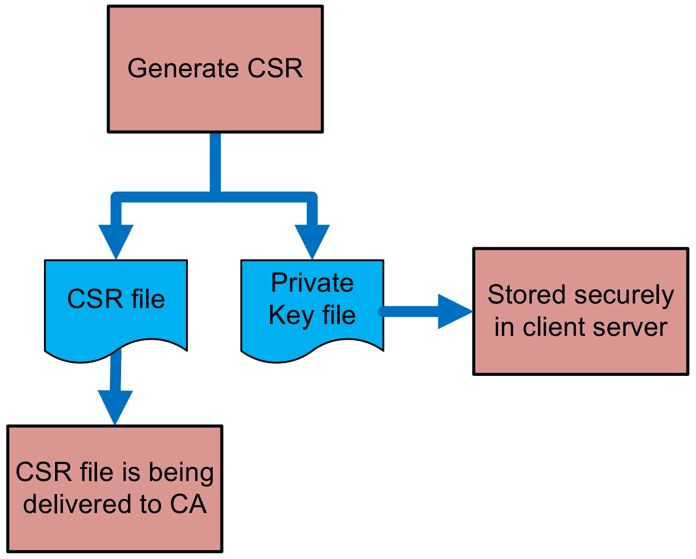


```
Owner: CN=localhost, OU=<<Org unit>>, O=<<Org>>, L=<<City>>, ST=<<State>>, C=<<Country code>>
Issuer: CN=localhost, OU=<<Org unit>>, O=<<Org>>, L=<<City>>, ST=<<State>>, C=<<Country code>>
Serial number: 519e7165
Valid from: Thu May 23 20:43:33 BST 2018 until: Wed Aug 21 20:43:33 BST 2019
Certificate fingerprints:
MD5:  34:B7:71:CD:C9:56:9A:EA:0C:F2:91:50:EA:7F:4B:64
SHA1: AA:DE:EC:1B:27:8E:BC:3A:7A:82:8C:B7:FA:C3:AA:11:2F:97:1F:2C
Signature algorithm name: SHA1withRSA
Version: 3
```

```
sudo openssl req -new -key /etc/apache2/server.key -subj "/C=NO/ST=HO/L=BE/O=/CN=tosin/emailAddress=/" -out /etc/apache2/server.csr
```

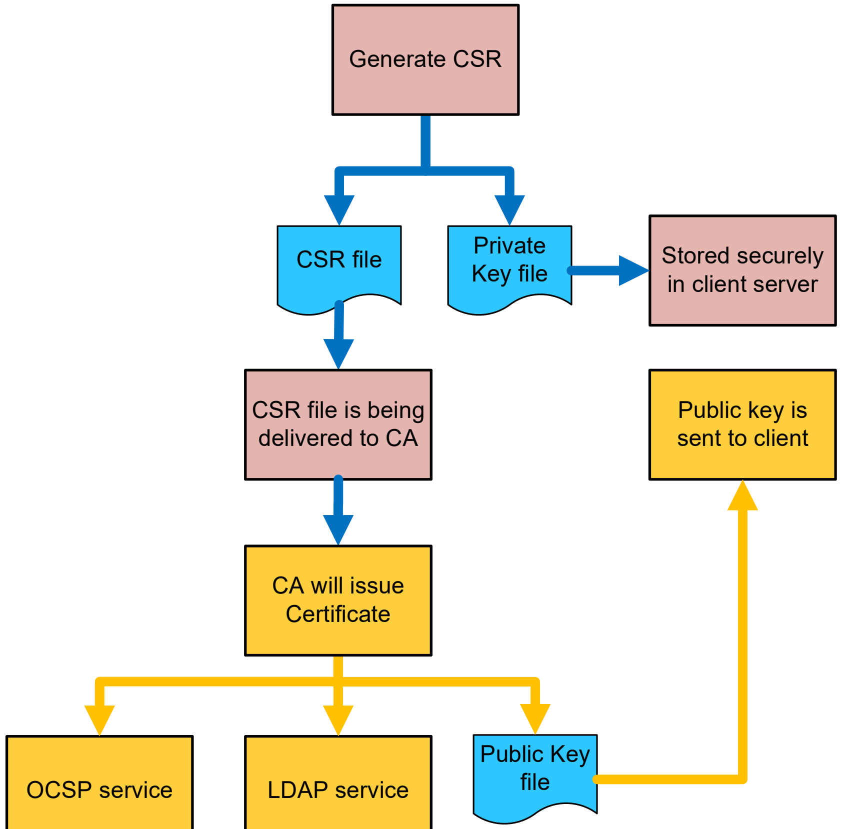
Obtaining a certificate

- CSR file is sent to the CA
- Private key is stored securely on client's computer


Generate CSR
CSR file
Private Key file
Stored securely in client server
CSR file is being delivered to CA

Obtaining a certificate

- CA signs and issue the certificate which is instantly available in their LDAP and OCSP services
  - LDAP – Lightweight Directory Access Protocol
  - OCSP – Online Certificate Status Protocol


Generate CSR
CSR file
Private Key file
Stored securely in client server
CSR file is being delivered to CA
Public key is sent to client
CA will issue Certificate
OCSP service
LDAP service
Public Key file

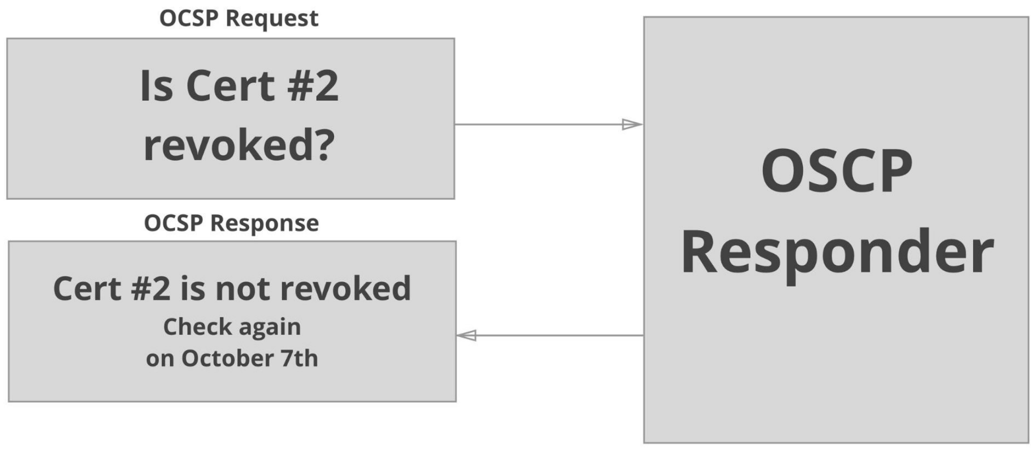
sudo openssl req -extensions v3_req -config /etc/apache2/ssl/localhost.conf -x509 -nodes -newkey rsa:4096 -keyout /private/etc/apache2/server.key -out /private/etc/apache2/server.crt -days 365 -subj "/C=NO/ST=HO/L=BE/O=Tosin/CN=localhost"

LDAP server

- Lightweight Directory Access Protocol (IETF standard)
  - Evolved from DAP and X.500 Identities
- Used by CA's to store user's Identity Certificate
- Standard protocol for lookup, entry, etc.
- Access control is implemented by user, password.

OCSP – Online Certificate Status Protocol

- is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate.
- List of non-valid certificates that should not be accepted by any member of the PKI
- CA can revoke certificates and provide updates to other PKI members via CRL



Certificate validation

- *Integrity*: signature is valid
- Signed by a *trusted* CA
    - *or certification path is rooted in a trusted CA*
- Certificate is valid *now*:
    - We are between *Not Valid Before* and *Not Valid After* time points in the certificate
- *Not Revoked*
- *Use* is *consistent* with the *policy*

Truststore and keystore

- Truststore
  - Stores the public keys and certificates
  - sudo security add-trusted-cert -d -r trustRoot -k /Library/Keychains/System.keychain /private/etc/apache2/server.crt
- Keystore
  - Where the private keys are stored
    - /private/etc/apache2/server.key

Another model: Web of Trust

- PGP – Pretty Good Privacy
- peer to peer
- Uses 2 certificates model
  - PGP Certificate
  - X.509 Certificate

PGP certificate format

- **The PGP version number —** this identifies which version of PGP was used to create the key associated with the certificate.
- **The certificate holder's public key —** the public portion of your key pair, together with the algorithm of the key: RSA, DH (Diffie-Hellman), or DSA (Digital Signature Algorithm).
- **The certificate holder's information —** this consists of "identity" information about the user, such as his or her name, user ID, photograph, and so on.
- **The digital signature of the certificate owner —** also called a *self-signature,* this is the signature using the corresponding private key of the public key associated with the certificate.
- **The certificate's validity period —** the certificate's start date/ time and expiration date/ time; indicates when the certificate will expire.
- **The preferred symmetric encryption algorithmfor the key —** indicates the encryption algorithm to which the certificate owner prefers to have information encrypted. The supported algorithms are CAST, IDEA or Triple-DES.

PGP Certificate

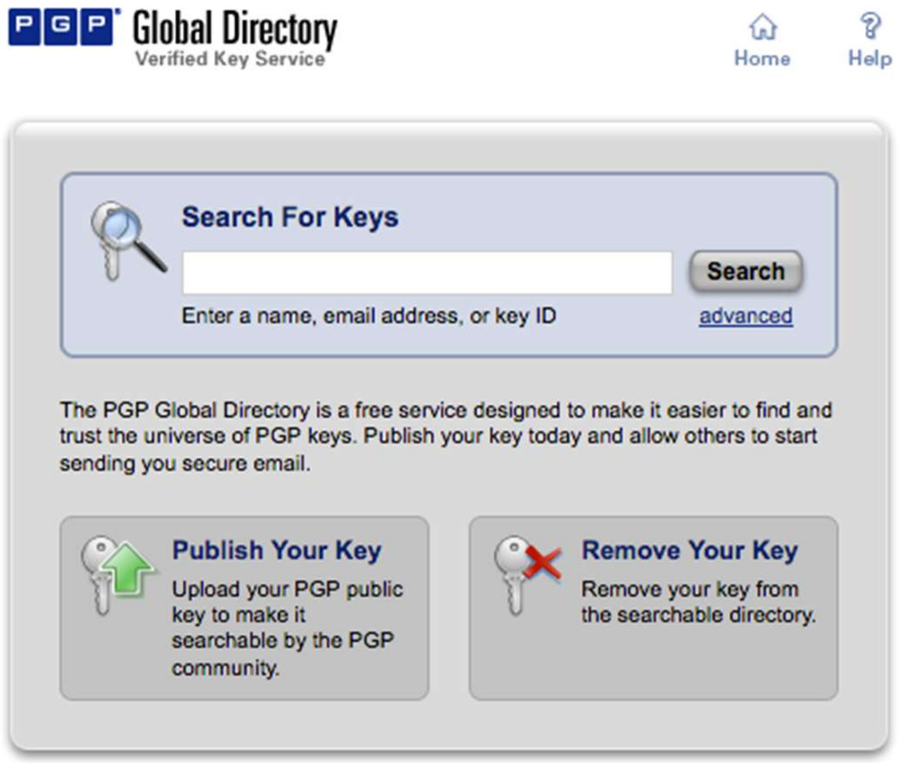
- Every PGP certificates contains a self signature
- a single PGP certificate can contain multiple signatures.
  - Several or many people may sign the key/ identification pair to attest to their own assurance that the public key definitely belongs to the specified owner

PGP model vs. CA/X.509 model

- you can create your own PGP certificate; you must request and be issued an X.509 certificate from a Certification Authority
- X.509 certificates natively support only a single name for the key's owner
- X.509 certificates support only a single digital signature to attest to the key's validity

PGP model vs. CA/X.509 model

- Public keys are maintained in an open server
  - e.g. https://keyserver.pgp.com/vkd/GetWelcomeScreen.event



The PGP Global Directory is a free service designed to make it easier to find and trust the universe of PGP keys. Publish your key today and allow others to start sending you secure email.

Publish Your Key
Upload your PGP public key to make it searchable by the PGP community.

Remove Your Key
Remove your key from the searchable directory.

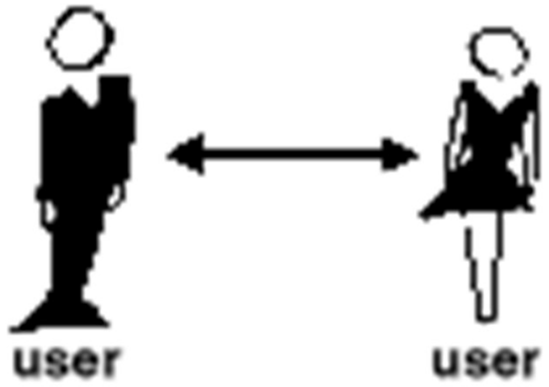
Threat: Forgery & Tampering??

Trust Models

- Direct Trust
- Hierachical Trust
- A Web of Trust

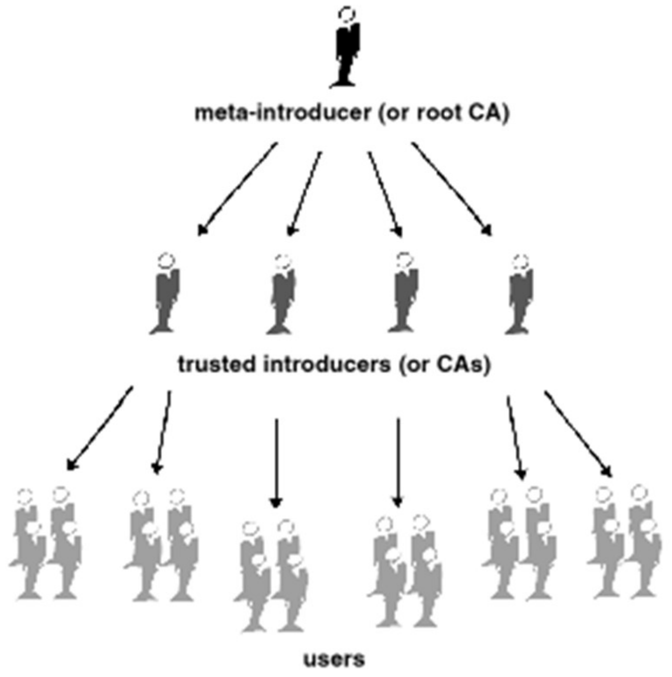
Direct Trust

- A user trusts that a key is valid because he/she knows where it came from.
- In PGP, a user who validates keys herself and never sets another certificate to be a trusted introducer is using direct trust.



Hierachical trust

- In a hierarchical system, there are a number of "root" certificates from which trust extends.
- Root certificates can certify intermediates CA
  - i.e. deligate authority to sign and verify public keys/other certificates down the chain
- Kind of a big trust "tree."
  - The "leaf" certificate's validity is verified by tracing backward from its certifier, to other certifiers, until a directly trusted root certificate is found.



Web of Trust

- A web of trust encompasses both of the other models
- adds the notion that trust is in the eye of the beholder (which is the real-world view) and the idea that more information is better
- a cumulative trust model
  - PGP uses digital signatures as its form of introduction. When any user signs another's key, he or she becomes an introducer of that key. As this process goes on, it establishes a web of trust.
- any user can act as a certifying authority.
  - Any PGP user can validate another PGP user's public key certificate.
  - However, such a certificate is only valid to another user if the relying party recognizes the validator as a trusted introducer.

Level of Trusts

- On each user’s PGP keyring, there is info
  - whether or not the user considers a particular key to be valid
  - the **level of trust** the user places on the key that the key's owner can serve as certifier of others' keys
  - Reputation system
- 3 Types of trust levels
  - Complete trust
  - Marginal trust
  - No trust (or untrusted)

Level of Trusts

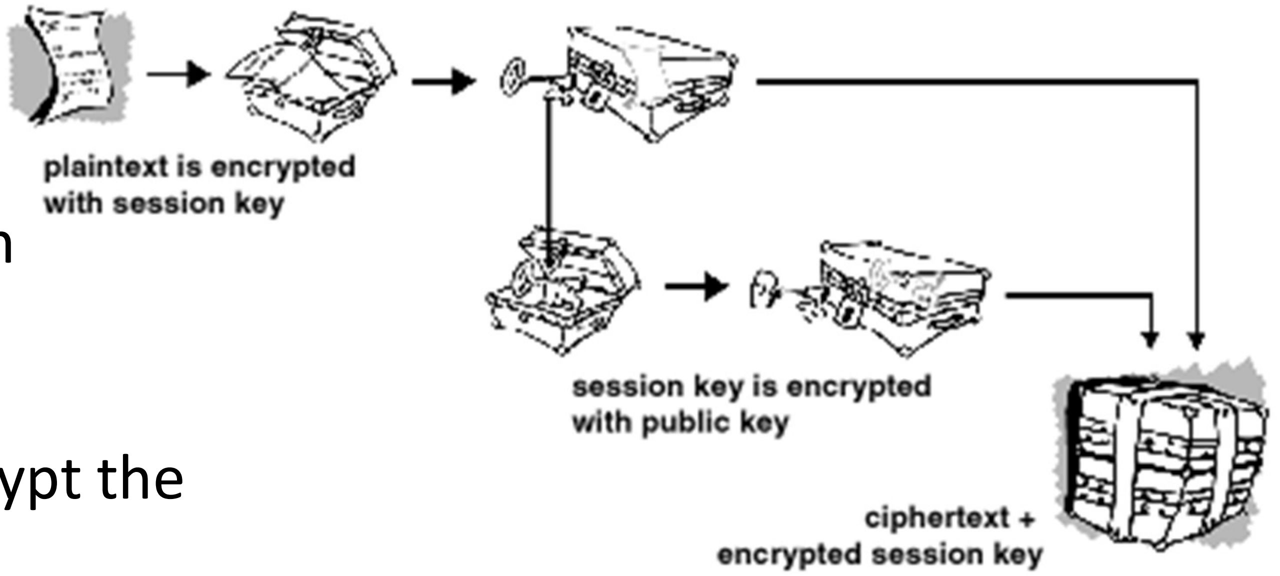
- To define another's key as a trusted introducer, you start with:
  - a valid key, one that is either signed by you or signed by another trusted introducer
  - Set the level of trust you feel the key's owner is entitled.
- A trusted introducer becomes a Certification Authority.
- If S/he signs another's key, it appears as **Valid** on your keyring.
- PGP requires one Completely trusted signature or two Marginally trusted signatures to establish a key as valid.

Key revocation

- Anyone who has signed a certificate can revoke his or her signature on the certificate (using the same private key that created the signature)
  - Suspect that the certificate's corresponding private key has been compromised
  - The certificates have expired

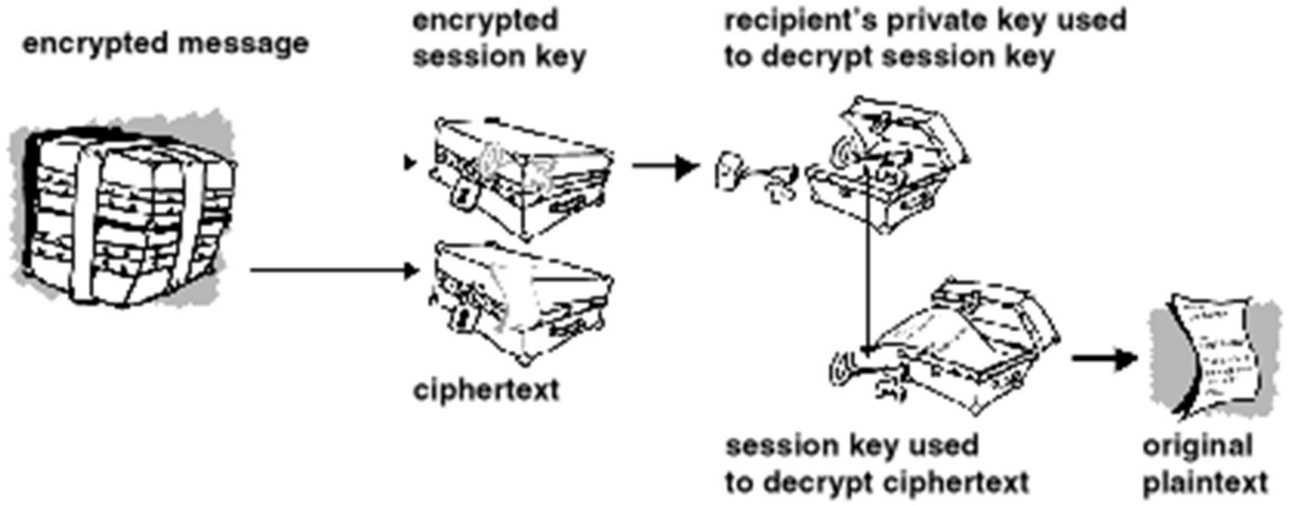
Read: 15 Reasons not to start using PGP: https://secushare.org/PGP

PGP – Encryption


plaintext is encrypted with session key
session key is encrypted with public key
ciphertext + encrypted session key

- PGP is a hybrid cryptosystem
- Compresses message
- Generates a session key
- Uses the session key to encrypt the message
- Uses the public key to encrypt the session key
- Sends both the encrypted message and encrypted session key

PGP – Decryption process

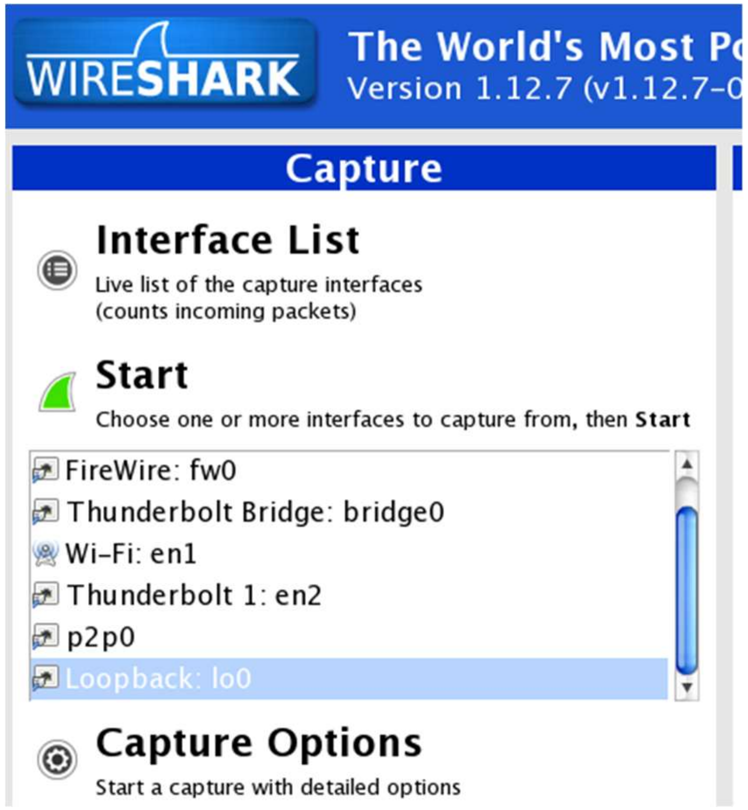

encrypted message
encrypted session key
ciphertext
recipient's private key used to decrypt session key
session key used to decrypt ciphertext
original plaintext

- Decrypt the session key with the private key
- Use the decrypted session key to decrypt the message

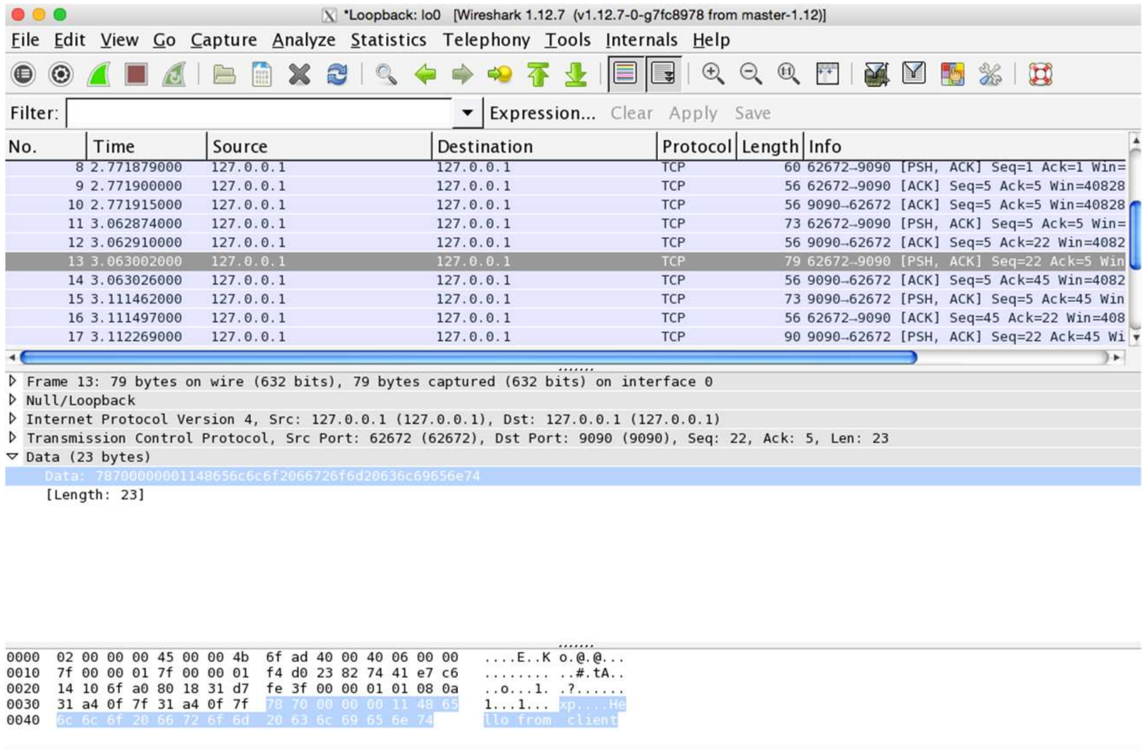
Lab #3 – Obl.

- Client/Server using plaintext communication (CipherLab.zip)
- Client/Server using SSL and Certificates (install keys in the keystore and truststore)
- Client/Server using DES encryption (implement DES in CipherLab.zip)
- Use Wireshark to capture data on localhost (127.0.01)
  - Capture the traffic between the client and server for each case
  - Document the actual data packets in each case.
- Submit the modified programs with the wireshark report

Download wireshark



WIRESHARK
The World's Most Po
Version 1.12.7 (v1.12.7-0
Capture
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
FireWire: fw0
Thunderbolt Bridge: bridge0
Wi-Fi: en1
Thunderbolt 1: en2
p2p0
Loopback: lo0
Capture Options
Start a capture with detailed options



*Loopback: lo0 [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.     Time         Source      Destination  Protocol Length Info
      8 2.771879000  127.0.0.1   127.0.0.1    TCP      60 62672→9090 [PSH, ACK] Seq=1 Ack=1 Win=
      9 2.771900000  127.0.0.1   127.0.0.1    TCP      56 62672→9090 [ACK] Seq=5 Ack=5 Win=40828
     10 2.771915000  127.0.0.1   127.0.0.1    TCP      56 9090→62672 [ACK] Seq=5 Ack=5 Win=40828
     11 3.062874000  127.0.0.1   127.0.0.1    TCP      73 62672→9090 [PSH, ACK] Seq=5 Ack=5 Win=
     12 3.062910000  127.0.0.1   127.0.0.1    TCP      56 9090→62672 [ACK] Seq=5 Ack=22 Win=4082
     13 3.063002000  127.0.0.1   127.0.0.1    TCP      79 62672→9090 [PSH, ACK] Seq=22 Ack=5 Win
     14 3.063026000  127.0.0.1   127.0.0.1    TCP      56 9090→62672 [ACK] Seq=5 Ack=45 Win=4082
     15 3.111462000  127.0.0.1   127.0.0.1    TCP      73 9090→62672 [PSH, ACK] Seq=5 Ack=45 Win
     16 3.111497000  127.0.0.1   127.0.0.1    TCP      56 62672→9090 [ACK] Seq=45 Ack=22 Win=408
     17 3.112269000  127.0.0.1   127.0.0.1    TCP      90 9090→62672 [PSH, ACK] Seq=22 Ack=45 Wi

▷ Frame 13: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0
▷ Null/Loopback
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ Transmission Control Protocol, Src Port: 62672 (62672), Dst Port: 9090 (9090), Seq: 22, Ack: 5, Len: 23
▽ Data (23 bytes)
    Data: 78700000001148656c6c6f2066726f6d20636c69656e74
    [Length: 23]

0000  02 00 00 00 45 00 00 4b  6f ad 40 00 40 06 00 00   ....E..K o.@.@...
0010  7f 00 00 01 7f 00 00 01  f4 d0 23 82 74 41 e7 c6   ........ ..#.tA..
0020  14 10 6f a0 80 18 31 d7  fe 3f 00 00 01 01 08 0a   ..o...1. .?......
0030  31 a4 0f 7f 31 a4 0f 7f  78 70 00 00 00 11 48 65   1...1... xp....He
0040  6c 6c 6f 20 66 72 6f 6d  20 63 6c 69 65 6e 74      llo from  client
```

Quiz – 1min

1. What does PKI achieve in public key systems? (multiple answers)
   a) Binds an identity to a public key
   b) Verifies users transactions
   c) Provides key management such as key distribution, validation and revocation
2. How is PGP certificate different from X.509 certificate model? (multiple answers)
   a) You can create your own certificate
   b) You need a single authority to sign your certificate
   c) A PGP certificate can contain multiple signatures
3. PGP uses
   a) Symmetric and asymmetric cryptosystems
   b) Only symmetric
   c) Only asymmetric

Resources

- Lecture Notes 14: Public Key Infrastructure by Ron Rivest
- An Introduction to Distributed Security Concepts and Public Key Infrastructure (PKI) by Mary Thompson
- *www.cs.bu.edu/~itkis/558/slides/PKI.ppt*
- http://cups.cs.cmu.edu/courses/ups-fa11/slides/Ur_Maass_ups_oct_20.pdf
- https://medium.com/@peterryszkiewicz/secure-cryptocurrency-seeds-fdd99f39df7e
- Manico & Detlefsen. Iron-Clad Java: Buiding Secure Web Applications
- https://users.ece.cmu.edu/~adrian/630-f04/PGP-intro.html#p10
- *https://swedbank.ee/download/gateway/PKI-in-nutshell.pdf*