Lecture
Digital signature & PKI

By Rachel yuan

2019.11

1

RSA exercise

1. Select primes: $p=17$ & $q=11$
2. Calculate $n = pq = 17 * 11 = 187$
3. Calculate $\phi(n) = (p-1)(q-1) = 16*10 = 160$
4. Select e: $\gcd(e,160)=1$; choose $e=7$
5. Determine d: $de=1 \bmod 160$ and $d < 160$

6. Publish public key PU={7,187}
7. Keep secret private key PR={d,187}

2

RSA exercise

- sample RSA encryption/decryption is:
- given message M = 88 (88<187)
- Encryption:

- Decryption:

3

RSA exercise

1. Select primes: $p=17$ & $q=11$
2. Calculate $n = pq = 17 * 11 = 187$
3. Calculate $\phi(n) = (p-1)(q-1) = 16*10 = 160$
4. Select e: $\gcd(e,160)=1$; choose $e=7$
5. Determine d: $de=1 \bmod 160$ and $d < 160$
   Value is d=23 since $23*7 = 161 = 10*160+1$
6. Publish public key PU={7,187}
7. Keep secret private key PR={23,187}

4

RSA exercise

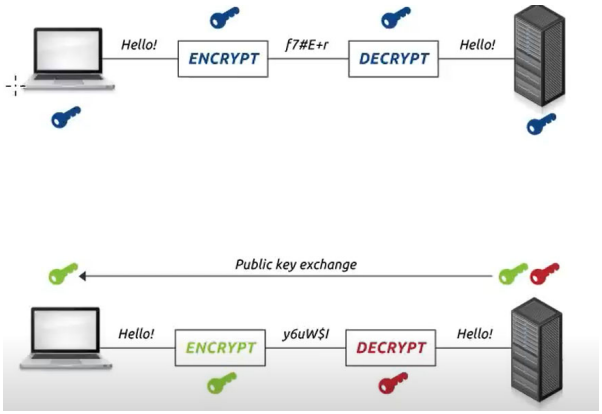
- sample RSA encryption/decryption is:
- given message $M = 88$ (nb. $88<187$)
- Encryption with <7, 187>:

  $C = 88^7 \bmod 187 = 11$
- Decryption with <23, 187>:

  $M = 11^{23} \bmod 187 = 88$

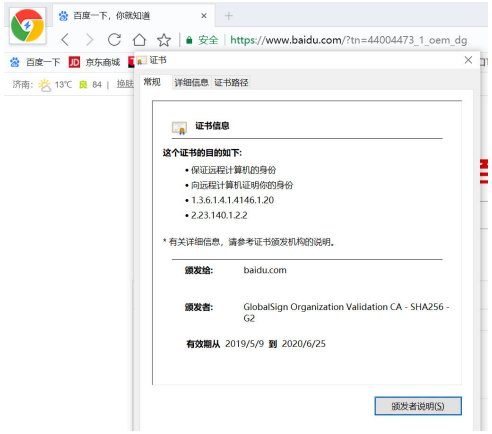
5

Review



6

Using a certificate for SSL communications



7

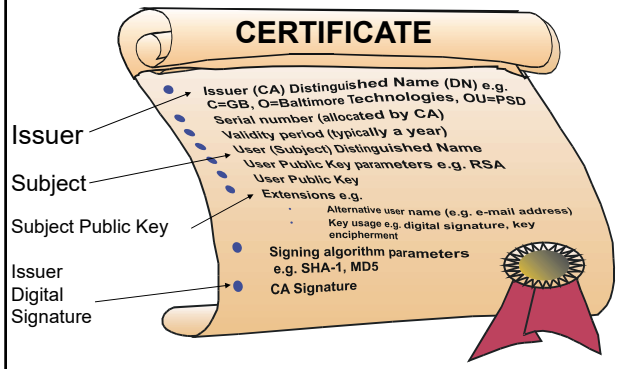
So, what is a digital certificate?

electronic credentials used to assert the online identities of individuals, computers, and other entities on a network

similar to ID cards (passports, driver licenses) contain public key & identify of owner

8

Digital Certificate



9

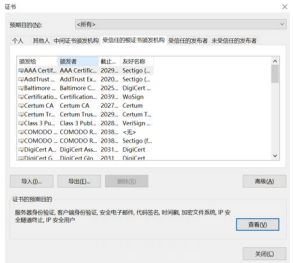
X.509 v3格式

内容	说明
版本V	X．509版本号
证书序列号	用于标识证书
算法标识符	签名证书的算法标识符
参数	算法规定的参数
颁发者	证书颁发者的名称及标识符(X.500)
起始时间	证书的有效期
终止时间	证书的有效期
持证者	证书持有者的姓名及标识符
算法	证书的公钥算法
参数	证书的公钥参数
持证书人公钥	证书的公钥
扩展部分	CA对该证书的附加信息，如密钥的用途
数字签名	证书所有数据经H运行后CA用私钥签名

10

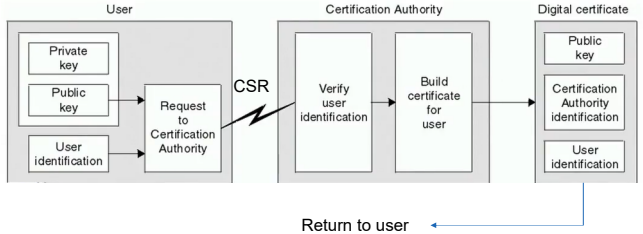
Certificate issuer

- Issued by a certificate authority (CA) that validate the identity of the certificate holder.
- Self-signed certificate



11

How to get a digital cerfiticate?



CSR：Certificate signing request

12

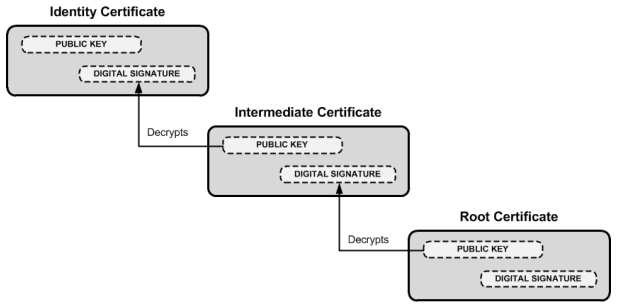
Generate CSR using openssl

- openssl genrsa -aes128 -out fd.key 2048
- openssl rsa -text -in fd.key
- openssl rsa -in fd.key -pubout -out fd-public.key

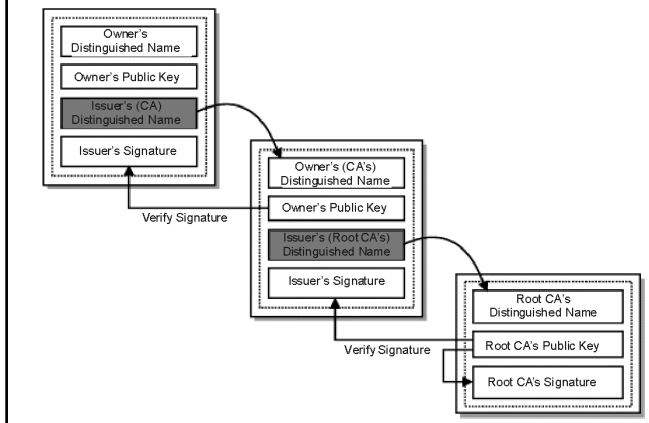
- openssl req -new -key fd.key -out fd.csr
- openssl req -text -in fd.csr -noout

13

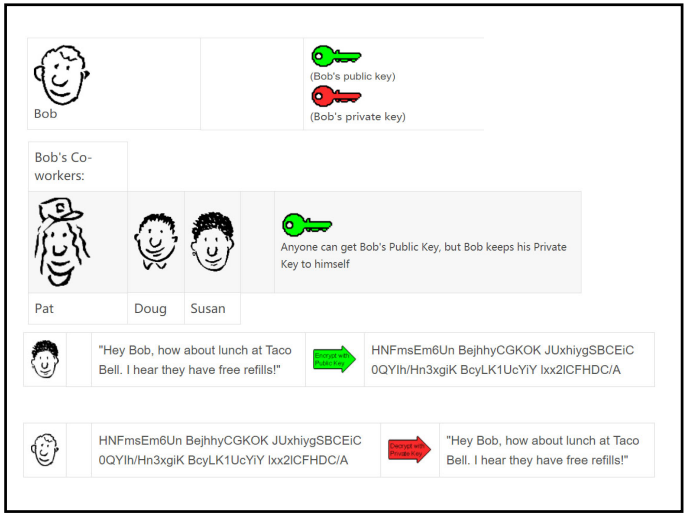
Chain of trust



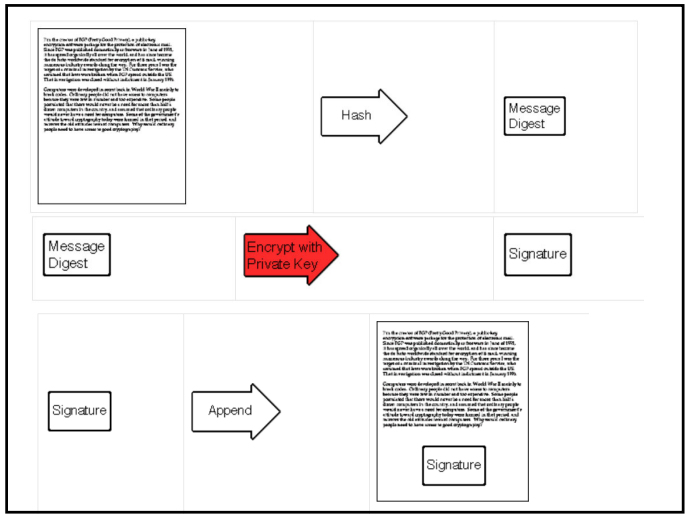
14



15

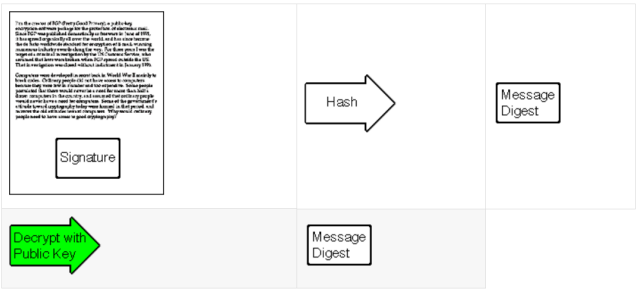


16



17

Digital signature verification



漫说数字签名： https://www.jianshu.com/p/1499b69e9c89

18

Openssl数字签名与验证实例

- openssl genrsa -out key.pem 2048
- openssl rsa -in key.pem -pubout -out key_pub.pem
- dd if=/dev/zero of=data.bin bs=1 count=16
- hexdump -Cv data.bin
- openssl dgst -sha256 -binary -out data.bin.sha256 data.bin
- openssl dgst -sha256 -out data.bin.signature -sign key.pem data.bin
- openssl dgst -sha256 -verify key_pub.pem -signature data.bin.signature data.bin

19

数字签名python实验

- >>> from Crypto.Hash import SHA256
- >>> from Crypto.PublicKey import RSA
- >>> from Crypto import Random

- >>> random_generator = Random.new().read
- >>> key = RSA.generate(1024,random_generator)

- >>> text='meetatnoon'.encode()
- >>> hash = SHA256.new(text).digest()
- >>> signature = key.sign(hash,'')

20

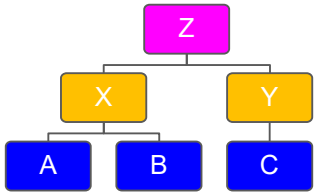
- >>> text_ver='meetatnoon'.encode()
- >>> hash_ver=SHA256.new(text_ver).digest()
- >>> public_key=key.publickey()
- >>> public_key.verify(hash_ver,signature)

21

假设数字证书含有用户身份（ID），用户公钥（PU），时间戳（T）以及权威机构的数字签名。一个系统的信任结构如下图所示，下级节点信任上级节点，如A、B信任X，X信任Z等。

如果使用‖表示连接符，E(k,p)和D(k,c)分别表示加密和解密算法，H()表示hash函数，PU表示公钥，PR表示私钥，

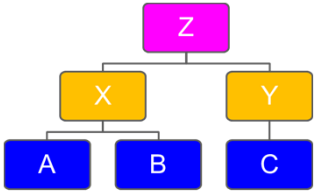
1. 那么请写一个表示用户A的证书（C_A）的公式。



22

$$C_A = ID_A||PU_A||T||E(PR_X, H(ID_A||PU_A||T))$$

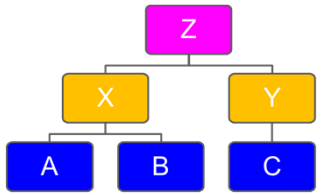
2. 是谁给Z签名的？Who signs Cz?



23

3. 假设A的证书为C_A，B的证书为C_B，X的证书为C_X，请问A如何证实C_B的真实性。

4. 同样的，C的证书为C_C，Y的证书为C_Y，请问A如何证实(verify)C_C的真实性。



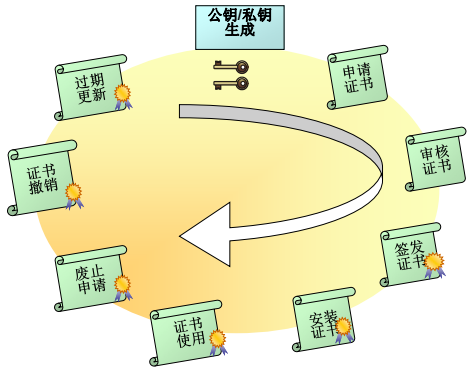
24

证书的管理——证书的生命周期

- 证书从产生到撤销具有一定的生命周期，从创建到销毁总共要经历五个阶段：
- (1) 证书申请
- (2) 证书生成
- (3) 证书存储
- (4) 证书发布（证书库）
- (5) 证书撤销

25

证书的生命周期



26

证书的管理——证书的撤销

- CA签发的证书捆绑了用户的身份和公钥，在生命周期里都是有效的。
- 但在现实环境中，由于这些原因包括：用户身份的改变、对密钥的怀疑（丢失或泄露）、用户工作的变动、认为CA证书已泄露等。
- 必须存在一种机制撤销这种认可。

27

证书的管理——证书的撤销

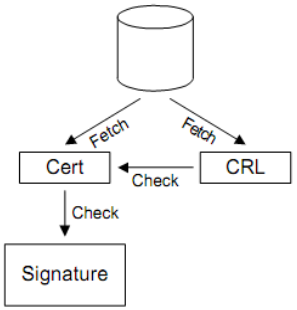
- 证书撤销最常用的方式是使用证书废除列表（CRL-Certificate Revocation List），CRL是一种包含了撤销的证书列表的签名数据结构。
- CA会定期地发布CRL，从几个小时到几个星期不等。不管CRL中是否含有新的撤销信息，都会发布一个新的CRL。

28

X.509 Certificate Usage Model

Relying party wants to verify a signature

- Fetch certificate
- Fetch certificate revocation list (CRL)
- Check certificate against CRL
- Check signature using certificate

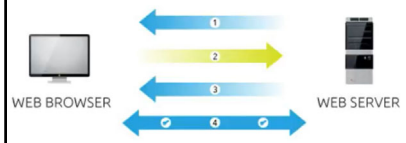


29

PKI的概念

- PKI(Pubic Key Infrastructure,公钥基础设置)
  - 生成、管理、存储、分发和撤销基于公开密码的公钥证书所需要的硬件、软件、人员、策略和规程的总和。
  - A **public key infrastructure** (**PKI**) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.
- PKI已广泛用于保障电子商务和电子政务的安全。

30



1. **Server** sends a copy of its asymmetric public key.
2. **Browser** creates a symmetric session key and encrypts it with the server's asymmetric public key.
3. **Server** decrypts the asymmetric public key with its asymmetric private key to get the symmetric session key.
4. **Server** and **Browser** now encrypt and decrypt all transmitted data with the symmetric session key. This allows for a secure channel because only the browser and the server know the symmetric session key, and the session key is only used for that session. If the browser was to connect to the same server the next day, a new session key would be created.

31

SSL

- SSL (secure socket layer)
- http --> https
- netscape IE, SSL, --> microsoft
- 2014.4 heartbleed (openssl vulnerability)
- position
- SSL can be used by any application, not only http

2019/12/12

32

SSL provides

- verification of identity of server
- message exchange with
  - confidentiality
  - integrity
  - freshness

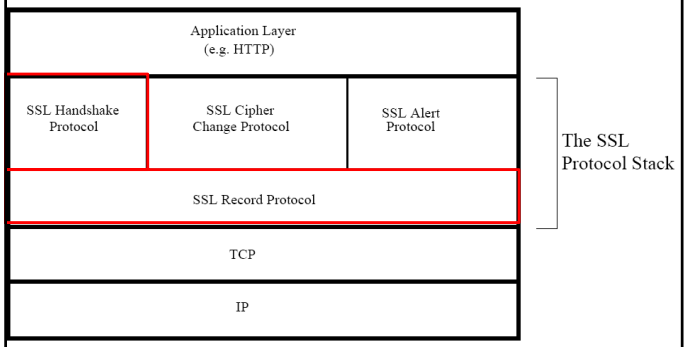
33

SSL suite

- SSL suite
- handshake protocol
  - authenticate server
  - negotiate various keys
- record protocol
  - compress
  - authenticate
  - Encryption
- Two less important protocols are: SSL Cipher Change Protocol and SSL Alert Protocol.

34

SSL Protocol Stack



35

handshake protocol

- authenticate server (optional: authenticate client)
- method:
- using public key authentication
- PKI

36

- C-->S helloserver, SSL version, preferences(algorithms i supported), randomC
- S-->C helloclient, SSL version, choices, randomS
- S-->C certificate, verification chain
- C: verification server
- C: {premaster key, randomC, randomS} --> session key
- C-->S: Es(premaster key)
- S: premaster key, randomC, randomS --> session key

37