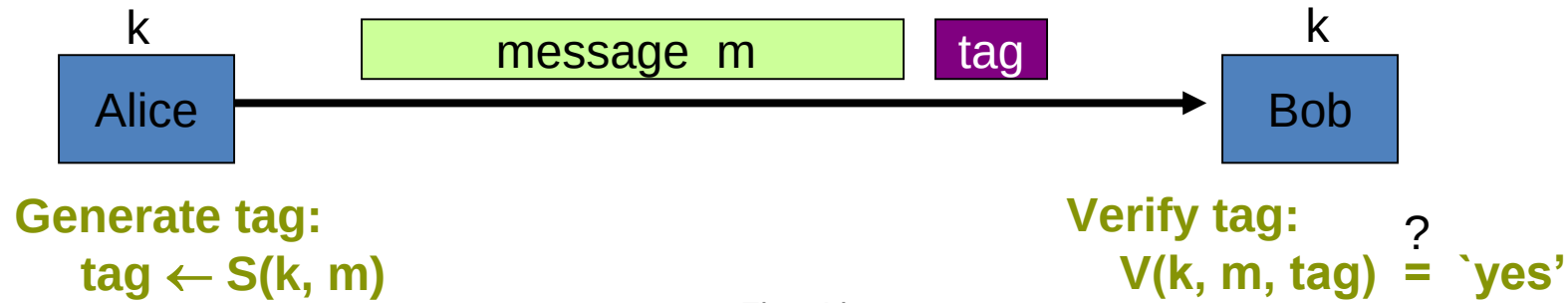
Cryptographic checksum

- A function that produces a checksum (smaller than message)
  - A digest function using a key only known to sender and recipient
  - Ensure both integrity and authenticity
  - Used for code-tamper protection and message-integrity protection in transit
  - The choices of hashing algorithms are Message Digest (MD) by Ron Rivest of RSA Labs or Secure Hash Algorithm (SHA) led by US government
  - MD: MD4, MD5; SHA: SHA1, SHA2, SHA3



Constructing cryptographic checksum

- HMAC (Hash Message Authentication Code)
- Problem to solve: how to merge key with hash

H: hash function; k: key; m: message;

example of H: SHA-2-256 ; output is 256 bits

- Building a MAC out of a hash function:

Paddings to make fixed-size block

$$\mathrm{HMAC}(K, m) = \mathrm{H}\left((K' \oplus opad) \parallel \mathrm{H}\left((K' \oplus ipad) \parallel m\right)\right)$$

$$K' = \begin{cases} \mathrm{H}(K) & K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

https://en.wikipedia.org/wiki/HMAC

Cryptoanalysis on crypto checksums

- MD5 broken by Xiaoyun Wang et al. at 2004
  - Collisions of full MD5 less than 1 hour on IBM p690 cluster
- SHA-1 theoretically broken by Xiaoyun Wang et al. at 2005
  - 2^{69} steps (<< 2^{80} rounds) to find a collision
- SHA-1 practically broken by Google at 2017
  - First collision found with 6500 CPU years and 100 GPU years

Input

Output

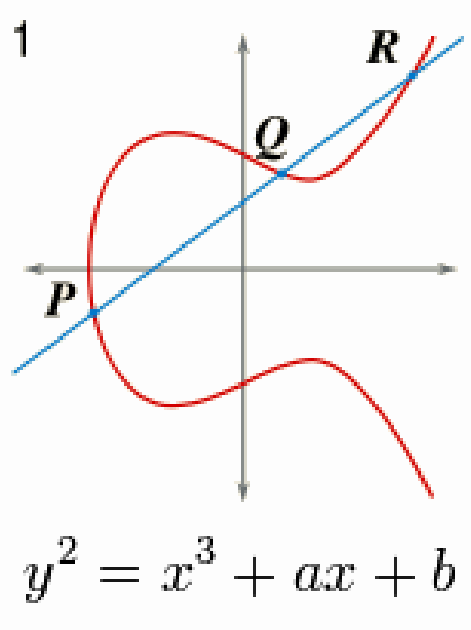
Algorithm	Maximum Message Size (bits)	Block Size (bits)	Rounds	Message Digest Size (bits)
MD5	2^{64}	512	64	128
SHA-1	2^{64}	512	80	160
SHA-2-224	2^{64}	512	64	224
SHA-2-256	2^{64}	512	64	256
SHA-2-384	2^{128}	1024	80	384
SHA-2-512	2^{128}	1024	80	512
SHA-3-256	unlimited	1088	24	256
SHA-3-512	unlimited	576	24	512

Current Secure Hash Standard Properties

https://shattered.io/static/shattered.pdf

Elliptic Curve Cryptography

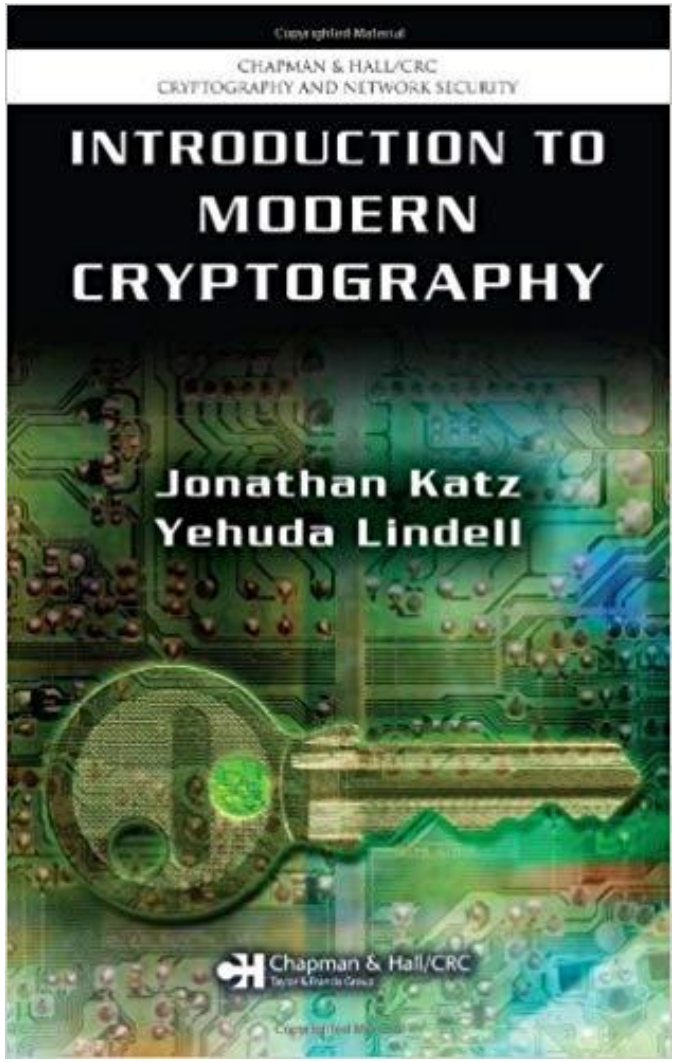
- RSA algorithm is patented
- Alternative asymmetric cryptography: Elliptic Curve Cryptography (ECC)
  - The general algorithm is in the public domain
  - ECC can provide similar security to RSA using a shorter key length
  - Mainly used for key exchange and digital signature
  - Satisfying group properties in mathematics (closure, associativity, identify element, inverse)
  - Public key: Q = d*P (d is private key, P is a point on curve)
  - Encryption: $C_1 = k*P$, $C_2 = M + k*Q$
  - Decryption: $M = C_2 - d*C_1$

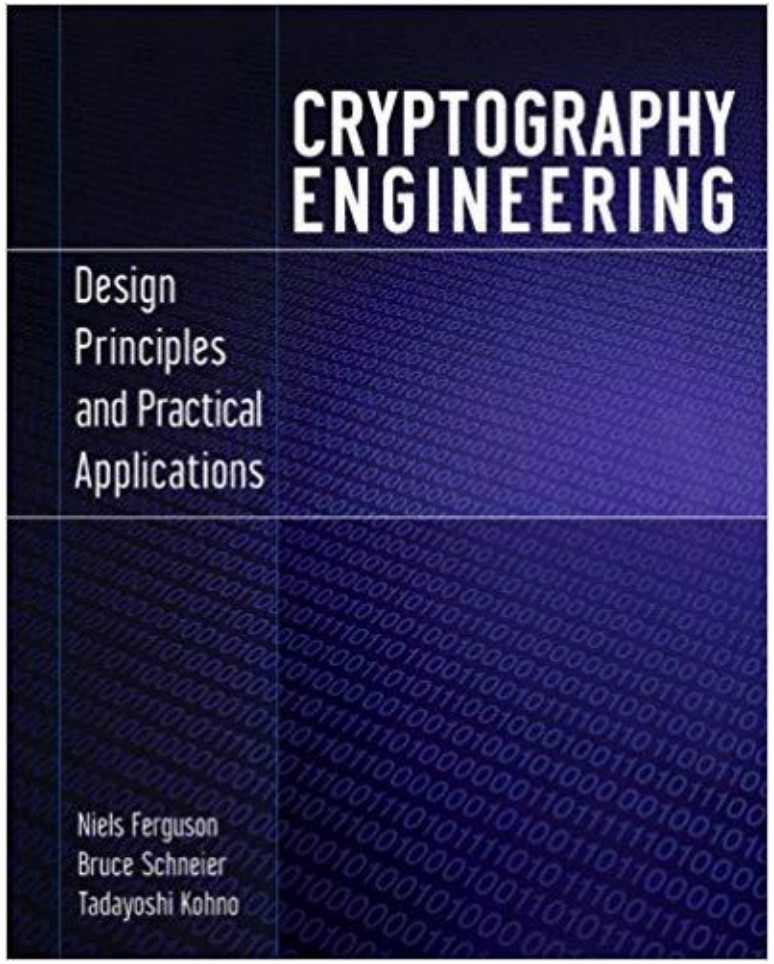


Simple elliptic curve

https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/
https://bithin.wordpress.com/2012/02/22/simple-explanation-for-elliptic-curve-cryptography-ecc/

To learn more about cryptography





Public Key Infrastructure (PKI)

EECS 195

Spring 2019

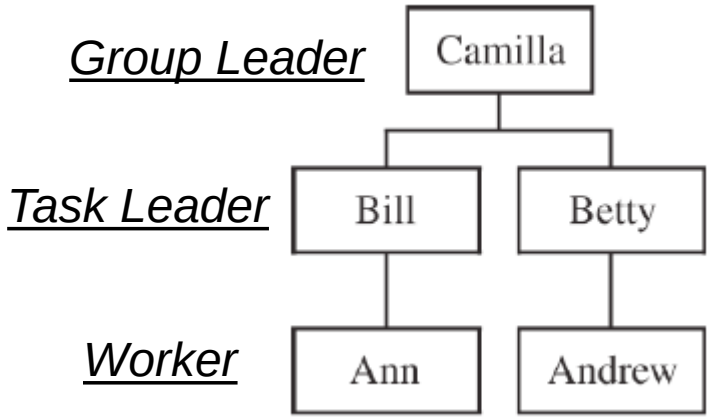
Zhou Li

Summary of cryptographic basics

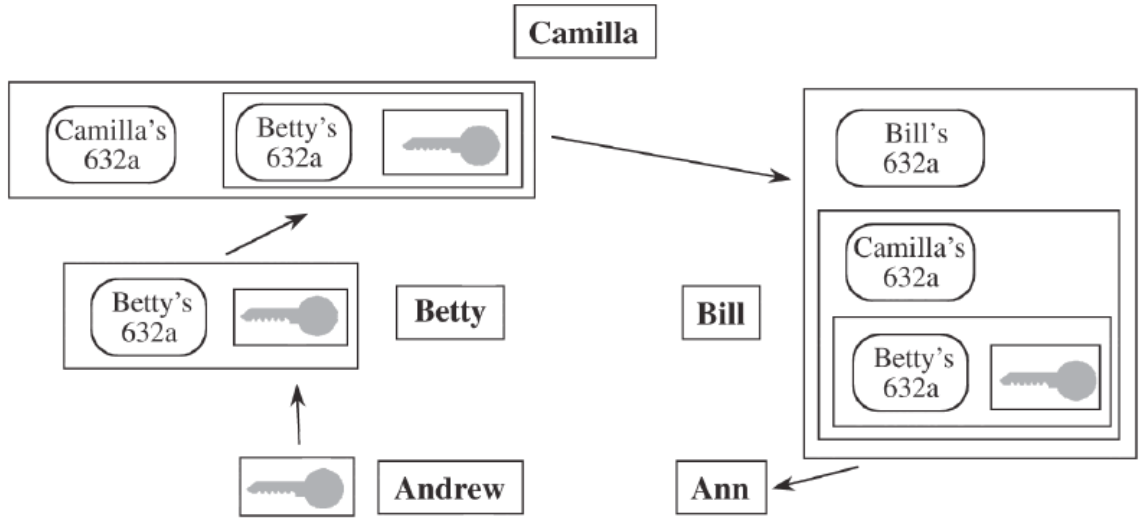
- Goal: secured message transmission
- What can be achieved?
  - Message confidentiality: symmetric encryption
  - Key management: asymmetric encryption
  - Message integrity: cryptographic checksum
- What's still missing?
  - Trust of sender/receiver identity!
    - E.g., given a public key of Microsoft, how do you know it's from Microsoft?
  - A trusted 3rd-party can "vouch for" is needed

Trust based on respected individual

- Question: how can Ann verifies Andrew work in the same company?
- Solution: Ann asks Bill asks Camilla asks Betty and gets response
  - Key exchange between Ann and Andrew: Betty attaches a 632a statement ("I know Andrew") to a Andrew's key and pass on



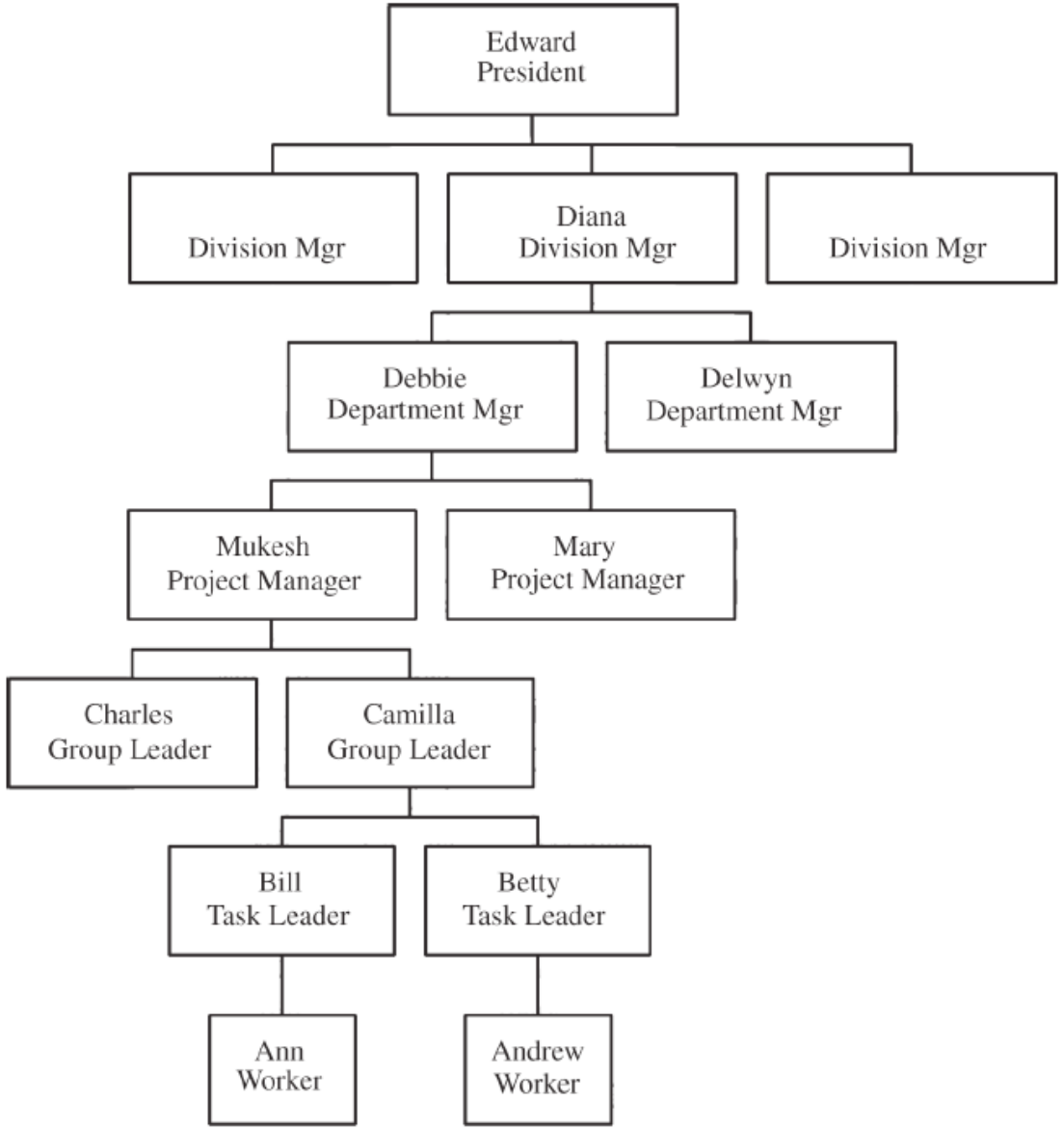
Org chart



Passing the key

Scale it up

- Problem: how to scale up the procedure for a large org (100K+ people)?
  - President: Edward
  - Division Manager: Diana, …
  - Department Manager: Delwyn, …
  - Project Manager: Mukesh, …
  - Group Leader: Camilla, …
  - Task Leader: Bill, …
  - Worker: Andrew, …



Trust based on respected individual (cond.)

- Problem 1: what if one person is not available sometime?
  - Andrew asks for his complete chain of 632a forms from president to him
  - Andrew gets signatures any time his superiors are available
  - Andrew gives a copy of 632a forms and his key to Ann
- Problem 2: what if one person (e.g., president) is never available?
  - President (Edward): "I attest the identity of my division manager (Diana) and I trust he/she to attest her subordinates"
  - Diana copies delegation letters
  - Andrew and Ann compare package of letters
  - Key exchange can happen when president is the same

Digital Certificate and CA

- Digital Certificate: trustable identity bounded with public key
- CA (Certificate Authority): trusted 3rd-party service certifying binding

To create Diana's certificate:

Diana creates and delivers to Edward:

Name: Diana Position: Division Manager Public key: 17EF83CA ...

Edward adds:

Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Edward signs with his private key:

Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Which is Diana's certificate.

To create Delwyn's certificate:

Delwyn creates and delivers to Diana:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...

Diana adds:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA

Diana signs with her private key:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA

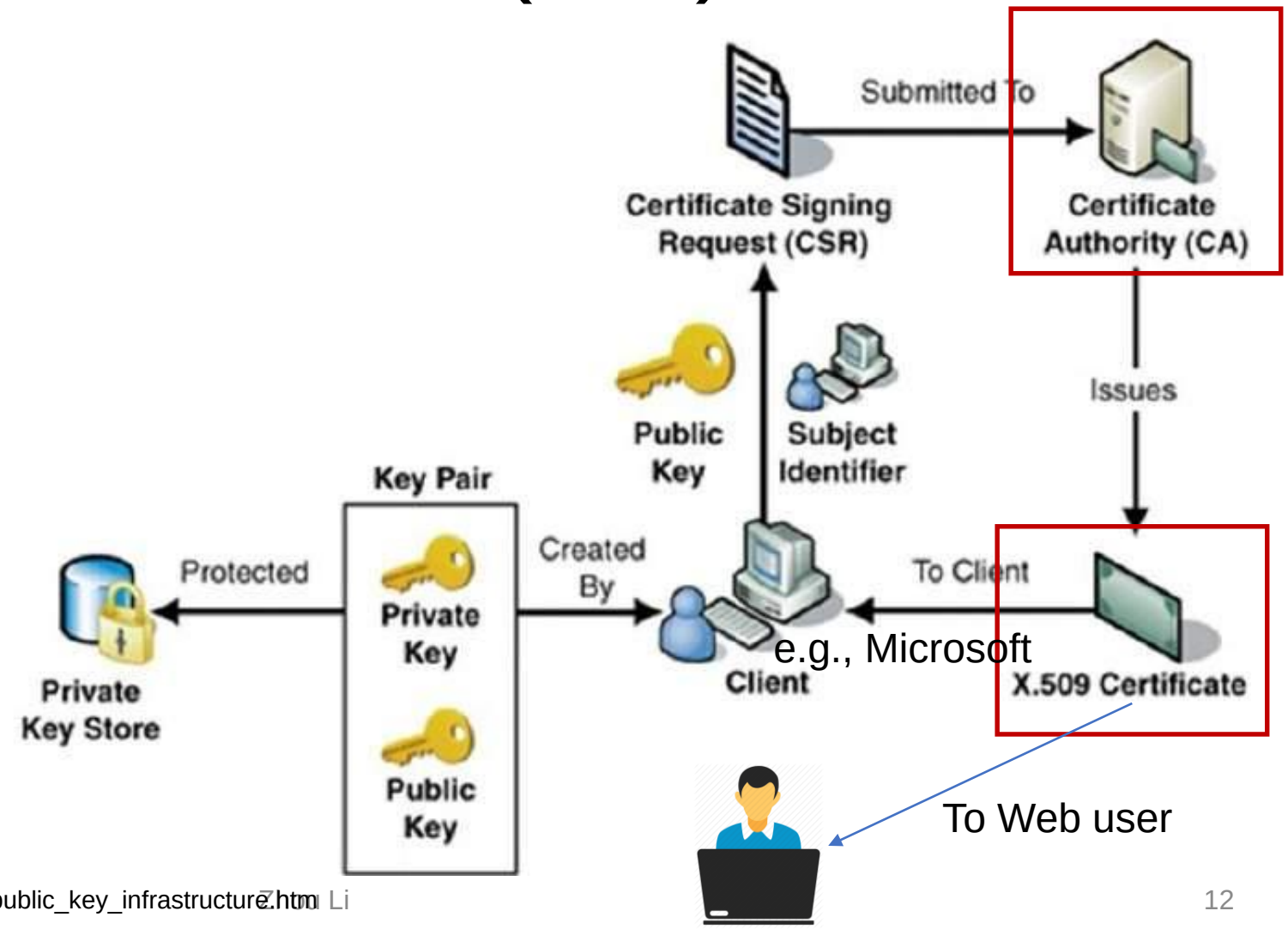
And appends her certificate:

Name: Delwyn Position: Dept Manager Public key: 3AB3882C ...	hash value 48CFA
Name: Diana Position: Division Manager Public key: 17EF83CA ...	hash value 128C4

Which is Delwyn's certificate.

Public Key Infrastructure (PKI)

- Tackle's the problem of certificate (public key and identity) creation and distribution



Implementations of PKI

- X.509 certificate
- Root and intermediate CAs
- Attacks against PKI

Getting certificates

- Let's get paypal's certificates

```
$ openssl s_client -showcerts -connect www.paypal.com:443 </dev/null
```

```
-----BEGIN CERTIFICATE-----
MIIHWTCCBkGgAwIBAgIQLNGVEFQ30N5KOSAFavbCfzANBgkqhkiG9w0BAQsFADB3
MQswCQYDVQQGEwJVUzEdMBsGA1UEChMUU3ltYW50ZWMgQ29ycG9yYXRpb24xHzAd
... (omitted) ...
GN/QMQ3a55rjwNQnA3s2WWuHGPaE/jMG17iiL2O/hUdIvLE9+wA+fWrey5//74xl
NeQitYiySDIepHGnng==
-----END CERTIFICATE-----
```

- Save the above data to paypal.pem, and use the following command decode it (see next slide)

```
$ openssl x509 -in paypal.pem -text -noout
```

Example of X.509 Certificate (1st Part)

The CA's identity (Symantec)

The owner of the certificate (paypal)

```
Certificate:
Data:
  Serial Number:
          2c:d1:95:10:54:37:d0:de:4a:39:20:05:6a:f6:c2:7f
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: C=US, O=Symantec Corporation, OU=Symantec Trust Network,
         CN=Symantec Class 3 EV SSL CA - G3
  Validity
     Not Before: Feb  2 00:00:00 2016 GMT
     Not After : Oct 30 23:59:59 2017 GMT
  Subject: 1.3.6.1.4.1.311.60.2.1.3=US/
          1.3.6.1.4.1.311.60.2.1.2=Delaware/
          businessCategory=Private Organization/
          serialNumber=3014267, C=US/
          postalCode=95131-2021, ST=California,
          L=San Jose/street=2211 N 1st St,
          O=PayPal, Inc., OU=CDN Support, CN=www.paypal.com
```

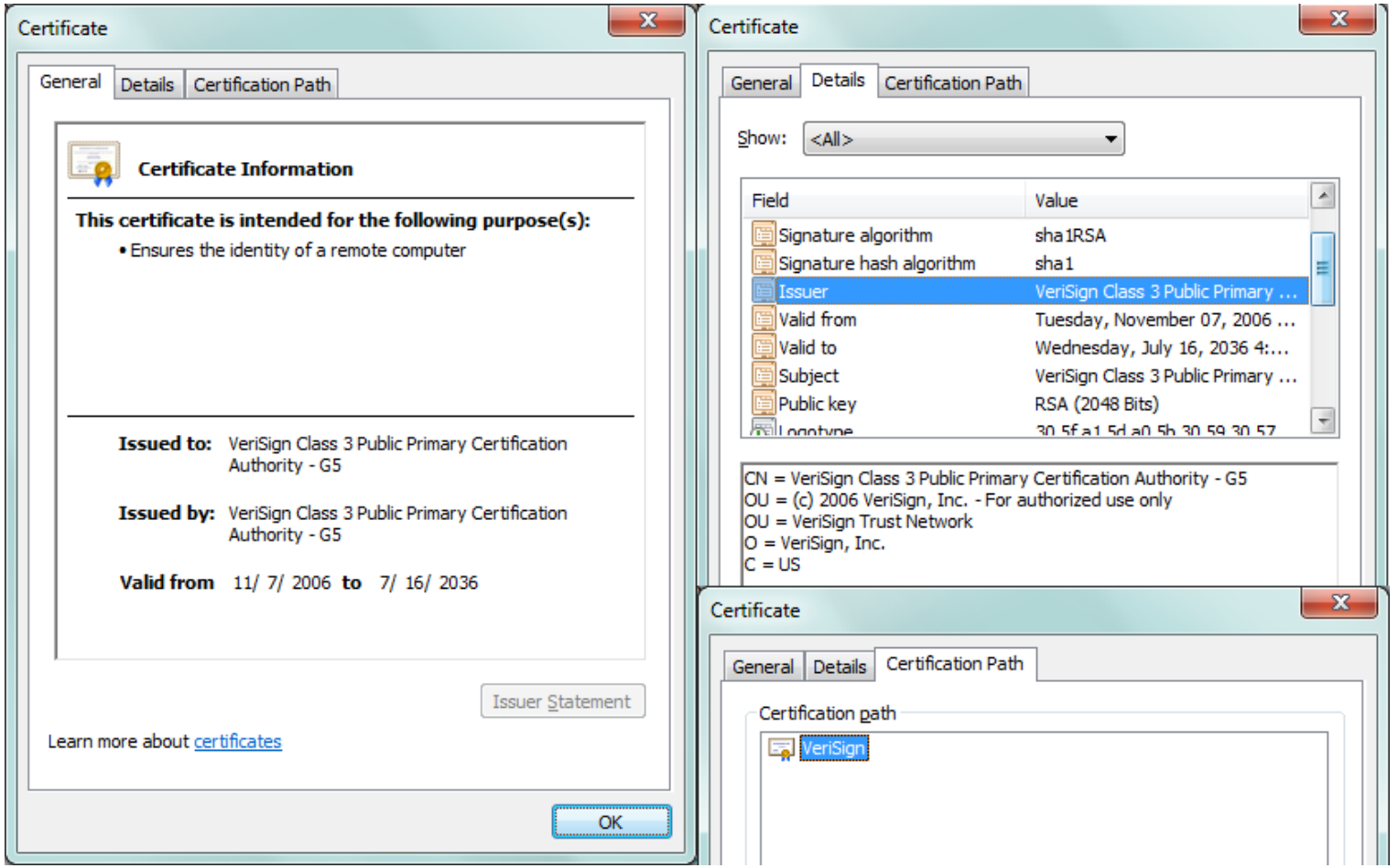
Example of X.509 Certificate (2nd Part)

Public key

CA's signature

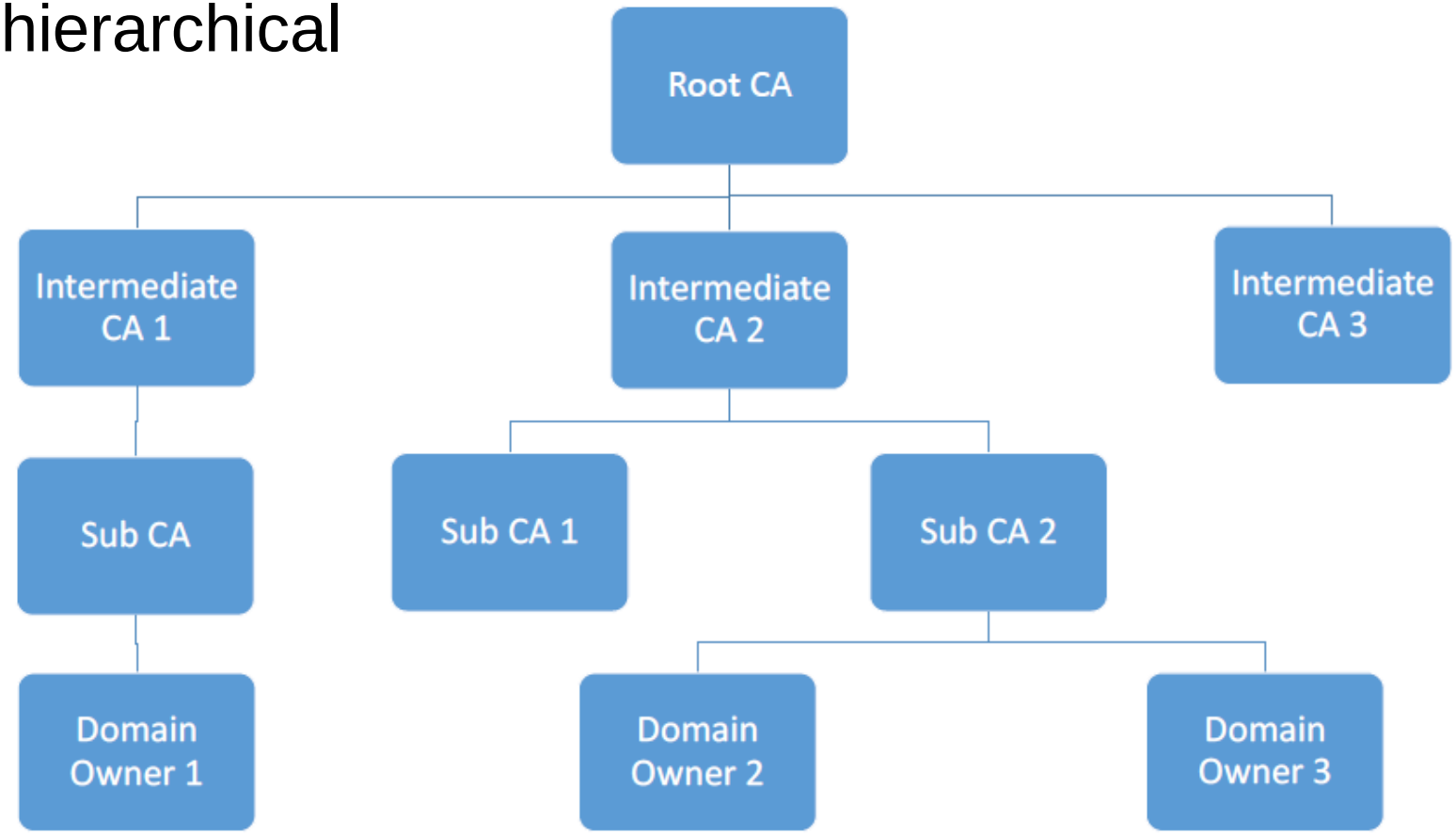
```
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
       Public-Key: (2048 bit)
       Modulus:
         00:da:43:c8:b3:a6:33:5d:83:c0:63:14:47:fd:6b:22:bd:
         bf:4e:a7:43:11:55:eb:20:8b:e4:61:13:ee:de:fe:c6:e2:
         ... (omitted) ...
         7a:15:00:c5:01:69:b5:10:16:a5:85:f8:fd:07:84:9a:c9:
       Exponent: 65537 (0x10001)
Signature Algorithm: sha256WithRSAEncryption
4b:a9:64:20:cc:77:0b:30:ab:69:50:d3:7f:de:dc:7c:e2:fb:93:84:fd:
78:a7:06:e8:14:03:99:c0:e4:4a:ef:c3:5d:15:2a:81:a1:b9:ff:dc:3a:
... (omitted) ...
fb:00:3e:7d:6a:de:cb:9f:ff:ef:8c:65:35:e4:22:b5:88:b2:48:32:1e:
```

X.509 certificate in browser



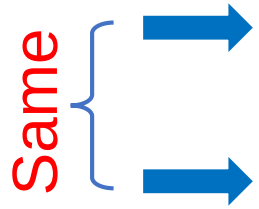
Root and Intermediate Certificate Authorities

There are many CAs in the real world, and they are organized in a hierarchical structure.



Root CAs and Self-Signed Certificate

- A root CA's public key is also stored in an X.509 certificate. It is self-signed.
- Self-signed: the entries for the issuer and the subject are identical.




```
Issuer: C=US, O=VeriSign, Inc., OU=VeriSign Trust Network,
    OU=(c) 2006 VeriSign, Inc. - For authorized use only,
    CN=VeriSign Class 3 Public Primary Certification Authority - G5
Subject: C=US, O=VeriSign, Inc., OU=VeriSign Trust Network,
    OU=(c) 2006 VeriSign, Inc. - For authorized use only,
    CN=VeriSign Class 3 Public Primary Certification Authority - G5
```

- How can they be trusted?
  - Public keys of root CAs are pre-installed in the OS, browsers and other software

Intermediate CAs and Chain of Trust

```
$ openssl s_client -showcerts -connect www.paypal.com:443

Certificate chain
 0 s: ... /CN=www.paypal.com
   i: ... /CN=Symantec Class 3 EV SSL CA - G3
-----BEGIN CERTIFICATE-----
MIIHWTCCBkGgAwIBAgIQLNGVEFQ30N5KOSAFavbCfzANBgkqhkiG9w0BAQsFADB3
...
-----END CERTIFICATE-----
 1 s: ... /CN=Symantec Class 3 EV SSL CA - G3
   i: ... /CN=VeriSign Class 3 Public Primary Certification
              Authority - G5
-----BEGIN CERTIFICATE-----
MIIFKzCCBBOgAwIBAgIQfuFKb2/v8tN/P61lTTratDANBgkqhkiG9w0BAQsFADCB
...
-----END CERTIFICATE-----
```

Paypal's certificate

Intermediate CA's certificate

B

A

A is used to verify B

Something else is needed to verify A (certificate from another intermediate CA or root CA)

Trusted CAs in the Real World

- Not all of the trusted CAs are present in all browsers.
- According to W3Techs in April 2017, Comodo takes most of the market share followed by IdenTrust, Symantec Group, GoDaddy Group, GlobalSign and DigiCert.
- The list of trusted CAs supported by browser can be found:
  - **For the Chrome browser:**
    - Settings -> Show advanced settings -> Manage Certificates
  - **For the Firefox browser:**
    - Edit -> Preferences -> Advanced -> Certificates -> View Certificates -> Certificate Manager -> Authorities

Attack on CA's Verification Process

- CA's job has two parts:
  - Verify the relationship between certificate applicant and the subject information inside the certificate
  - Put a digital signature on the certificate

- **Case study: Comodo Breach [March 2011]**
  - Popular root CA.
  - The approval process in Southern Europe was compromised.
  - Nine certificates were issued to seven domains and hence the attacker could provide false attestation.
  - One of the affected domain (a key domain for the Firefox browser): `addons.mozilla.org`

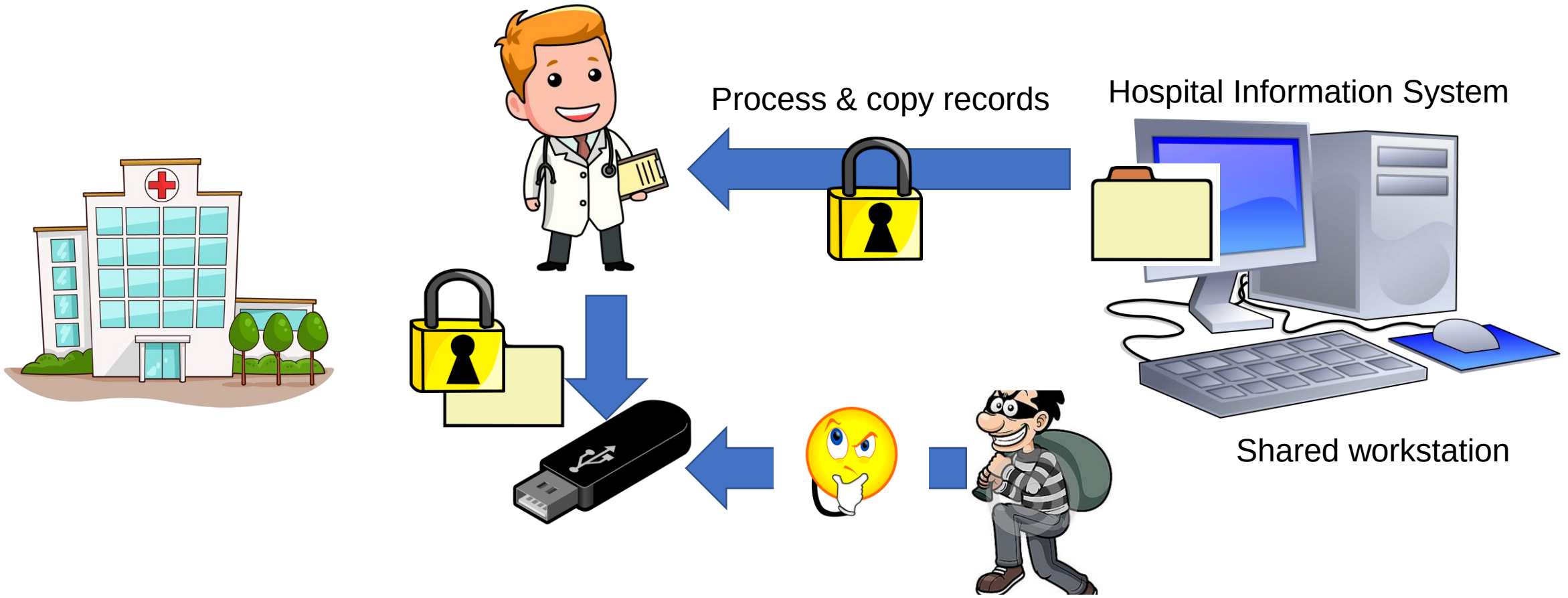
Attack on CA's Signing Process

- If the CA's private key is compromised, attackers can sign a certificate with any arbitrary data in the subject field.

- **Case Study: the DigiNotar Breach [June-July 2011]**
  - A top commercial CA
  - Attacker got DigiNotar's private key
  - 531 rogue certificates were issued.
  - Traffic intended for Google subdomains was intercepted: MITM attack.

- How CAs Protect Their Private Key
  - Hardware Security Model (HSM)

How do you implement cryptography?



Slides credit

- Security in computing 5th edition, Textbook Slides
- Computer security, a hands-on approach, Textbook Slides