PKI, Encryption, Certificates and You

MACTECH

Jim Flood

Jim has been programming computers for more than thirty years, Macintosh, Windows, and UNIX, including seven years at Apple.

He has lots of experience with networking protocols and distributed systems, and has spent the last six years at Townsend Security focusing on security and encryption.

When not programming, he is having fun raising his two middle-school kids.

1983

Security Examples

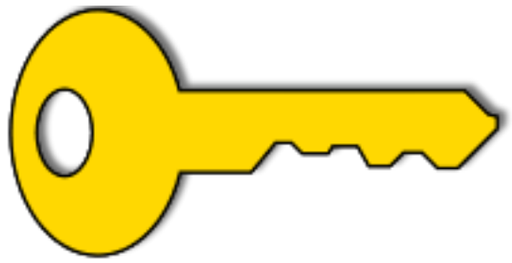
Secure Email	Code Signing
Web Server HTTPS	Apple ID

Security Tools

- Symmetric-Key Encryption
- Public-Key Cryptography
- Digital Signatures
- Certificates / PKI
- TLS (SSL)

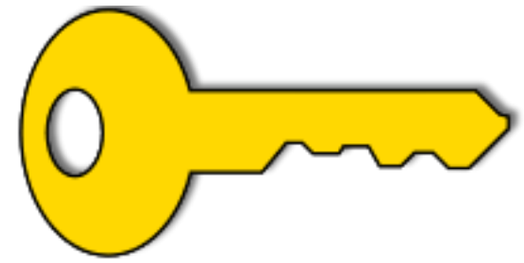
Alice and Bob

Symmetric-Key Encryption

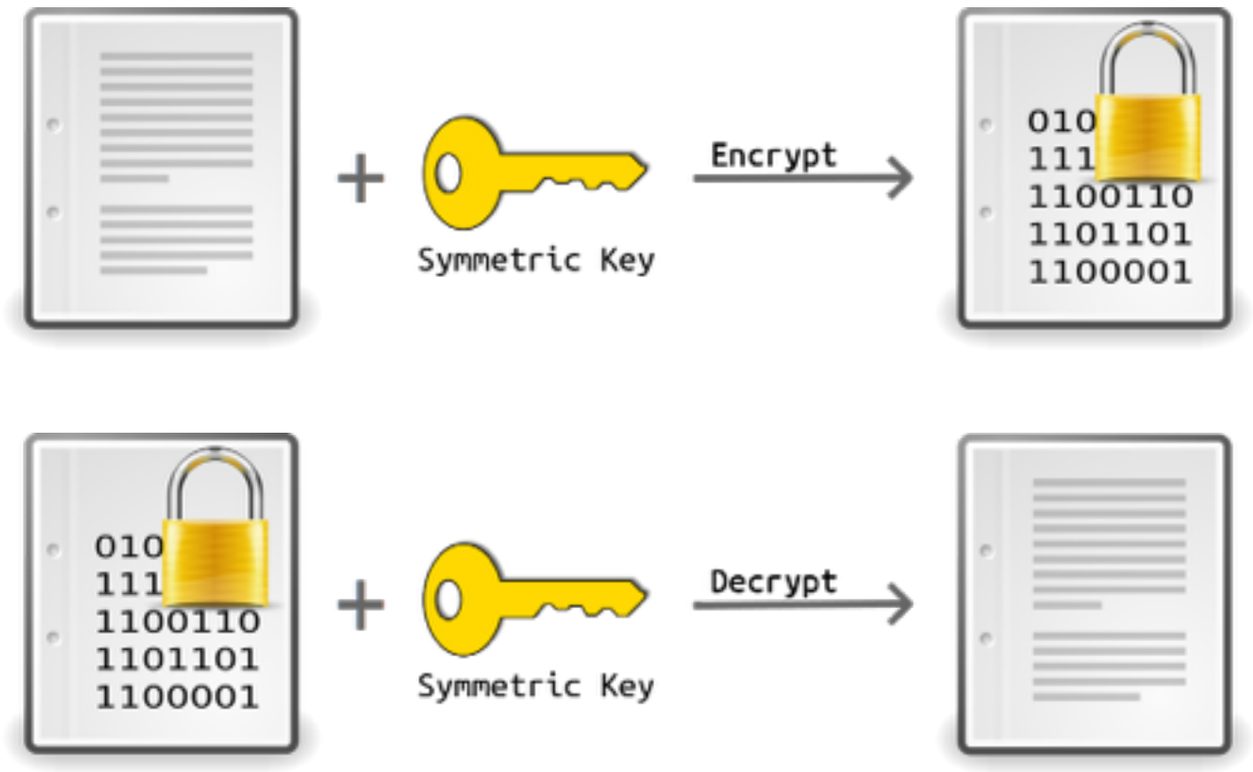


Symmetric Key

Symmetric-Key Encryption

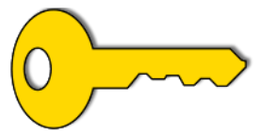


Symmetric Key



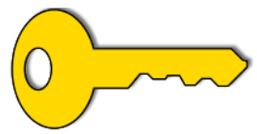
Symmetric-Key Encryption

Alice



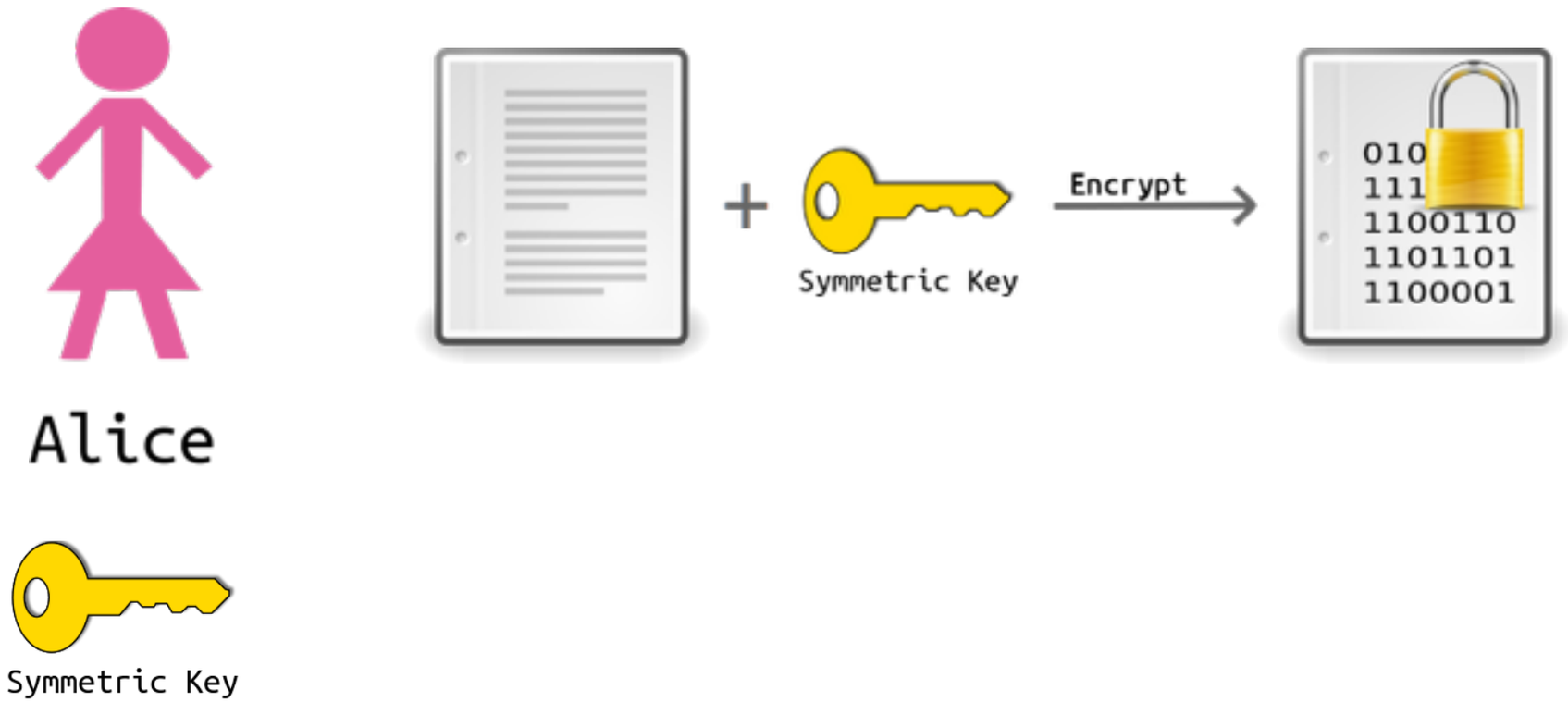
Symmetric Key

Bob



Symmetric Key

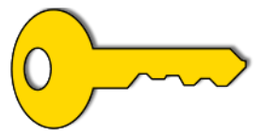
Symmetric-Key Encryption



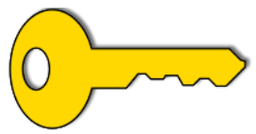
Symmetric-Key Encryption

Alice

Bob

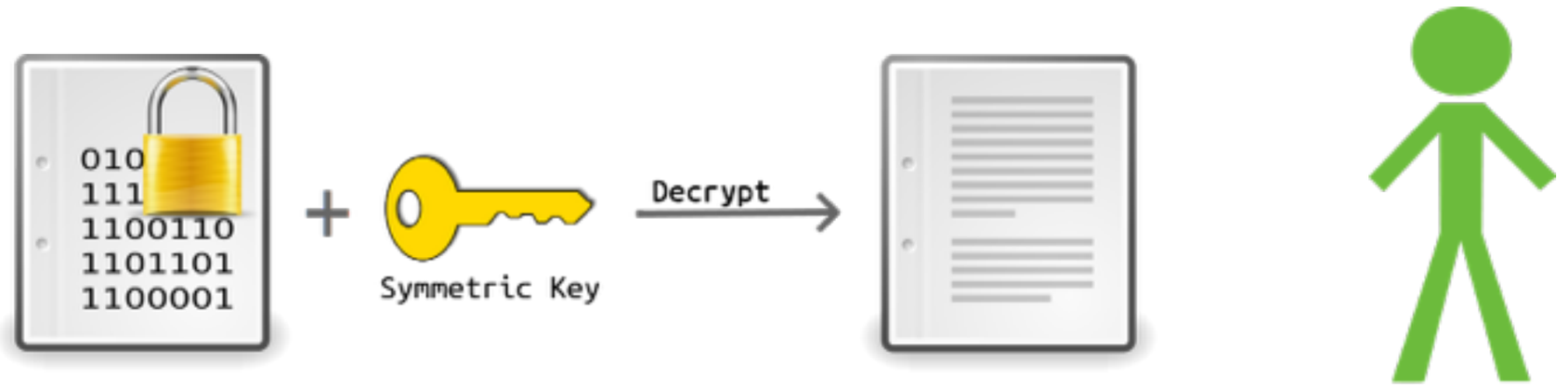


Symmetric Key

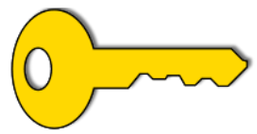


Symmetric Key

Symmetric-Key Encryption



Bob



Symmetric Key

Symmetric-Key Encryption

- Example: AES (Advanced Encryption Standard)
- Cipher Mode: CBC, GCM, …

- Initialization Vector (IV)
- Message Authentication Code (MAC)

Symmetric-Key Encryption

Example 128-bit AES key:
00100000011111101001110101111000010000000000000000011100101
01101100010110000111010111100101001110010011111011101010010
01110111010

Hexadecimal: 20 7E 9D 78 40 00 1C AD 8B 0E BC A7 27 DD 49 DA

Decimal: 43,192,719,134,787,147,294,228,222,557,443,410,394

Hash Function

Hash Function

```
1-800-977-6368
1 + 8 + 0 + 0 + 9 + 7 + 7 + 6 + 3 + 6 + 8
55
5
```

```
503-588-2941
5 + 0 + 3 + 5 + 8 + 8 + 2 + 9 + 4 + 1
45
5
```

Hash Function

Example: SHA-1 hash (20 bytes)

1-800-977-6368

859d925489b53f266a5103f168a7bdbccdf99ca2

503-588-2941

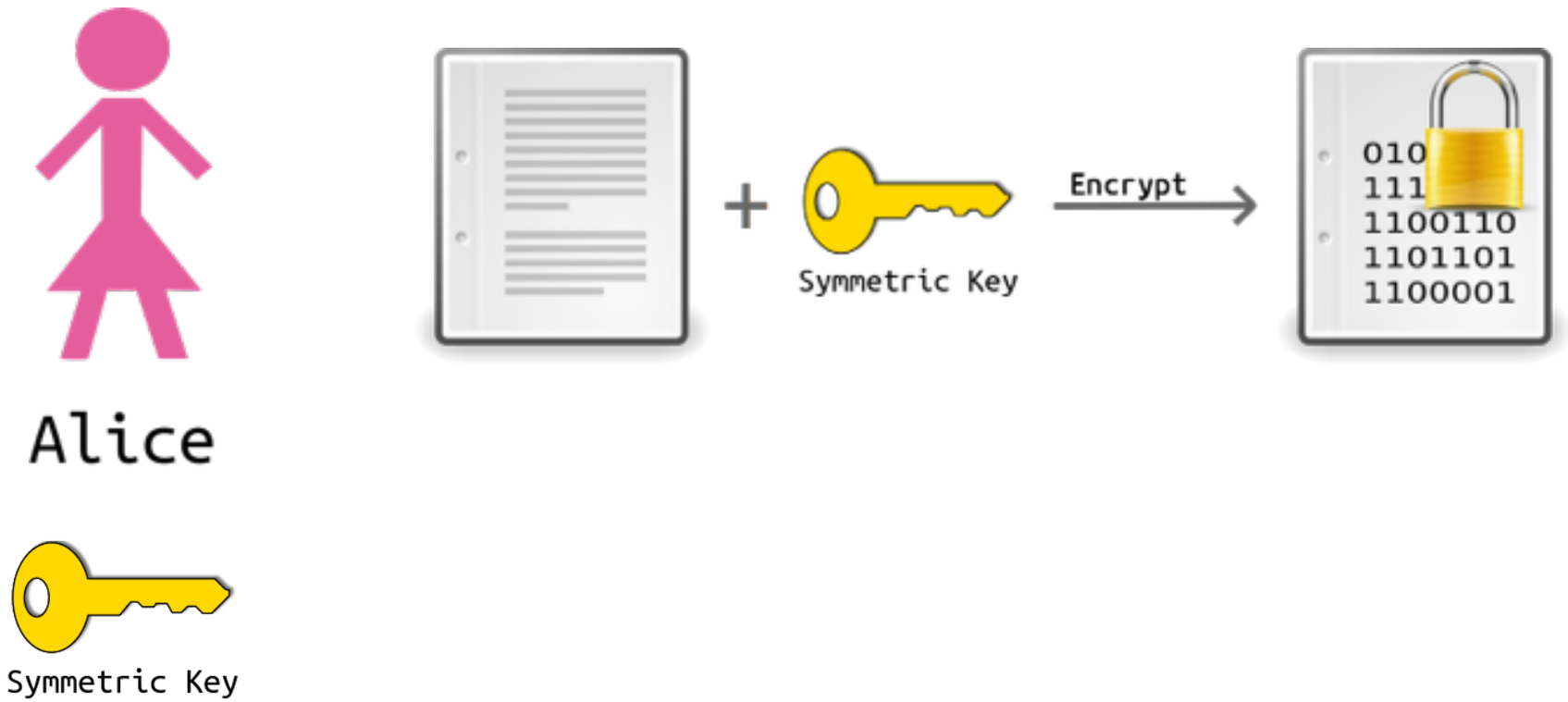
c66f132b9aaf20232afecc317f9e25c2ac692634

Hash Function

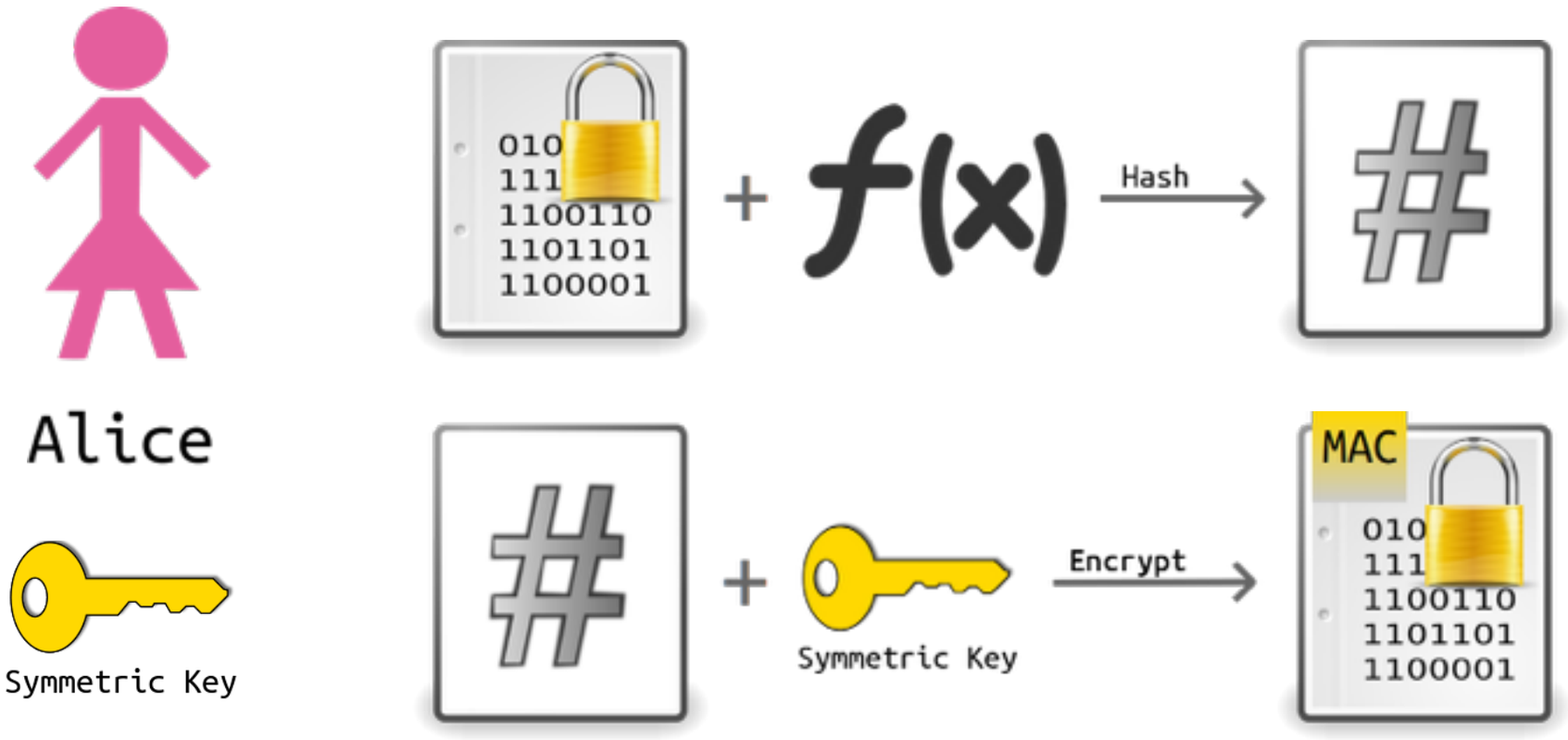
- With a cryptographic hash, it should be:

1. “Impossible” to change the input data without changing the output hash
2. “Impossible” to create two inputs that result in the same output hash

Message Authentication Code (MAC)



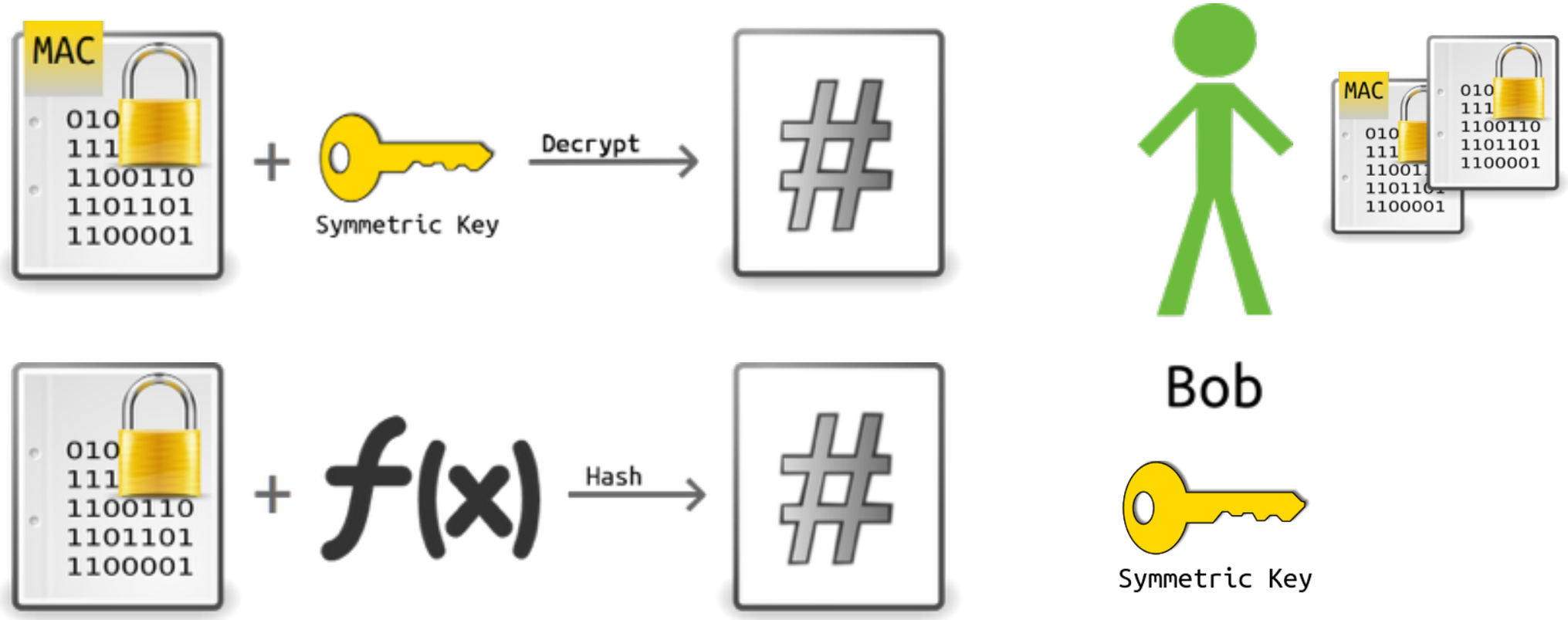
MAC


Alice
Symmetric Key
010
111
1100110
1101101
1100001
+
f(x)
Hash
#
#
+
Symmetric Key
Encrypt
MAC
010
111
1100110
1101101
1100001

MAC

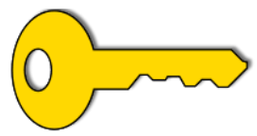

MAC
010
111
1100110
1101101
1100001
010
111
110011
110110
1100001
Alice
Bob
Symmetric Key
Symmetric Key

MAC


MAC
010
111
1100110
1101101
1100001
Symmetric Key
Decrypt
#
MAC
010
111
1100110
1101101
1100001
Bob
Symmetric Key
010
111
1100110
1101101
1100001
f(x)
Hash
#

MAC

Bob



Symmetric Key

MAC

- Encrypt, then MAC
- AES GCM mode has MAC built-in :-)

Password-Derived Key

Alice

Bob

Password-Derived Key

correct horse battery staple

$f(x)$ many iterations

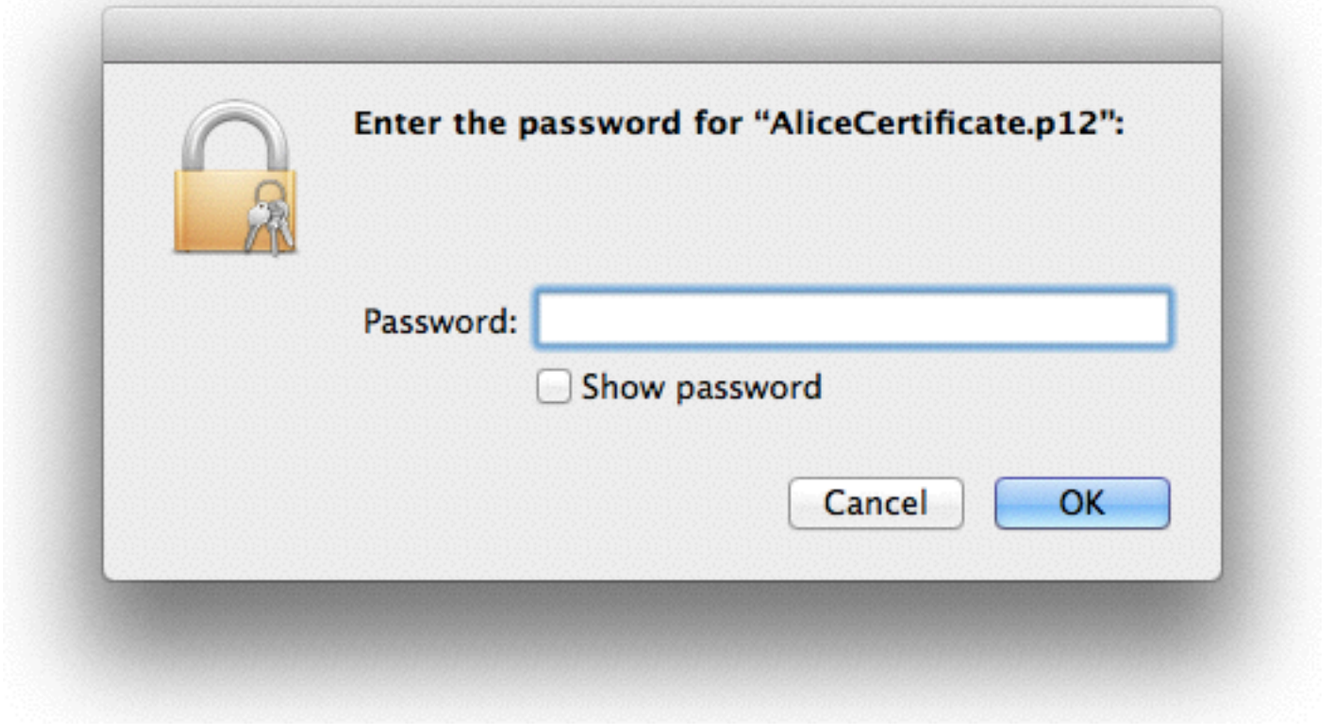
Password-Derived Key

correct horse battery staple

correct horse battery staple

Password-Derived Key

If you have seen a dialog like this on your Mac, then you have used a password-derived key.



Password-Derived Key

My example 128-bit AES key came from “correct horse battery staple”:
00100000011111101001110101111000010000000000000000011100101
01101100010110000111010111100101001110010011111011101010010
01110110 10

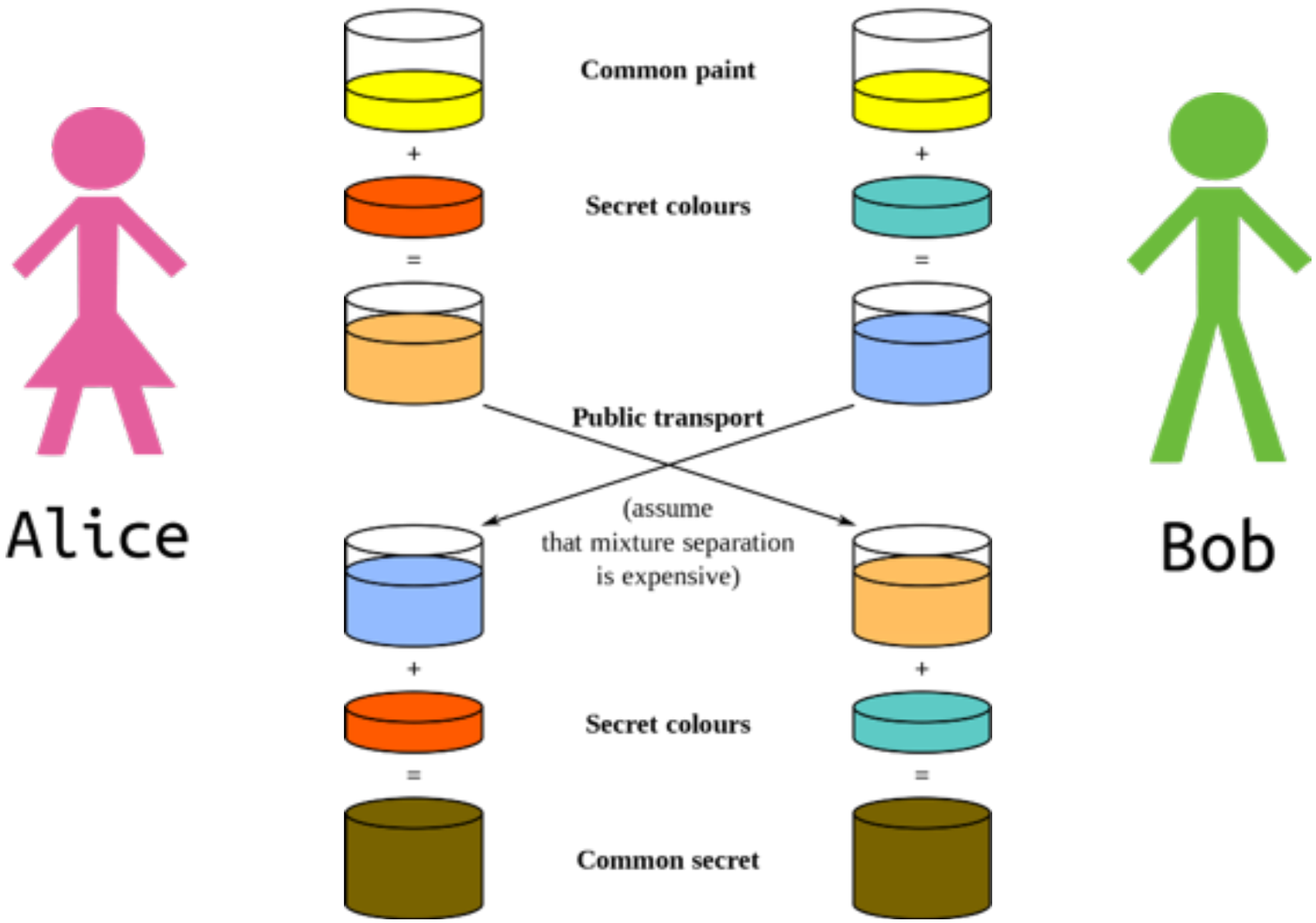
(With salt EE 27 D6 46 55 55 23 E6 and one SHA-1 iteration.)

Key Exchange

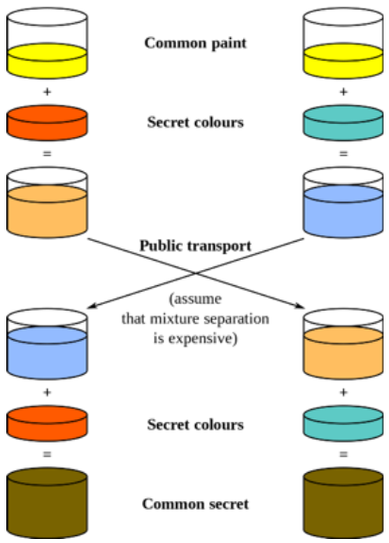
Alice

Bob

Key Exchange

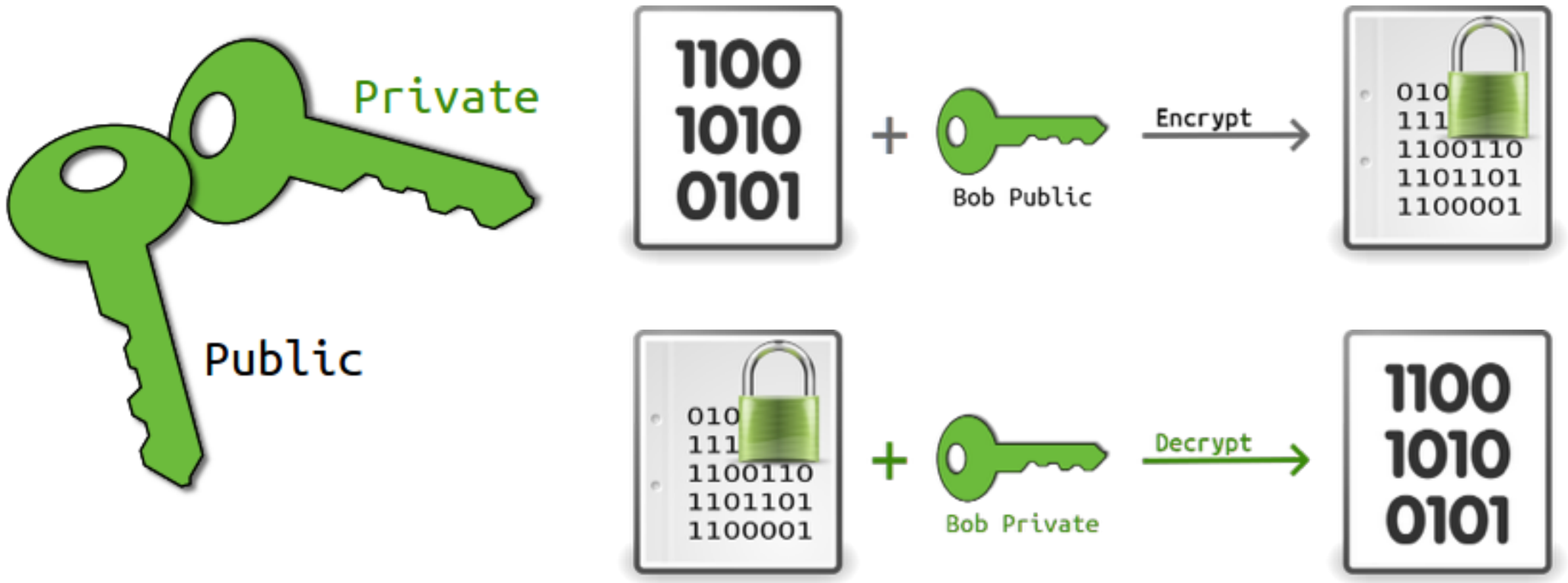


Key Exchange



- Example: Diffie-Hellman
- Other key exchanges possible with Public-Key Cryptography.

Public-Key Cryptography



Public-Key Cryptography

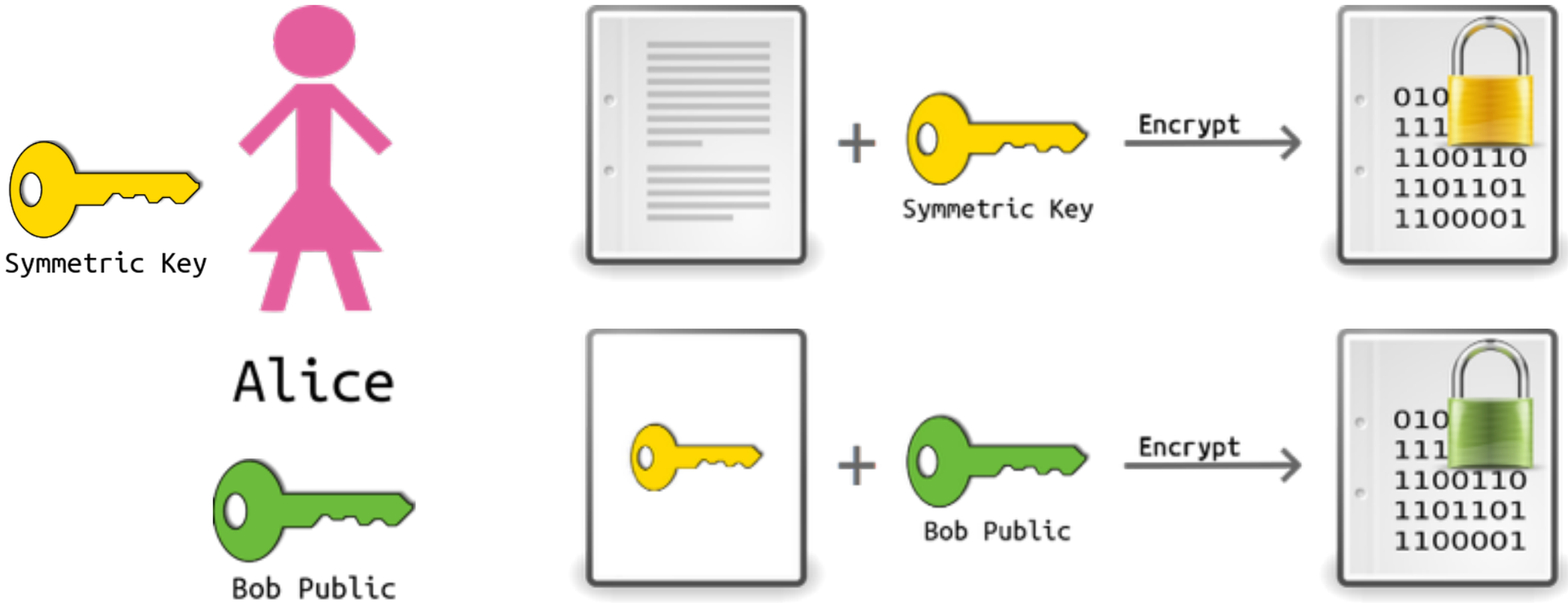
Alice

Bob Public

Bob

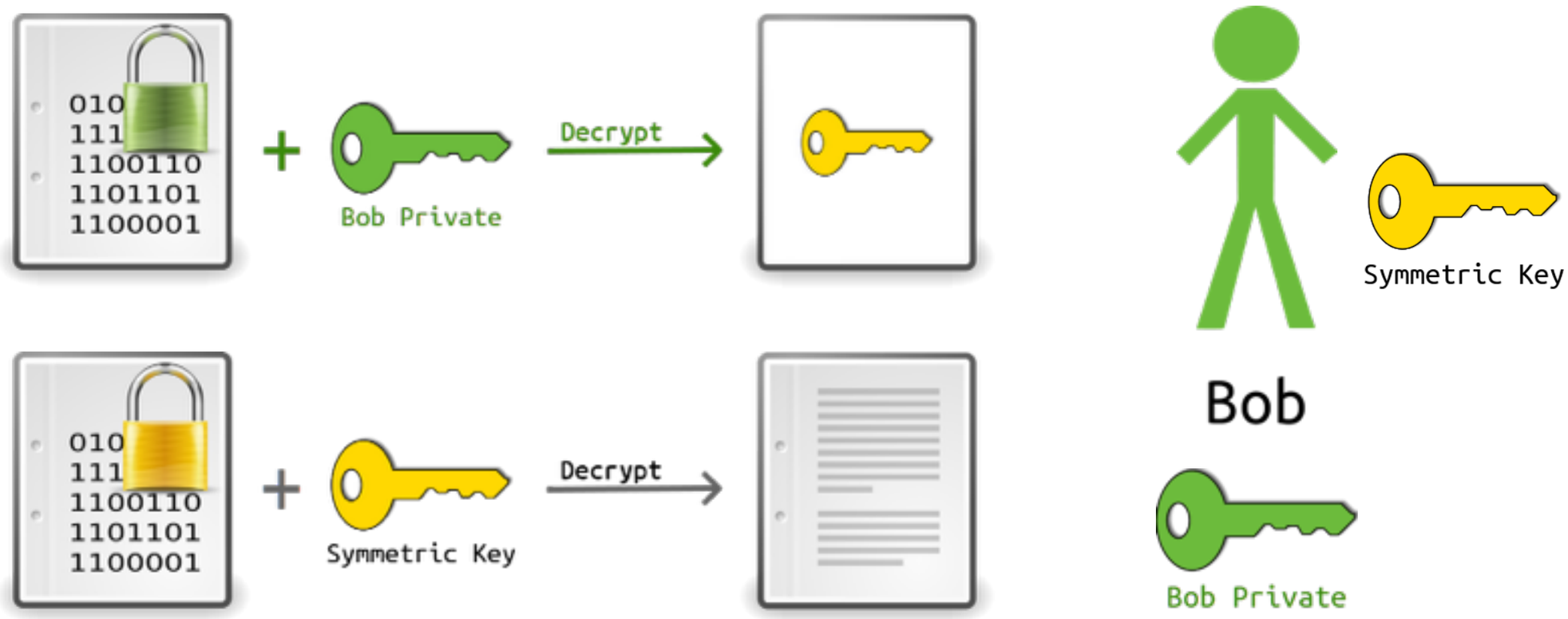
Bob Private

Public-Key Cryptography



Public-Key Cryptography

Public-Key Cryptography



Public-Key Cryptography

- Example: RSA
  - 1024-bit key common
  - 2048-bit recommended
- OAEP padding randomizes each encryption output

Public-Key Cryptography

Example RSA 1024-bit key:

n=132211356187213088007302242725218857315475016313665436448721544972388911024048419992675564946046445981882176279474768008224925305773923497714649499086540144796973661813101274747966178081439441547425138569829048878336025301769610663751411768474033240865973724419022280230311667539969778465368981923990056228559

e=65537

d=93157392099319864778692997307859016938736977636029151842914132531775563929818391724091310833247368395736074831523961076701872236599603876855319660173920834711579842450160476774312594490658875869886868318882863833074691935282886255826424588856473215047314634042206575979439773188716381551759438573163809101249

Given public, finding private is hard. Given private, finding public is not so hard. Keep private, private!

Public-Key Cryptography

FYI: n equals p times q, which are both prime numbers:

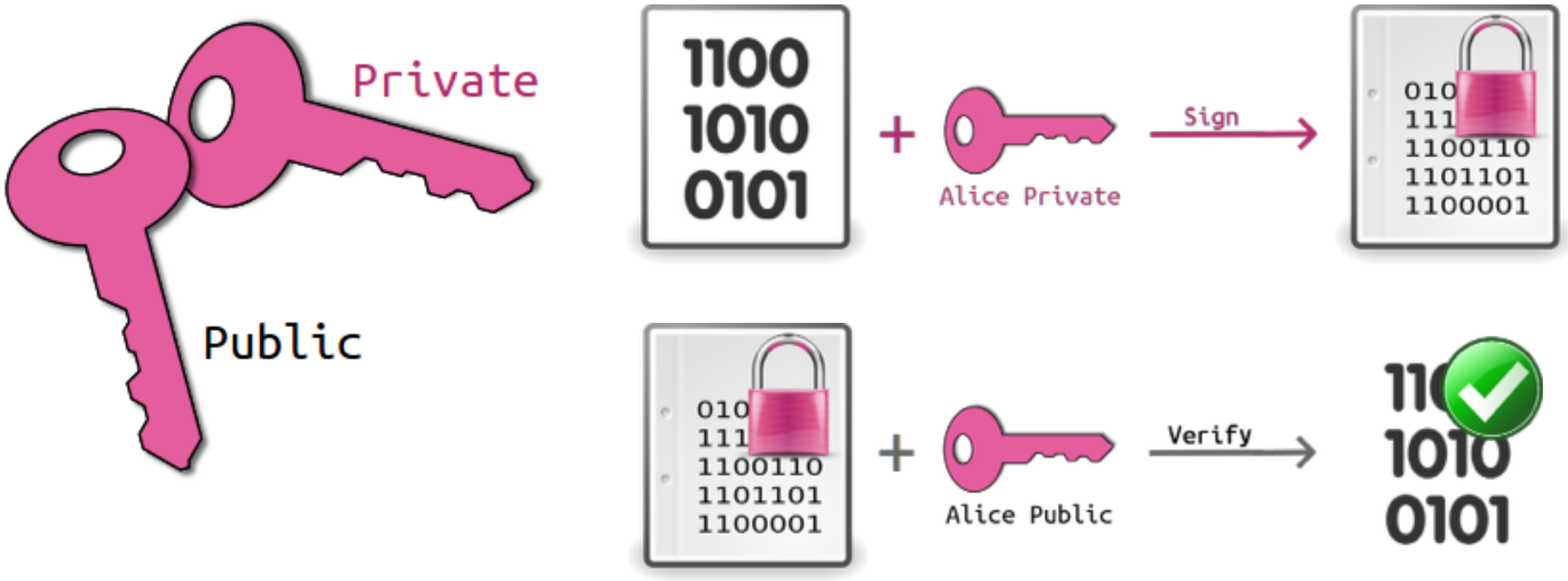
p=1152580355530515884836750875523346196149422378807954838030319603828981351313670378283852465196061314723222746475397221223677428878493708513066072297001046370378283852465196061314723222746475397221223677428878493708513066072297001046

q=1147090140421126073452810079187361066581514912455986734952649742810159543841344166097378223057873790901696040649394910933991140178053166948239130562064859344166097378223057873790901696040649394910933991140178053166948239130562064859

Public-Key Cryptography

- Elliptic Curve Cryptography (ECC)
- Not as common as RSA
- You may find ECDSA keys in your keychain (Elliptic Curve Digital Signature Algorithm)
- cf. http://www.nsa.gov/business/programs/elliptic_curve.shtml

Digital Signature



Digital Signature

Alice

Alice Private

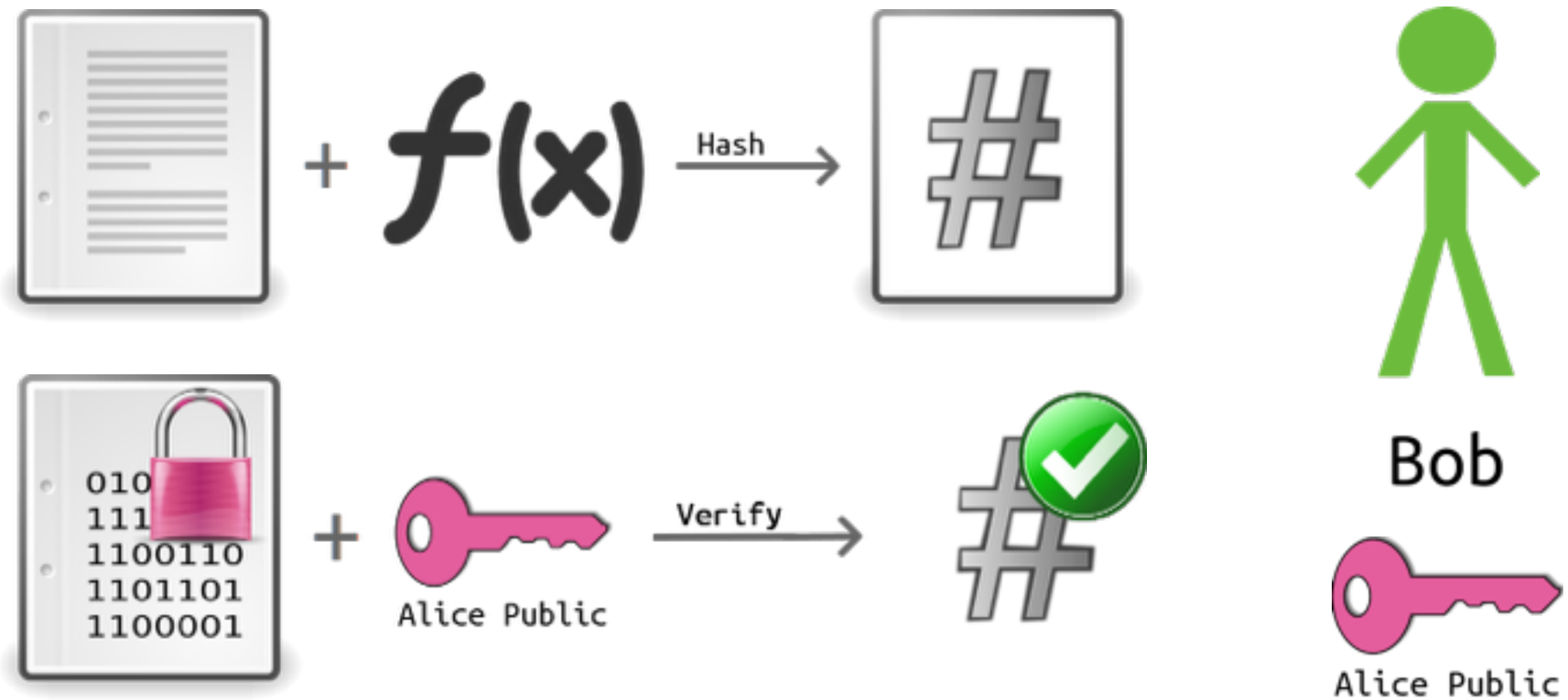
Bob

Alice Public

Digital Signature

Digital Signature

Digital Signature



Digital Signature

Don't use the same key pair for both encryption and signatures.

Sign then Encrypt

Alice

Alice Private

Bob Public

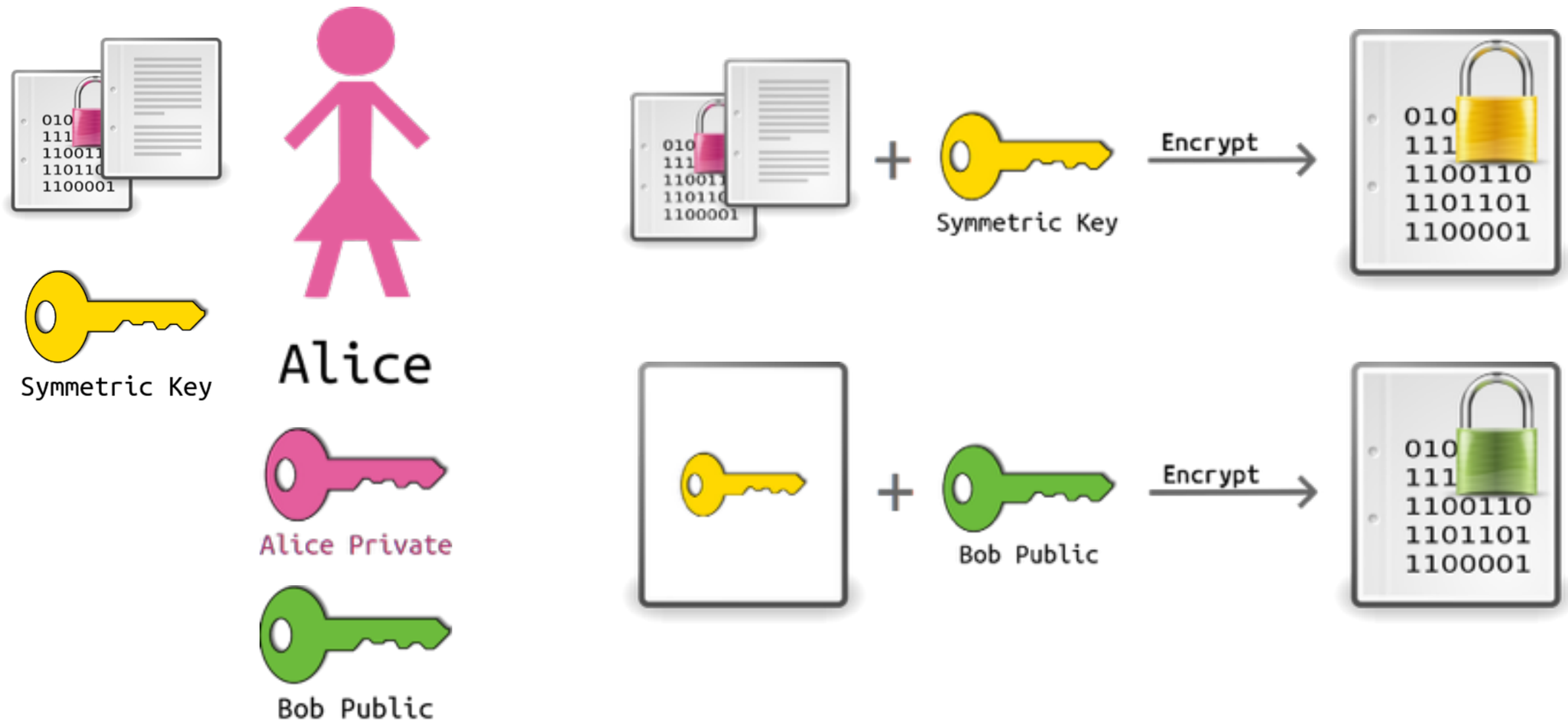
Bob

Bob Private

Alice Public

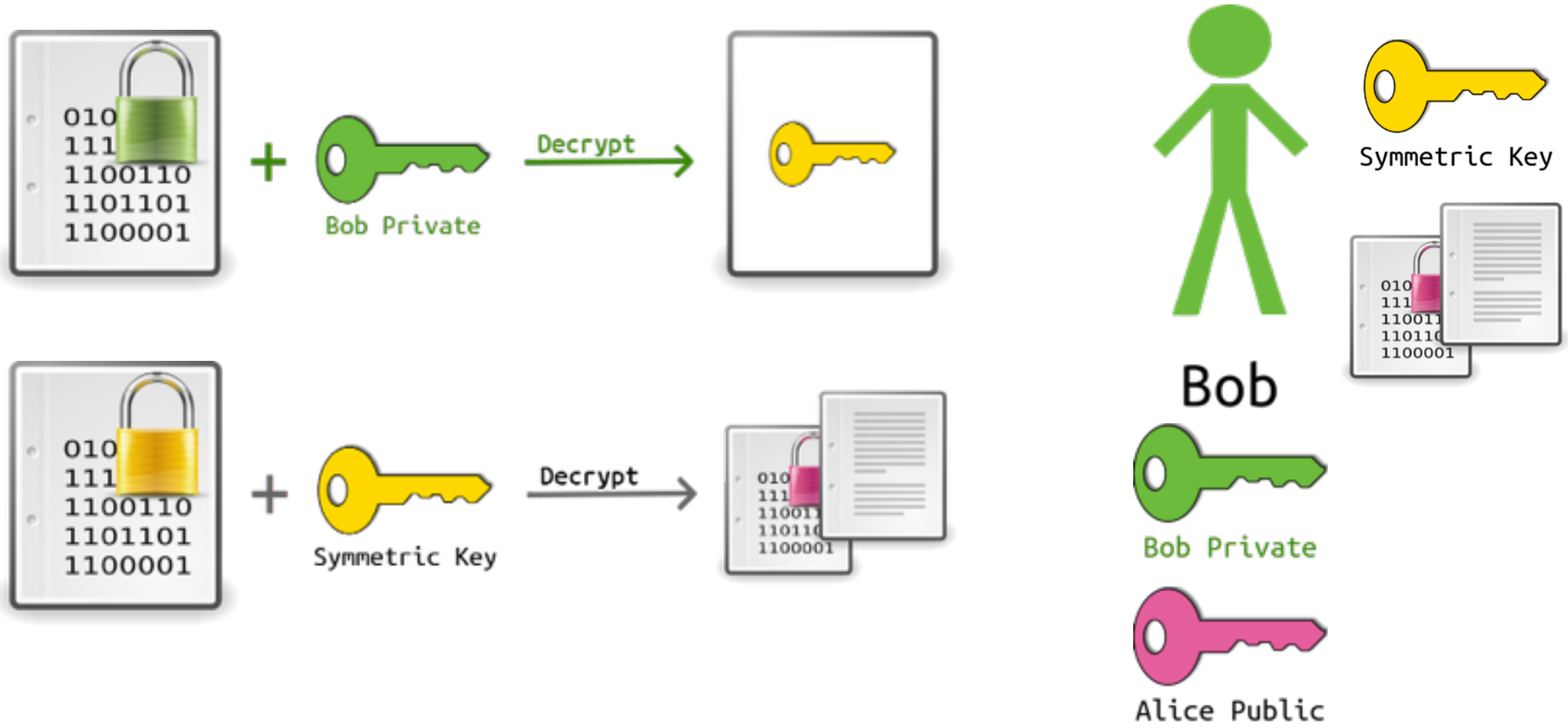
Sign then Encrypt

Sign then Encrypt

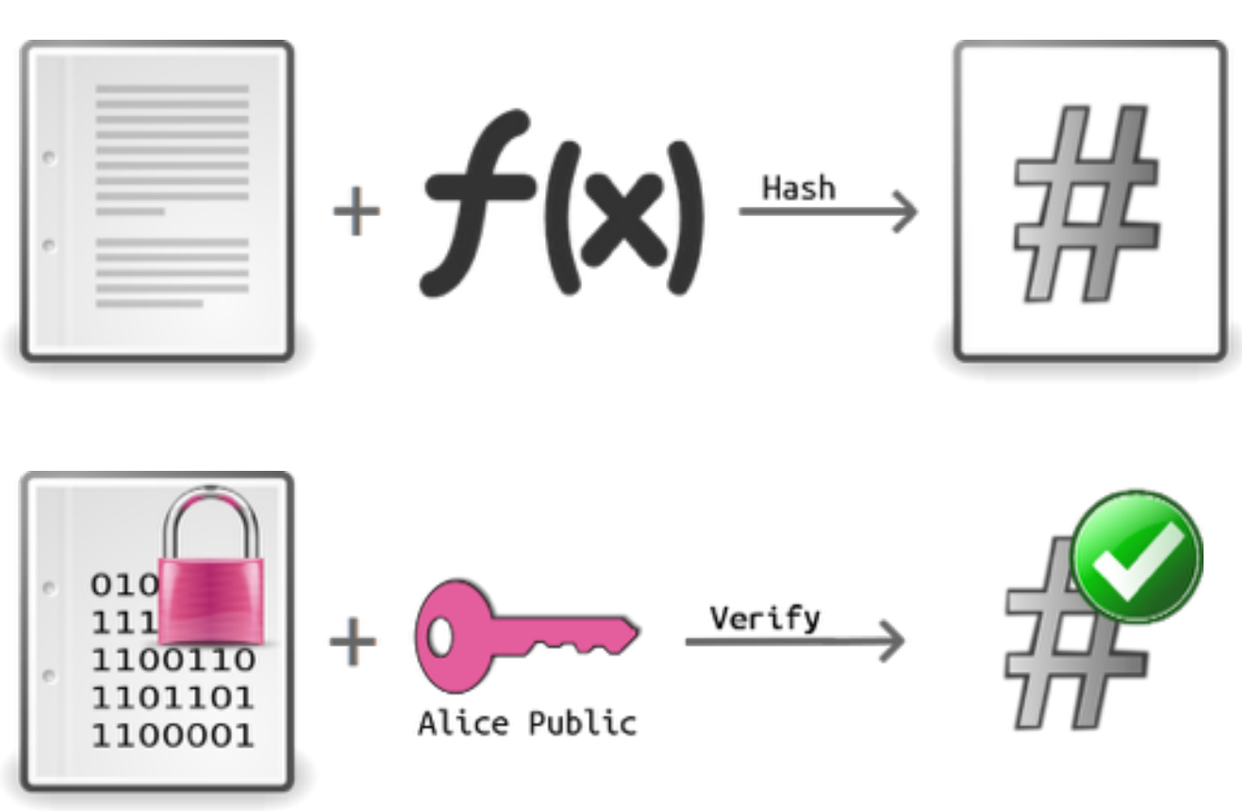


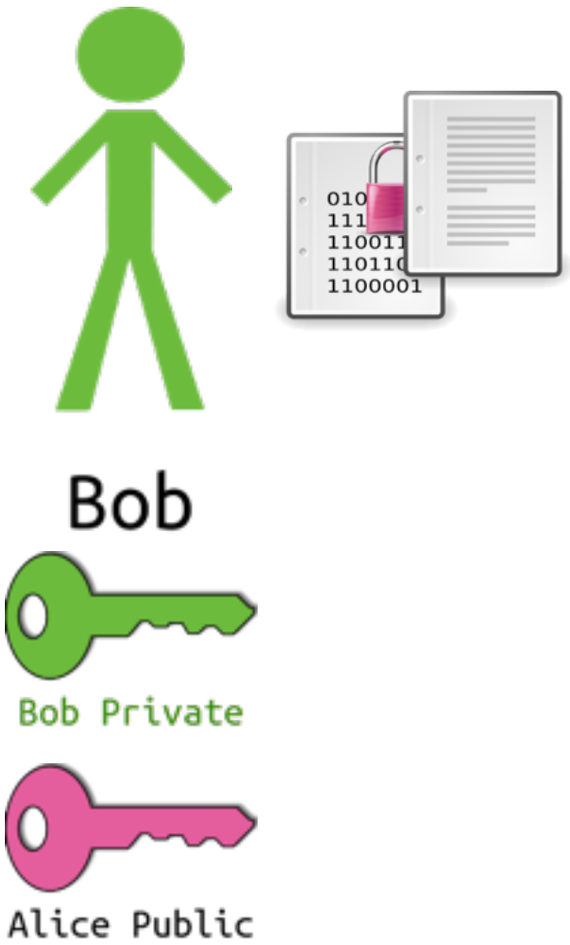
Sign then Encrypt

Sign then Encrypt



Sign then Encrypt

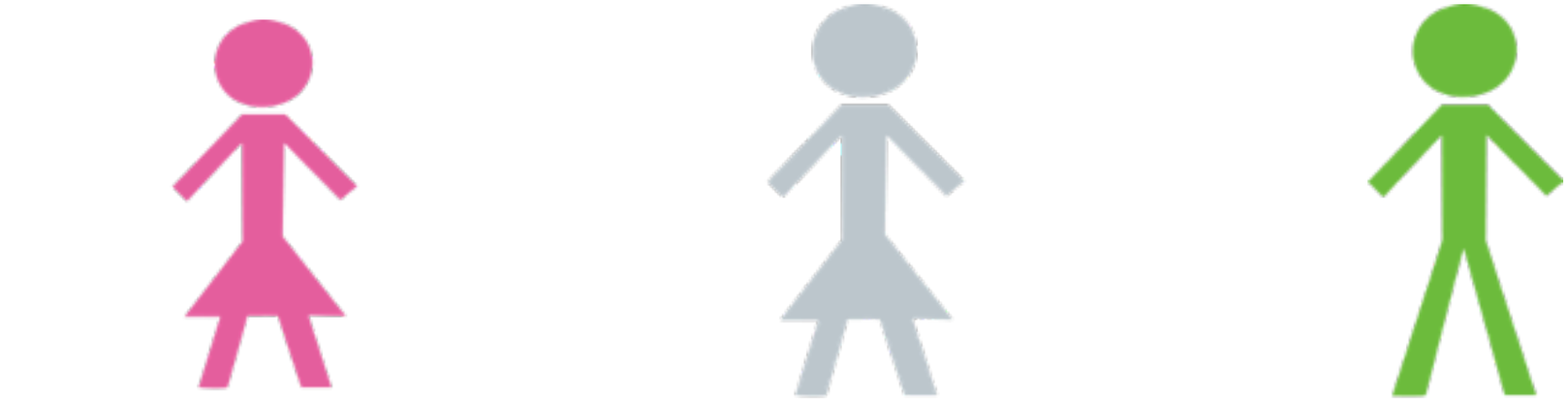


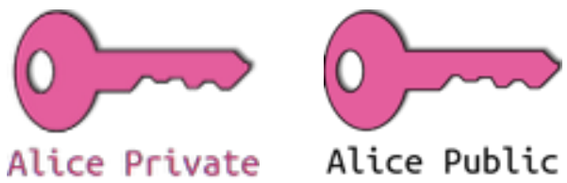


Sign then Encrypt

- Did somebody re-encrypt it?
- You can sign, encrypt, then sign again
- Don't encrypt then sign:
  - What did you sign?
  - Was it yours to sign?

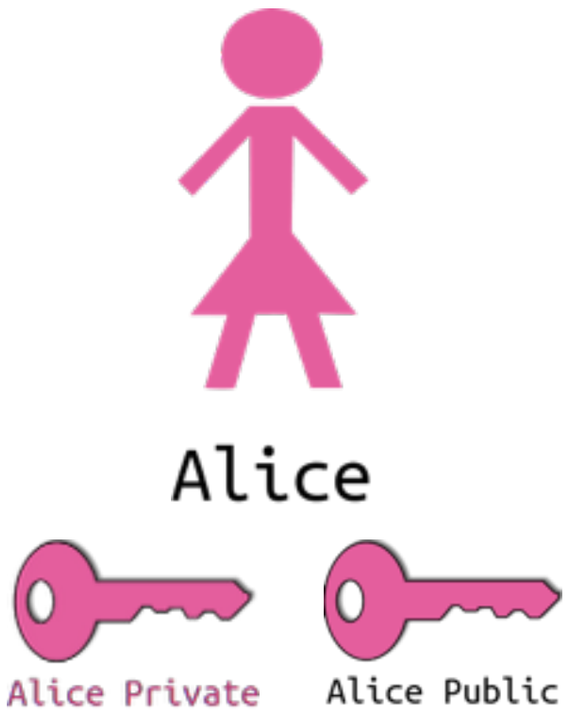
Certificates



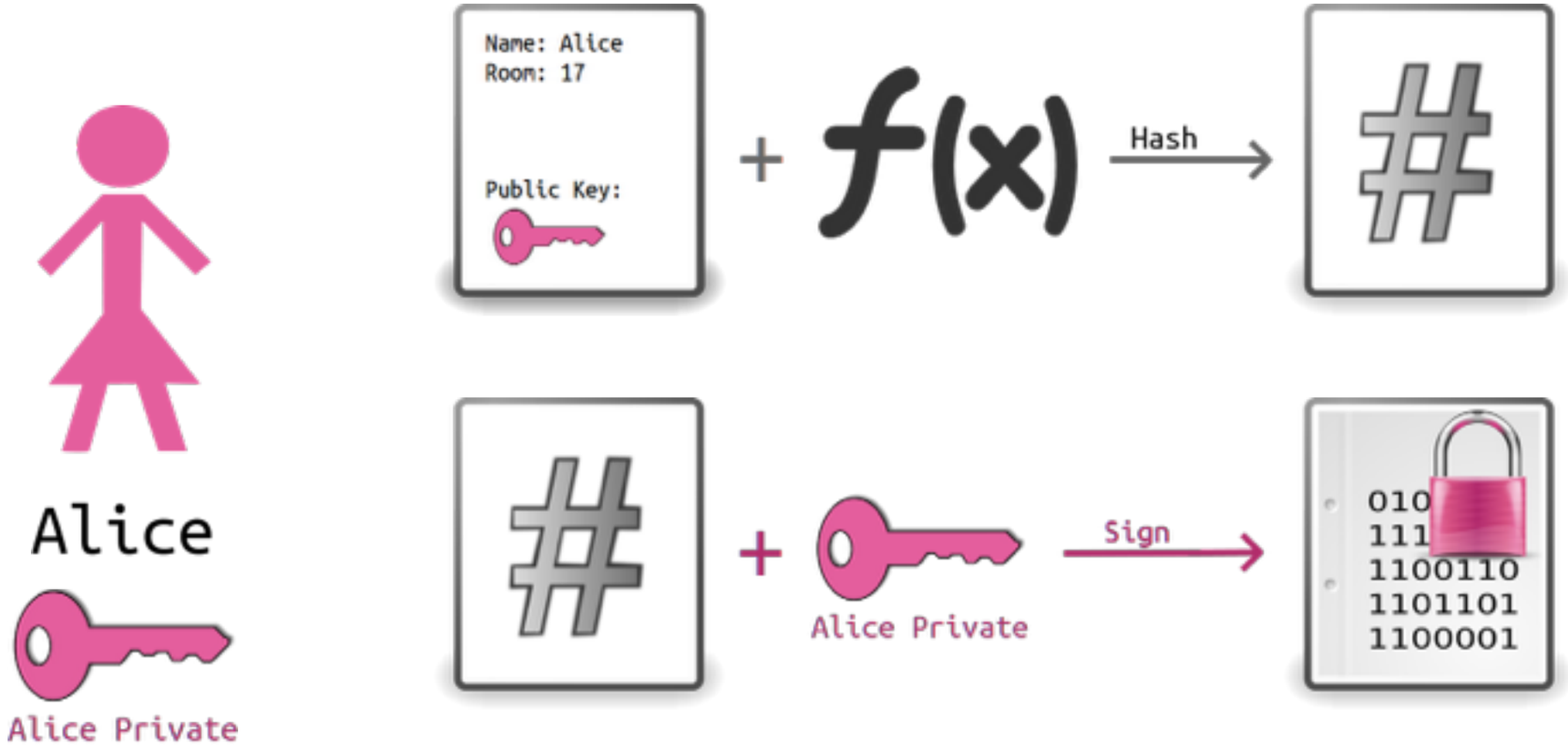


Certificate Authority

Certificates



Certificates



Certificates

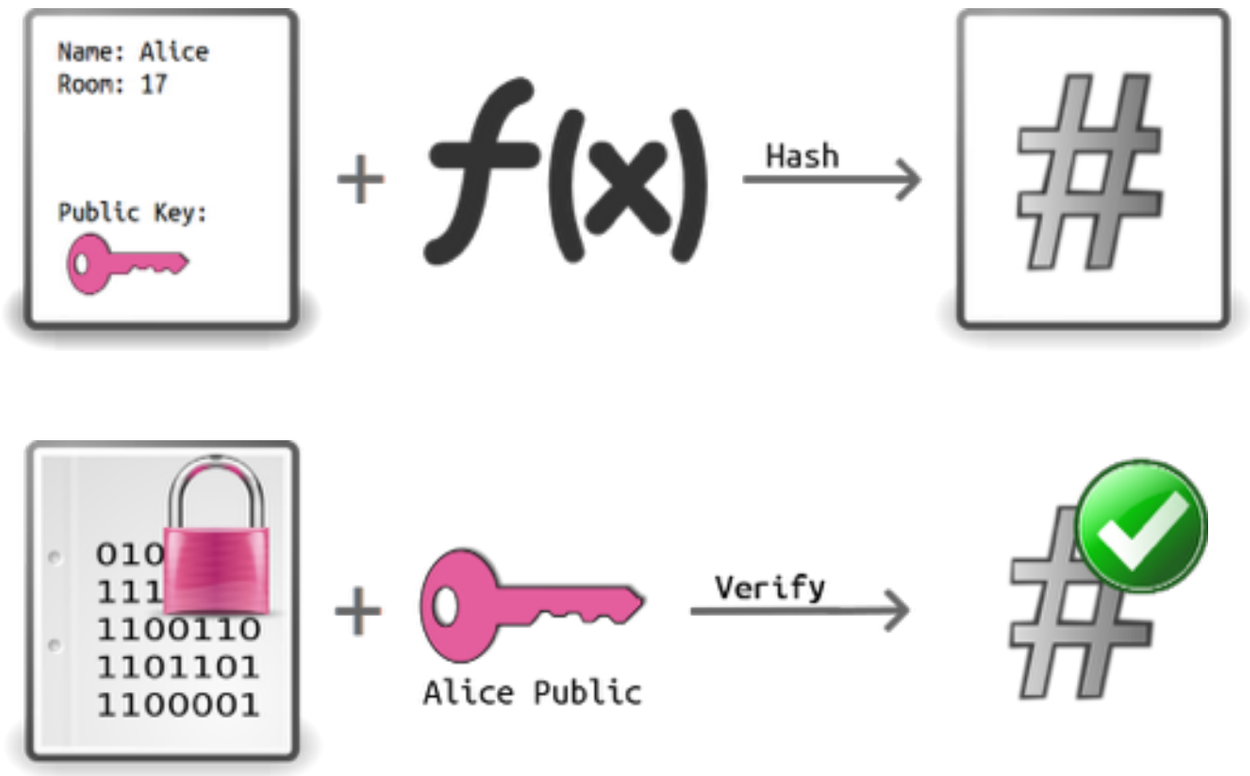
Alice

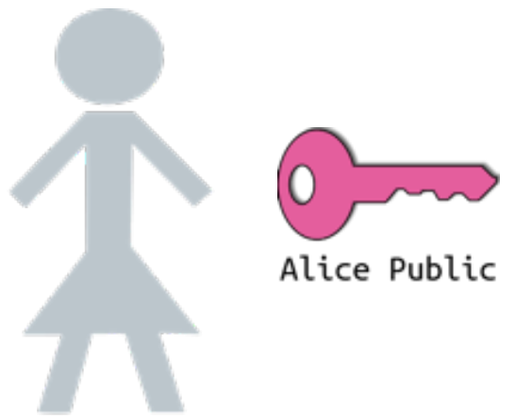
Alice Private

Miss Crabtree

Certificate Signing Request (CSR)

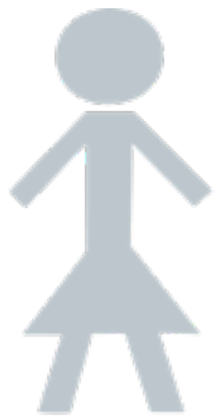
Certificates





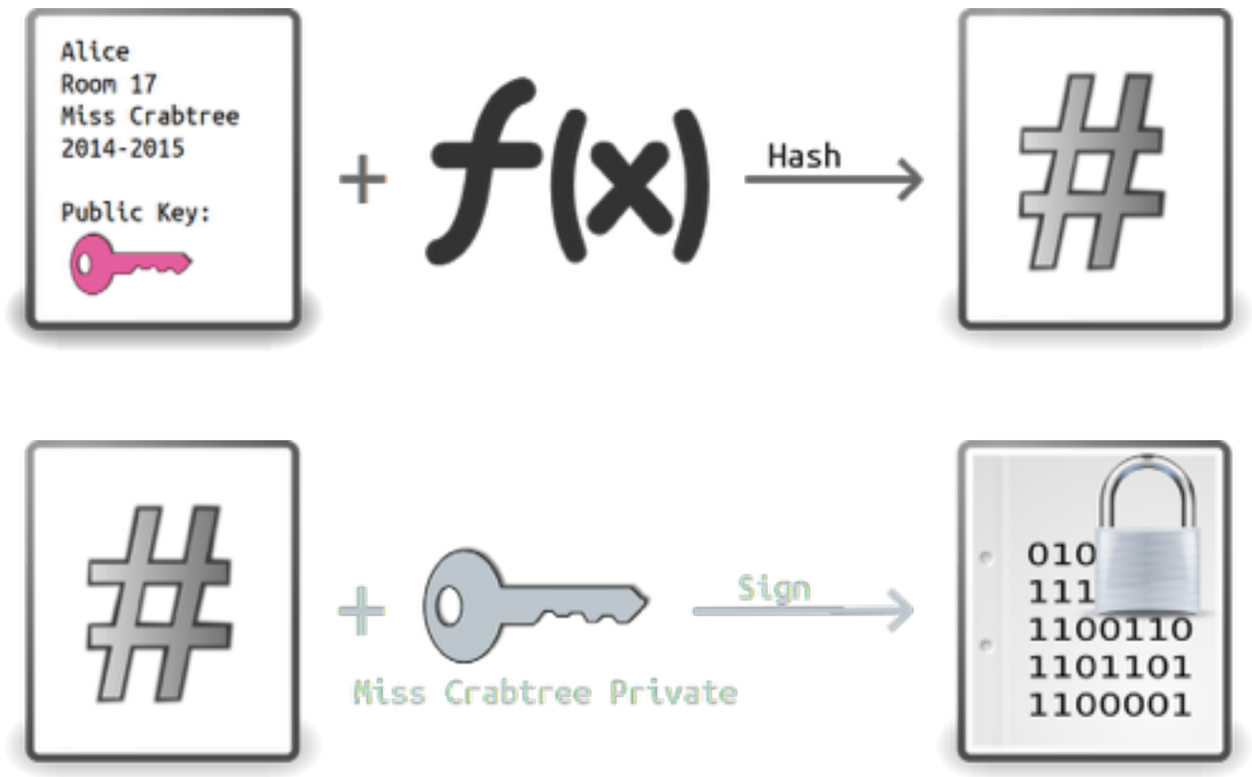
Miss Crabtree

Certificates

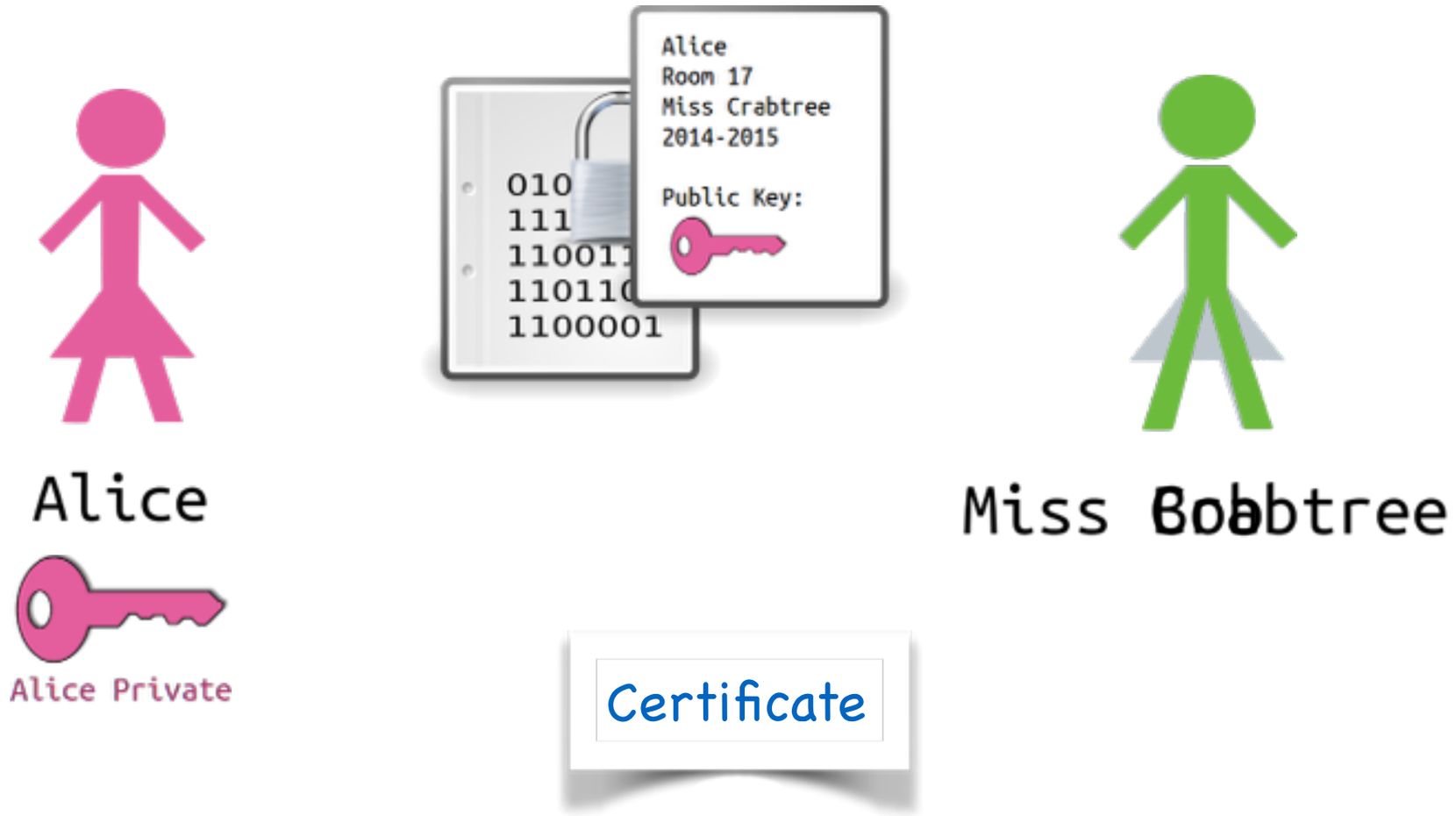


Miss Crabtree

Certificates



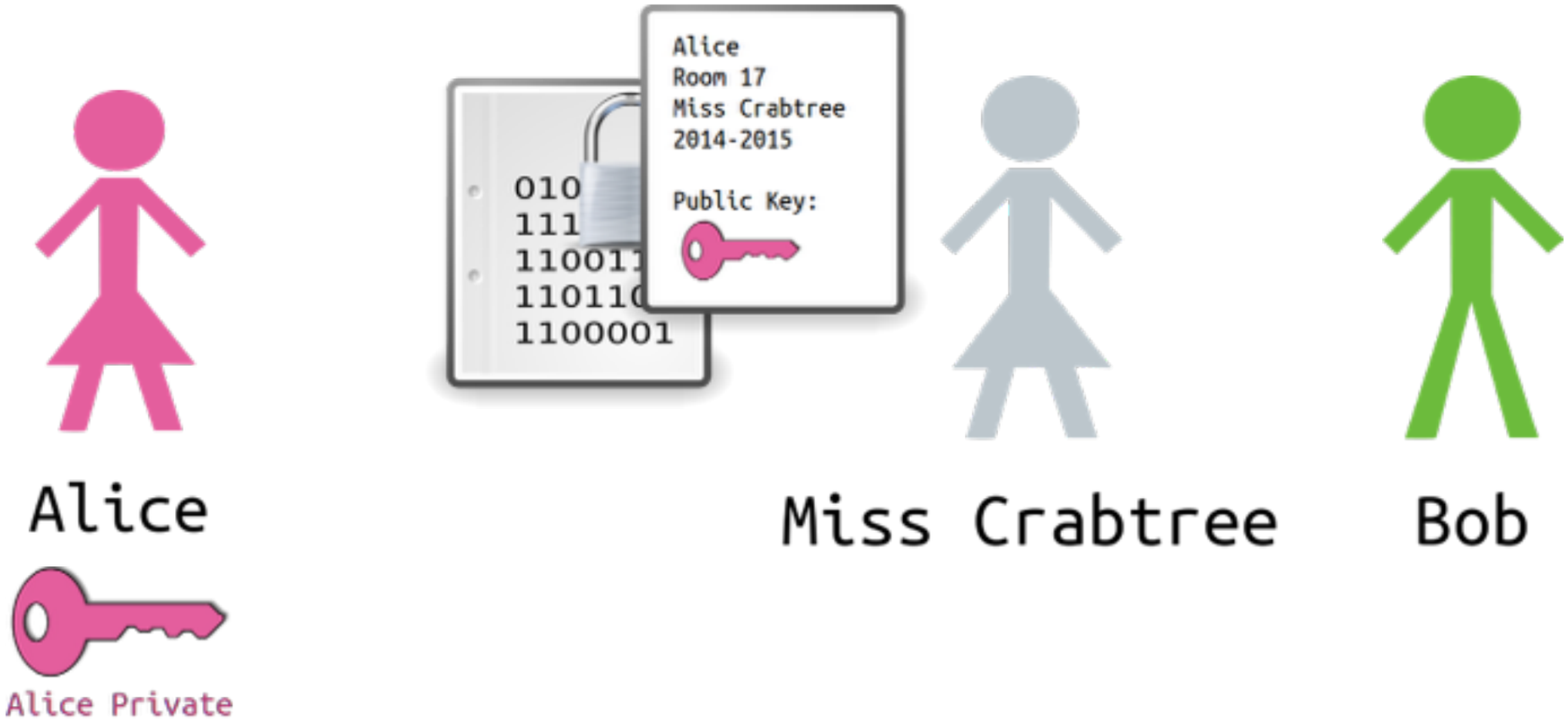
Certificates



Certificates

- Example: X.509, a.k.a. "SSL certificates"
- Subject (Alice) and issuer (Miss Crabtree)
- Validity period
- Can be revoked by issuer
- Alternate names, key usage flags, …
- Contains the public key
- Does not contain the private key!

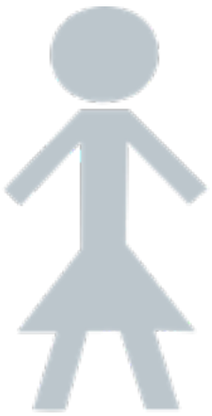
Certificates



Certificates

Miss Crabtree
Room 17
Miss Crabtree
2014-2024

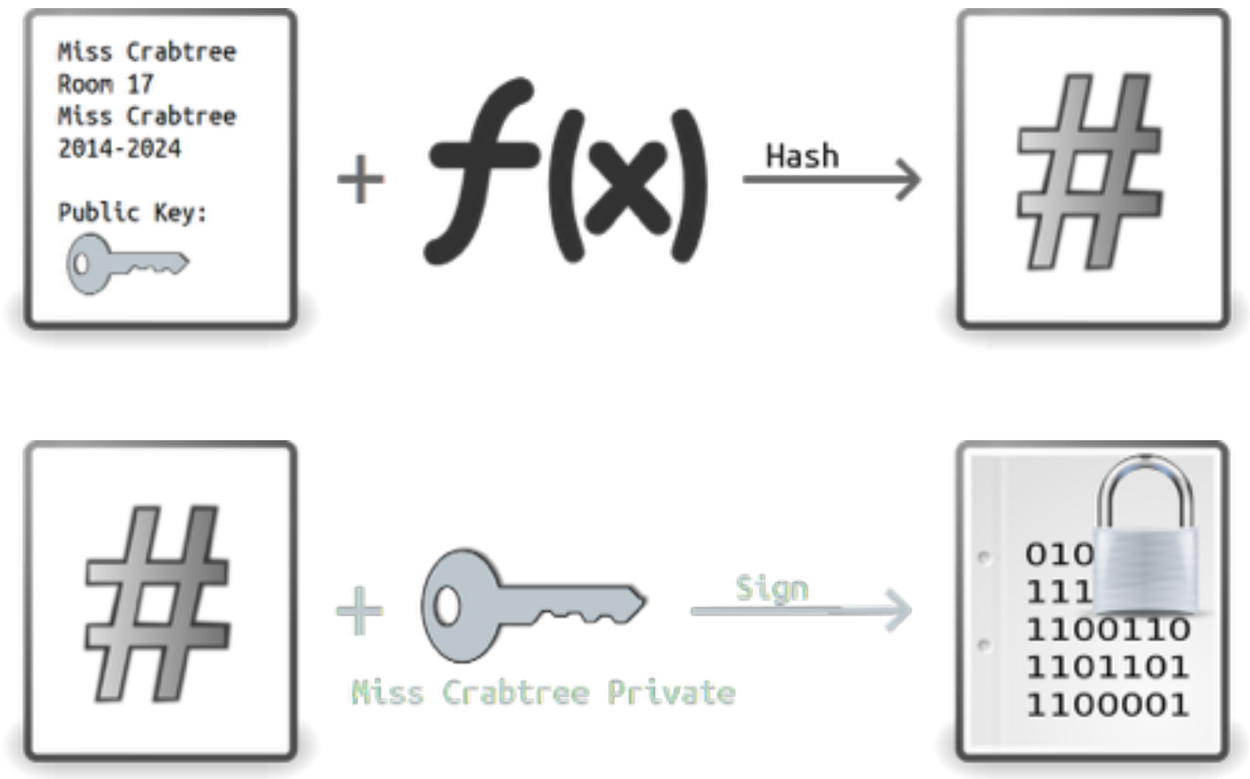
Public Key:



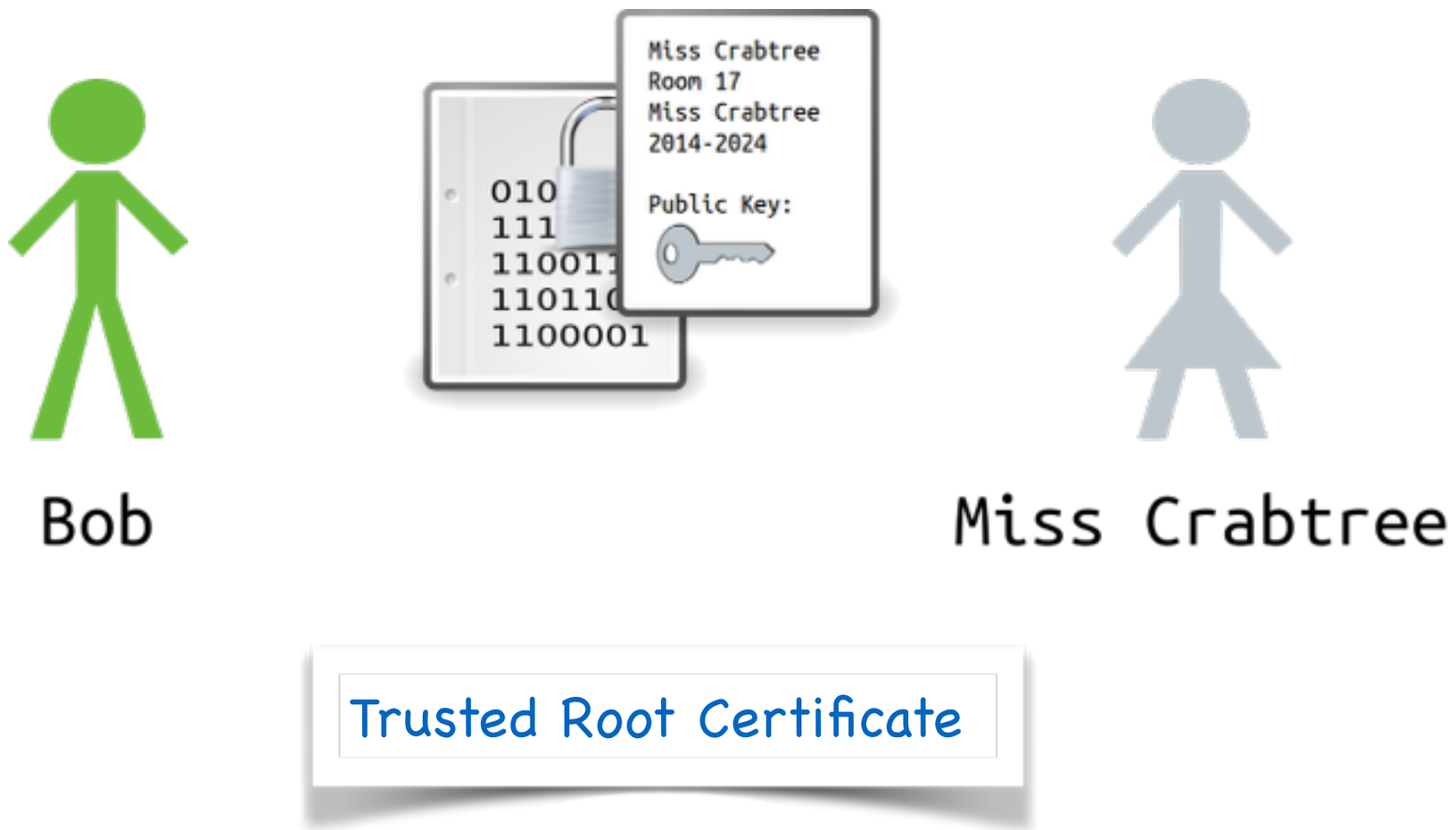
Miss Crabtree

Miss Crabtree Private

Certificates



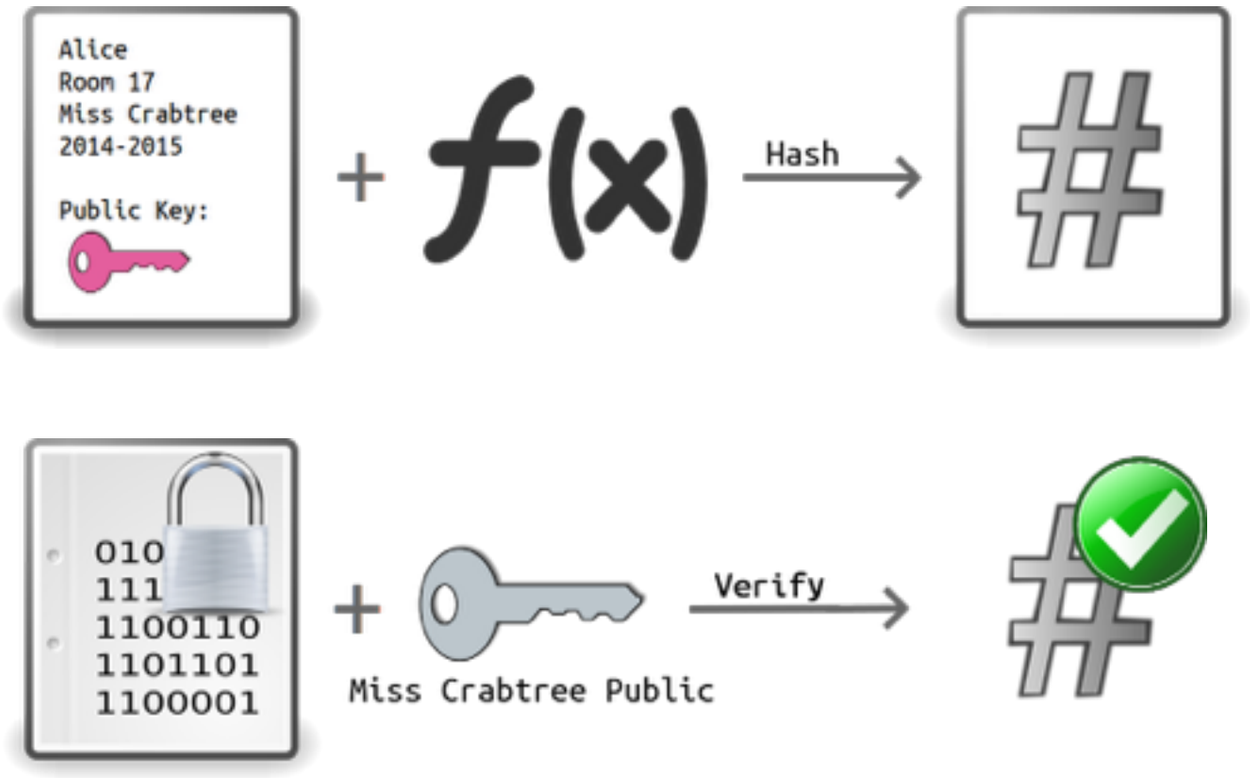
Certificates

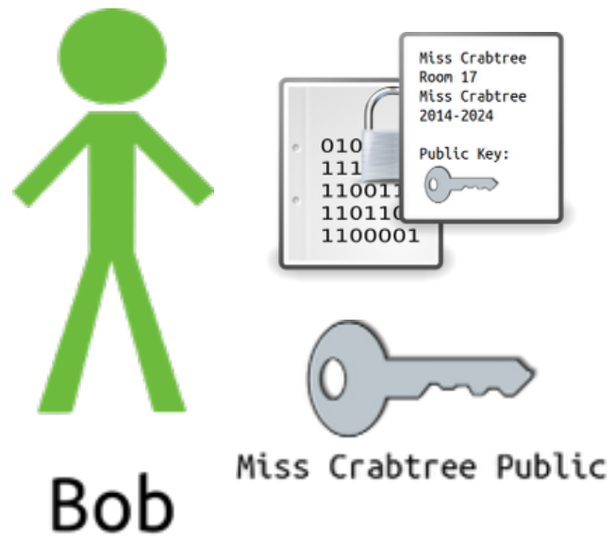


Trusted Root Certificate

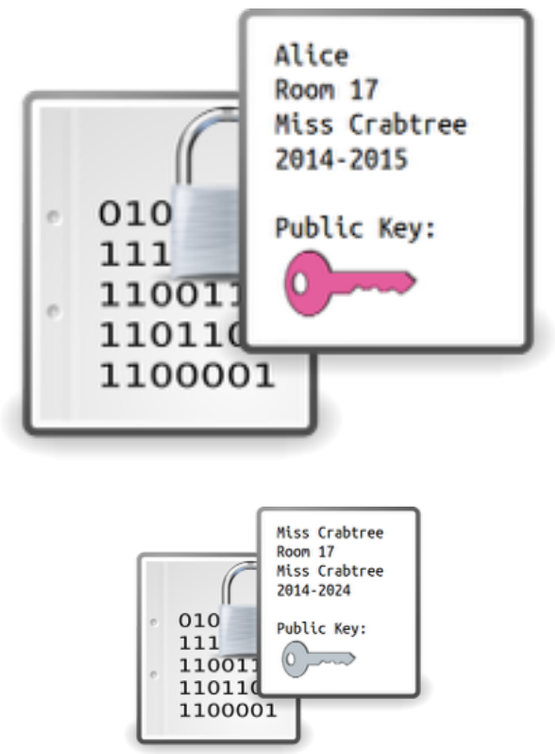
Certificates

Certificates





Certificates

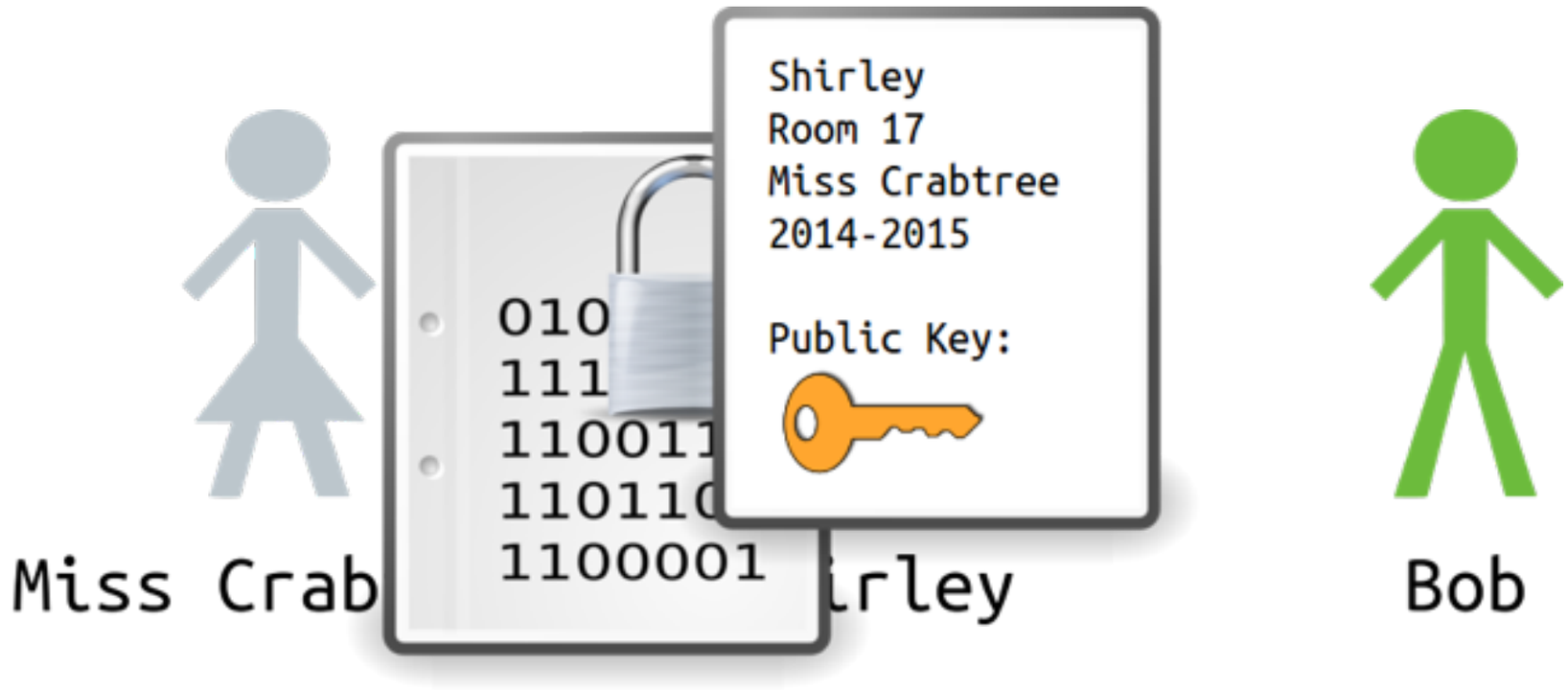


- Types of certificate:
  - "End-entity", e.g. Alice, Bob, …
  - Certificate Authority (CA)
- 211 trusted root certificates come with Apple Mavericks: http://support.apple.com/kb/ht6005
- 317 trusted root certificates come with Apple iOS 7: http://support.apple.com/kb/ht5012

Certificates

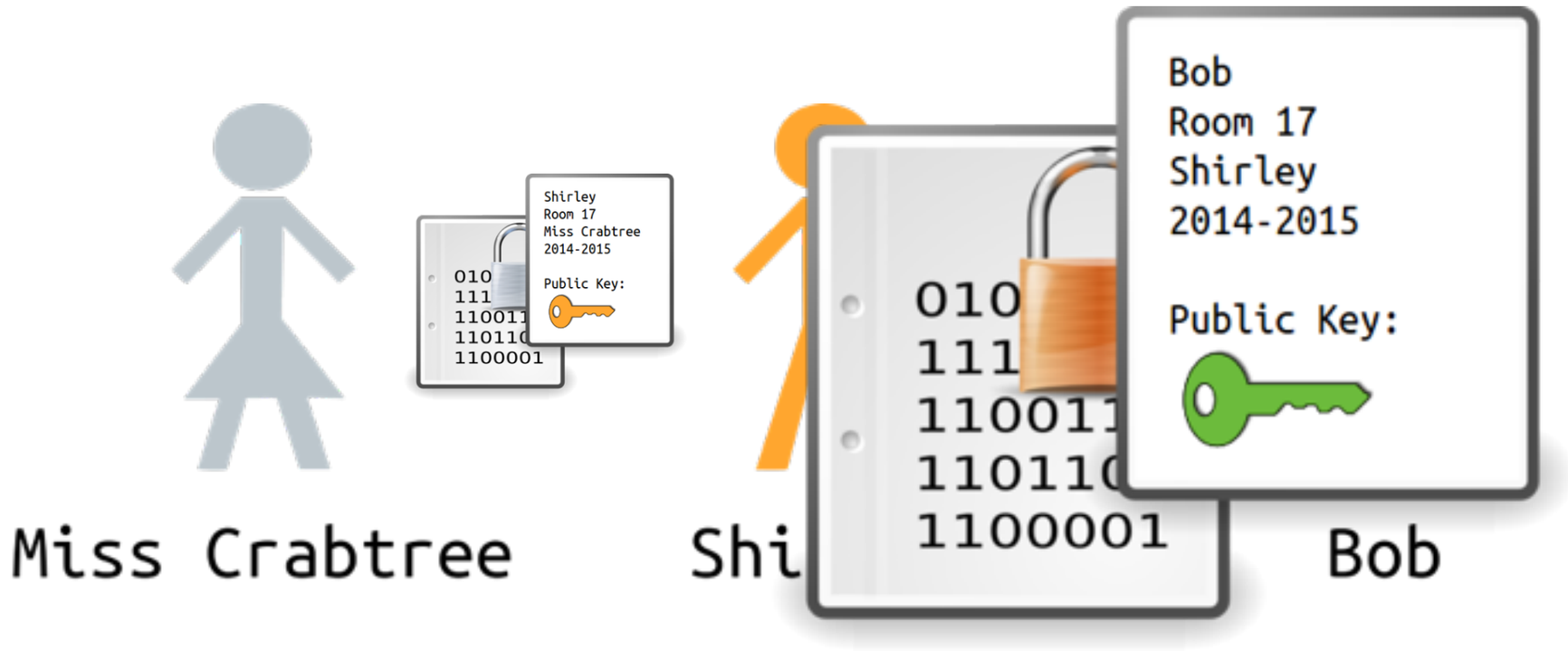
Miss Crabtree Shirley Bob

Certificates

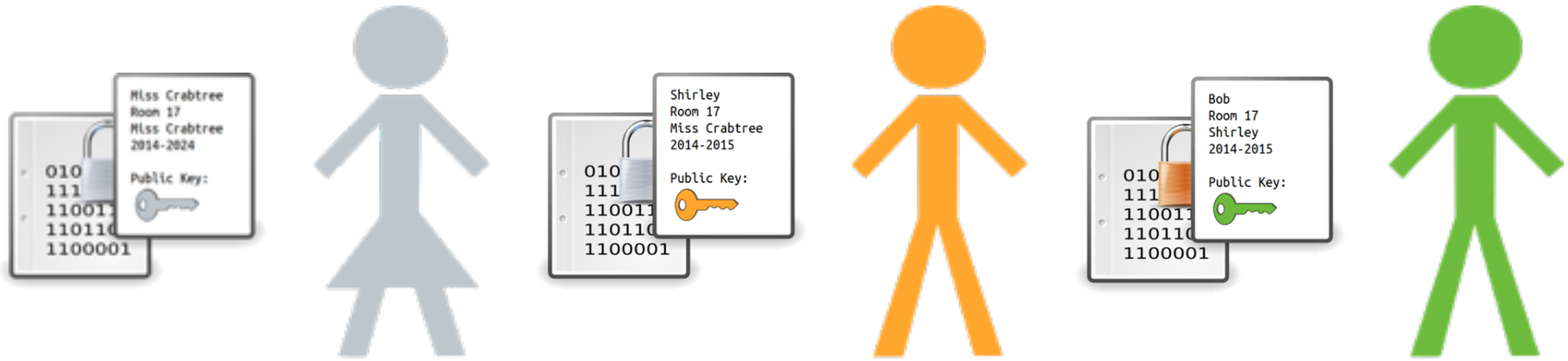


Intermediate Certificate Authority

Certificates

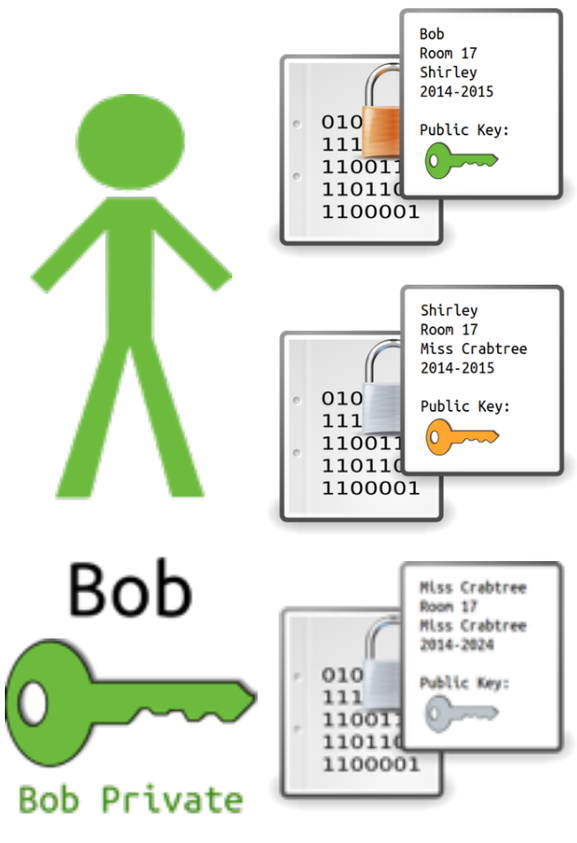


Certificates

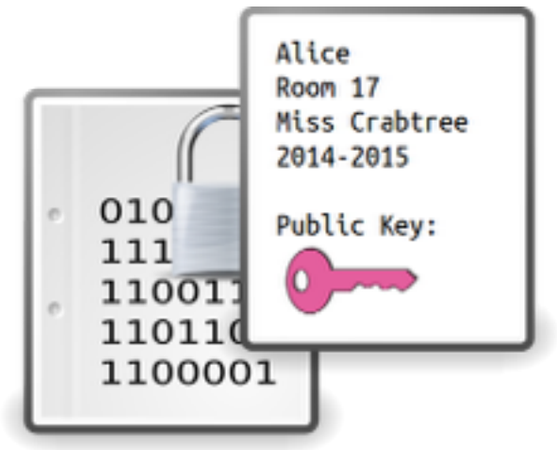


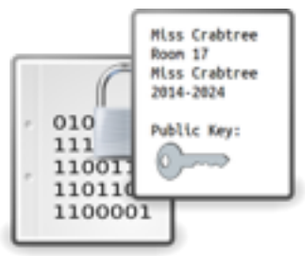
Certificate Chain


Shirley
Room 17
Miss Crabtree
2014-2015
Public Key:
Miss Crabtree
Room 17
Miss Crabtree
2014-2024
Public Key:
Alice
Alice Private


Bob
Room 17
Shirley
2014-2015
Public Key:
Shirley
Room 17
Miss Crabtree
2014-2015
Public Key:
Miss Crabtree
Room 17
Miss Crabtree
2014-2024
Public Key:
Bob
Bob Private

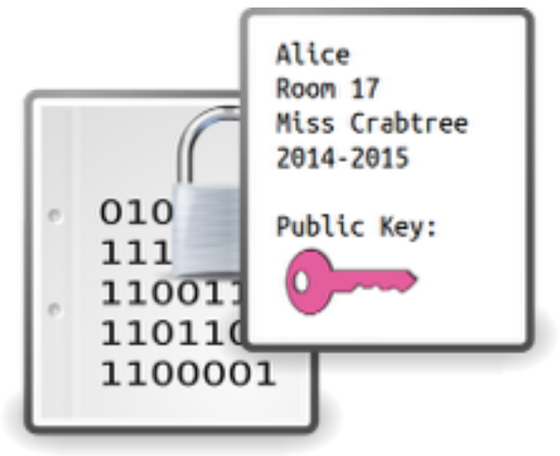
PKI

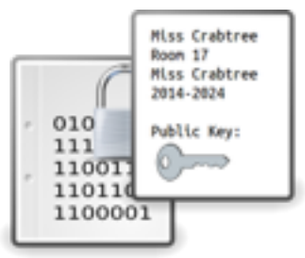




- Certificate Signing Request (CSR)
- Certificate Authority
- Intermediate Certificate Authority
- Trusted Root Certificates
- *Cross-signed, Bridge CA, Proxy Certificates*

PKI



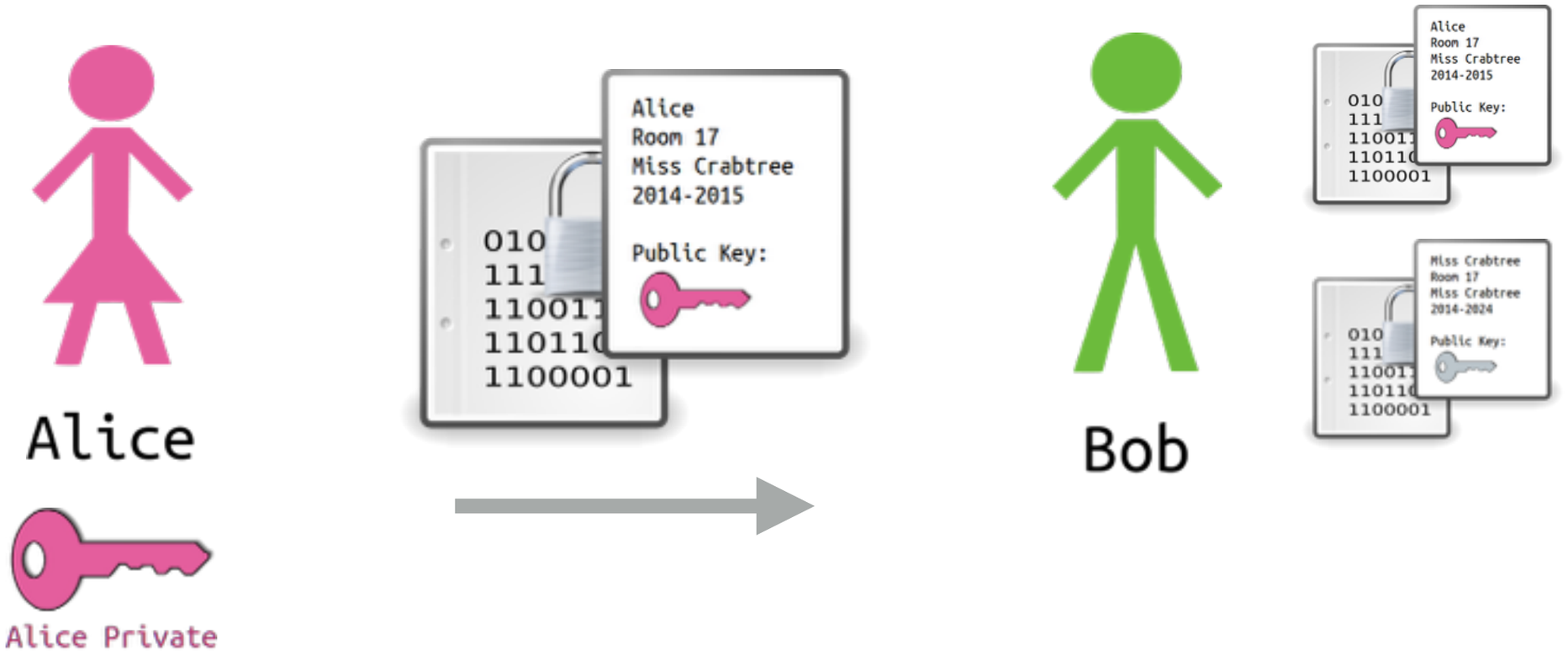


- Certificate can be revoked by Issuer
- You have to check it yourself
  - Certificate Revocation List (CRL)
    - CRL is a text file
    - CRL Distribution Point
  - Online Certificate Status Protocol (OCSP)
    - OCSP Responder
    - OCSP Stapling

Transport Layer Security (TLS)

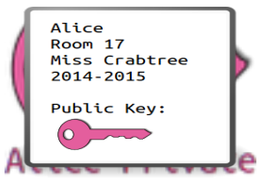
TCP/IP Connection

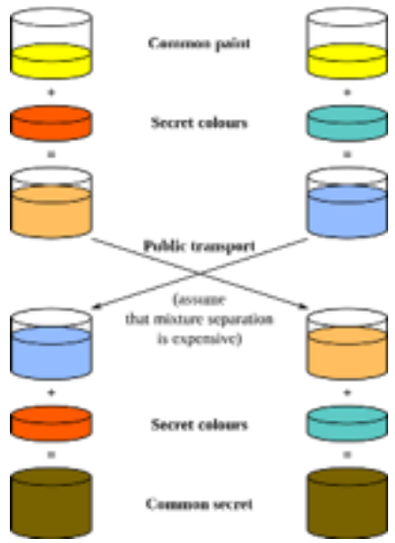
TLS


Alice
Room 17
Miss Crabtree
2014-2015
Public Key:
010
111
110011
110110
1100001
Alice
Alice Private
Bob
Alice
Room 17
Miss Crabtree
2014-2015
Public Key:
Miss Crabtree
Room 17
Miss Crabtree
2014-2024
Public Key:

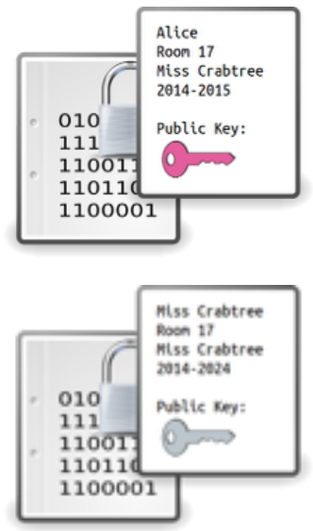
TLS

Alice





Bob



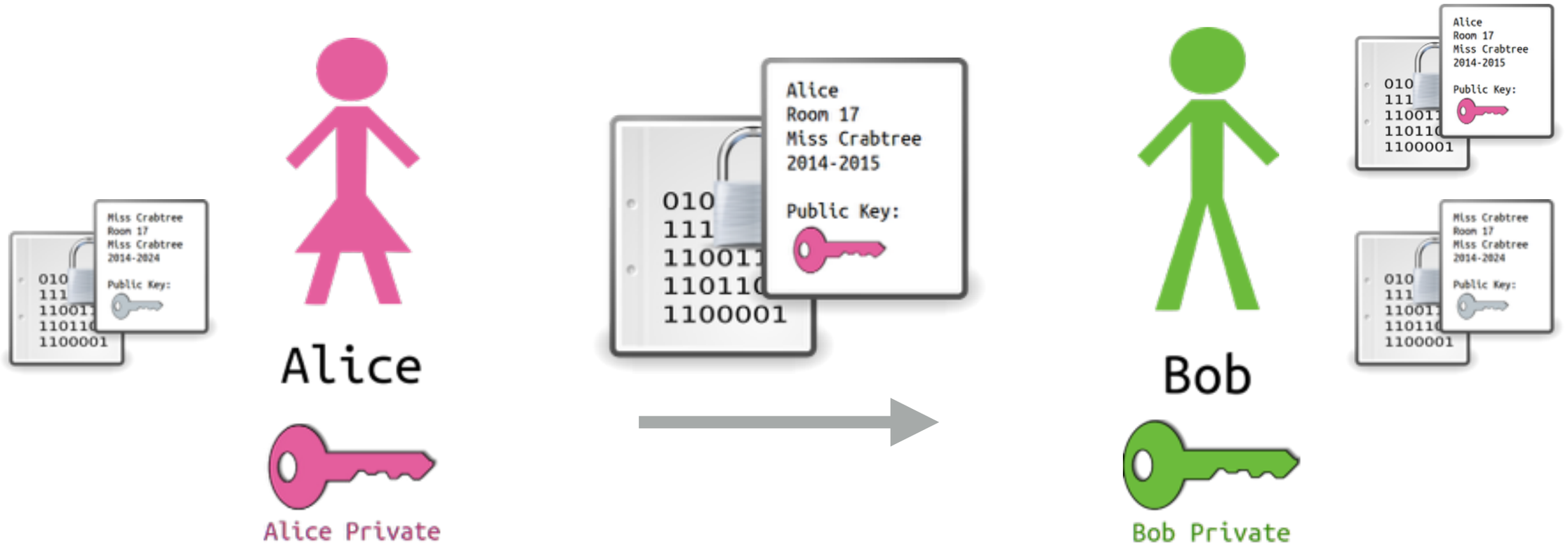
Key Exchange

TLS

Encrypted Connection

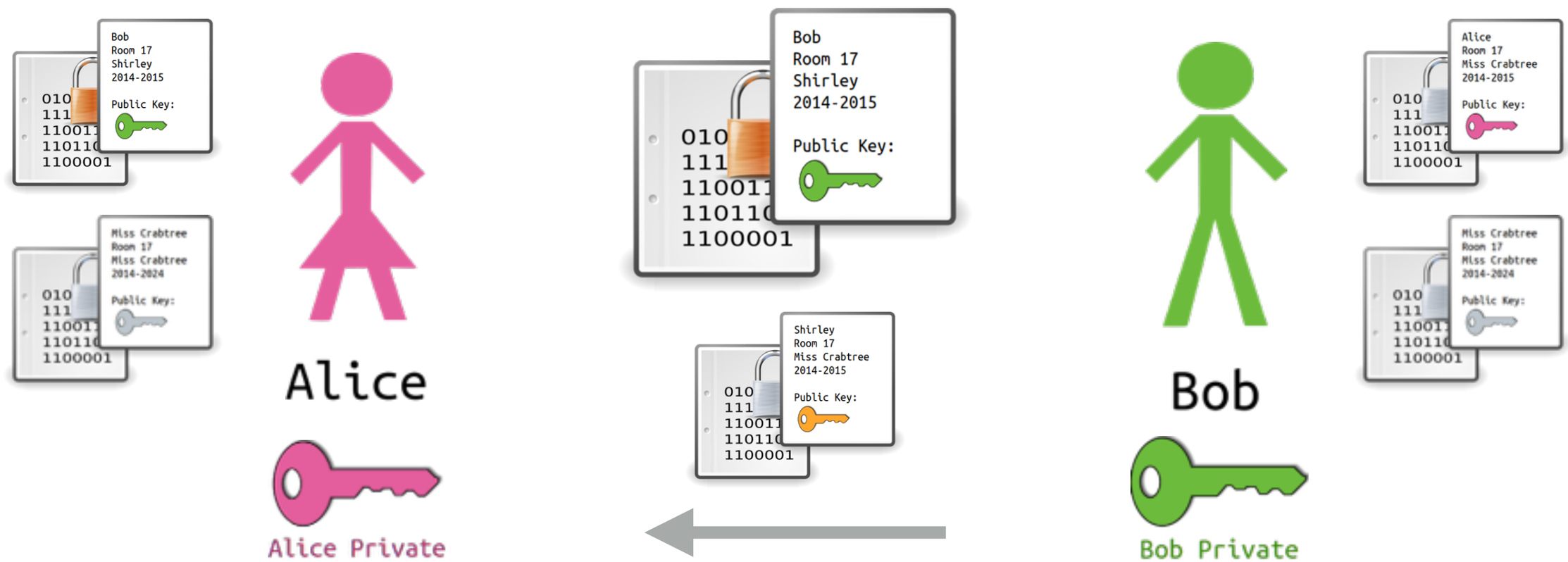
Transport Layer Security (TLS)

TLS

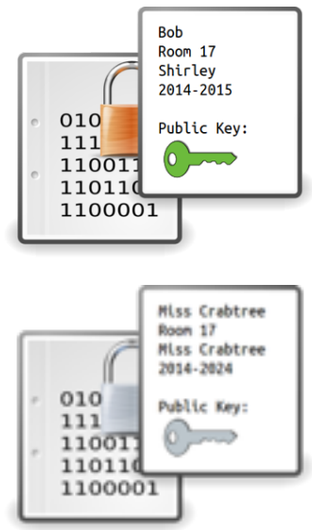

Alice
Room 17
Miss Crabtree
2014-2015
Public Key:
Alice
Alice Private
Bob
Bob Private

Transport Layer Security (TLS)

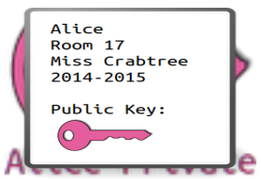
TLS

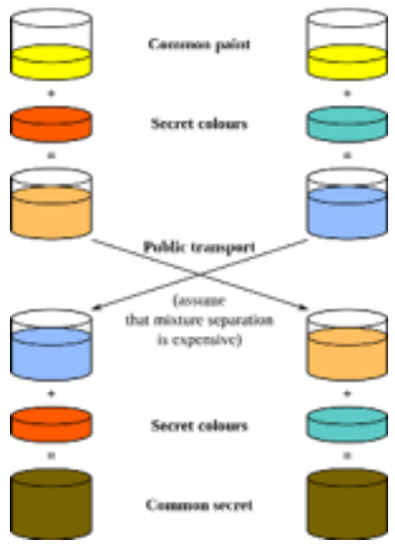

Bob
Room 17
Shirley
2014-2015
Public Key:
Miss Crabtree
Room 17
Miss Crabtree
2014-2024
Public Key:
Alice
Alice Private
Shirley
Room 17
Miss Crabtree
2014-2015
Public Key:
Alice
Room 17
Miss Crabtree
2014-2015
Public Key:
Miss Crabtree
Room 17
Miss Crabtree
2014-2024
Public Key:
Bob
Bob Private

TLS

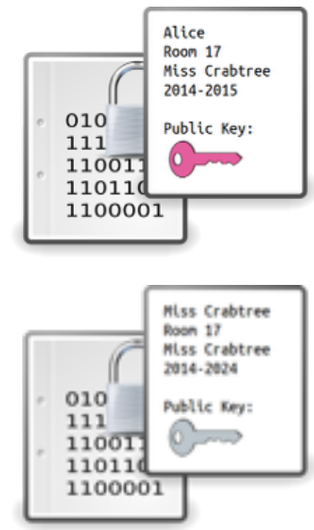


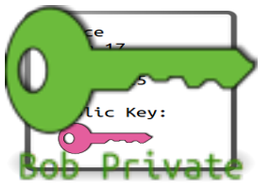
Alice





Bob





Key Exchange

TLS

Encrypted Connection

TLS

- Transport Layer Security (TLS)
  - Authentication with certificates
  - Privacy with encryption
  - Integrity with tamper-detection
    - TLS uses MAC-then-encrypt
- TLS is based on and replaces SSL (Secure Sockets Layer)

Security Tools

- Symmetric-Key Encryption
- Public-Key Cryptography
- Digital Signatures
- Certificates / PKI
- TLS (SSL)

Security Examples

Secure Email	Code Signing
Web Server HTTPS	Apple ID

Secure Email

- POP/IMAP over TLS (SSL)
- S/MIME (Secure/Multipurpose Internet Mail Extensions)
  - Apple Mail
  - Thunderbird
  - Outlook
- PGP (Pretty Good Privacy)
  - OpenPGP, GPG
  - Same idea with different system of certificates

Secure Email

- You'll need a certificate used for emailing
  - S/MIME uses X.509 certificates
- Your email address goes in the certificate
  - In olden days: "emailAddress" in Subject
  - Today: as a Subject Alternate Name (SAN)
    - Email alternate name
- *Particular key usage flags may be needed*

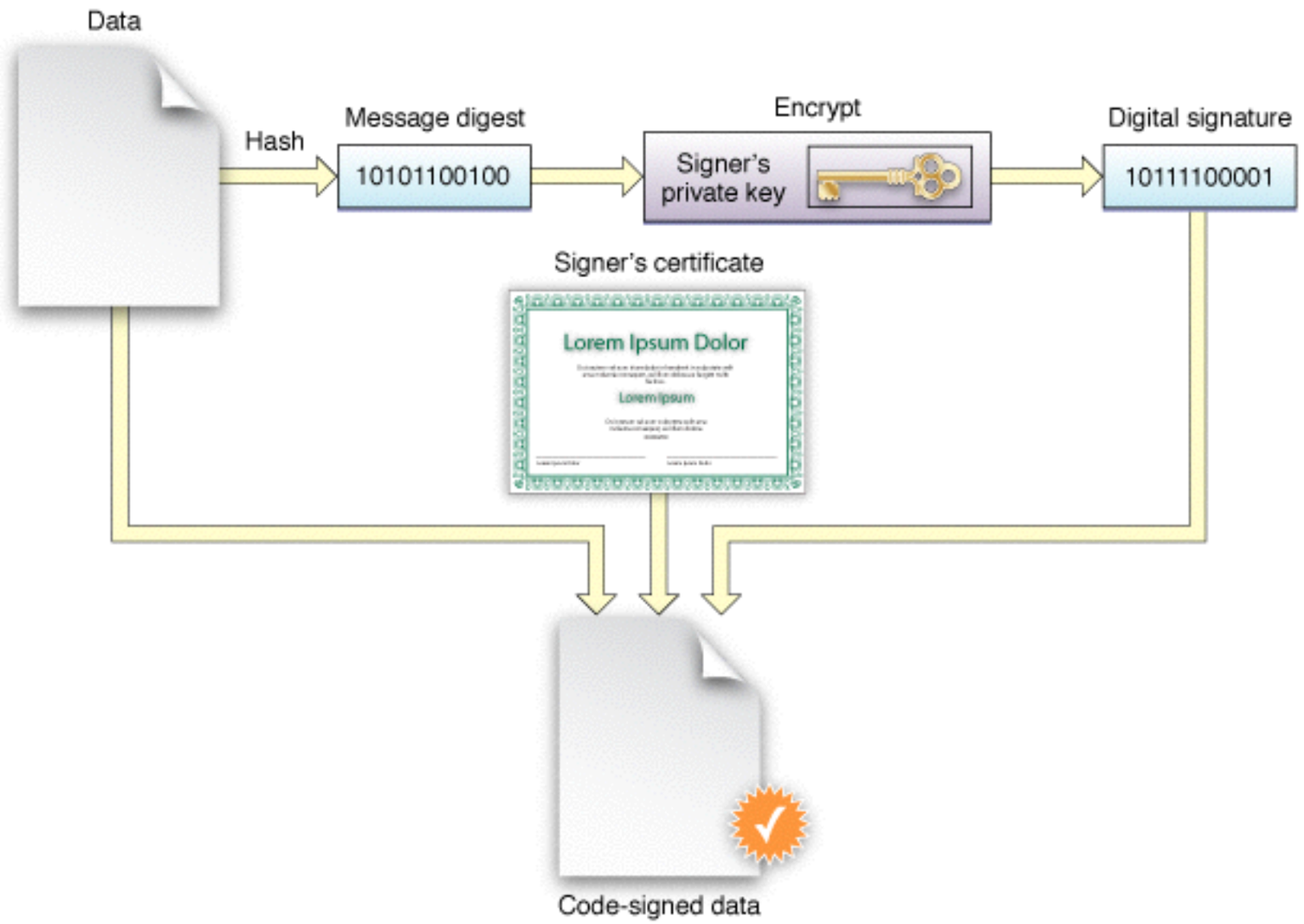
MACTECH

Using the Tools

Secure Email	Code Signing
Web Server HTTPS	Apple ID

MACTECH

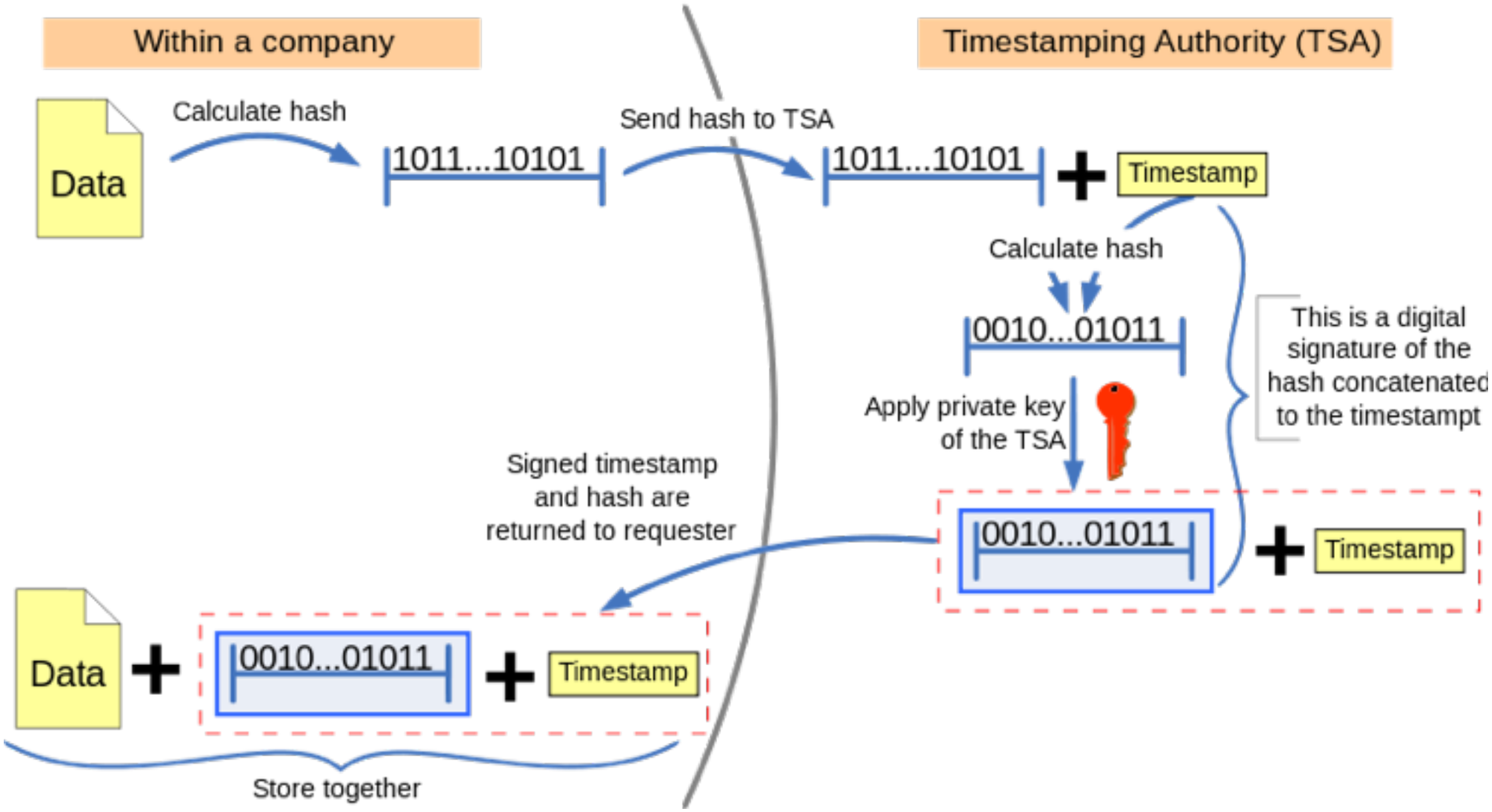
Code Signing



Code Signing

- You should include a trusted timestamp
  - From a Time Stamping Authority (TSA) during the code-signing step of your build
  - Your code remains trusted after your code-signing certificate has expired

Code Signing



Using the Tools

Secure Email	Code Signing
Web Server HTTPS	Apple ID

MACTECH

Web Server HTTPS

- HTTPS on port 443 instead of HTTP port 80
- HTTPS uses TLS (SSL)
  - Server should use at least TLS v1.0
  - Less common are TLS v1.1 and TLS v1.2
  - Some still use SSL v3
  - Nobody uses SSL v2—it's no longer secure

Web Server HTTPS

- Server certificate is X.509
- Use of client certificates is not common
  - Browser support is clunky :-(
- Fully-qualified domain name of server:
  - In olden days: Common Name of Subject
  - Today: as Subject Alternate Name (SAN)
    - DNS alternate name
    - Can have more than one listed
- *Particular key usage flags may be needed*

Web Server HTTPS

- Web server and browser negotiate the kind of symmetric encryption key cipher to use
- Your server should offer only the best choices
- https://www.ssllabs.com/ssltest/viewMyClient.html

Using the Tools

Secure Email	Code Signing
Web Server HTTPS	Apple ID

MACTECH

Apple ID

- With an Apple ID you get an X.509 client certificate
- You can see this certificate using Keychain Access
  - Finder > Applications > Utilities > Keychain Access
- Look in the login keychain
  - You'll find the certificate and next to it, a private key
  - Nobody else has this private key
  - It's likely an RSA 2048-bit key (mine is)
  - It has various usage flags
- You can see it was issued by *Apple Application Integration Certification Authority*, an intermediate CA, itself issued by *Apple Root CA*

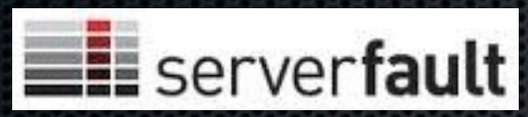
Using the Tools

Secure Email	Code Signing
Web Server HTTPS	Apple ID

Tools

- Symmetric-Key Encryption
- Public-Key Cryptography
- Digital Signatures
- Certificates / PKI
- TLS (SSL)

Resources



- stackoverflow.com
- serverfault.com
- security.stackexchange.com
  - crypto.stackexchange.com
  - apple.stackexchange.com
- IETF RFCs
  - www.ietf.org
    - Easiest to google “rfc …”
      - “rfc pki”
      - “rfc s/mime”

MACTECH

Questions?

Jim Flood
Townsend Security
jim.flood@townsendsecurity.com