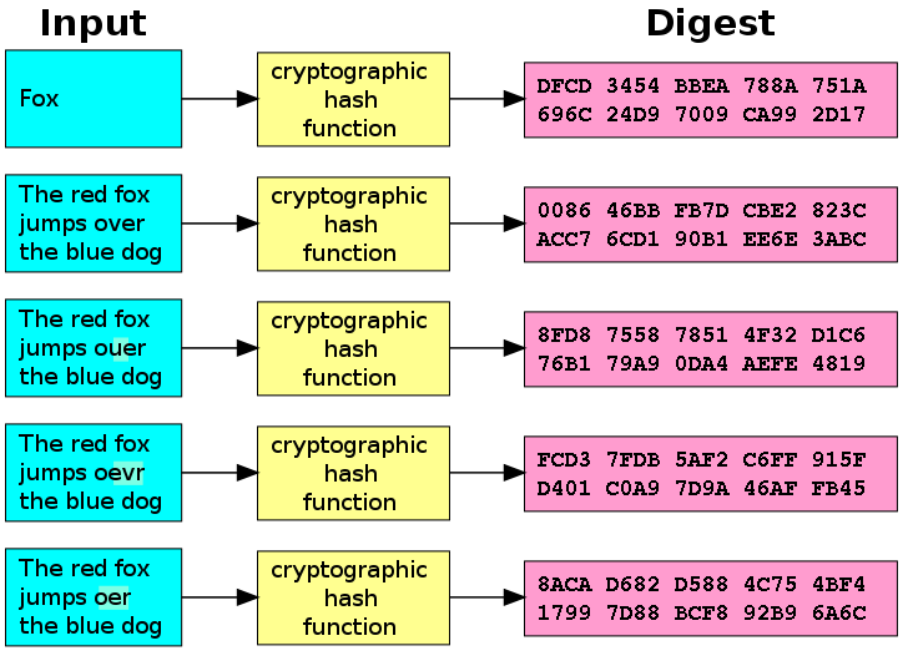
Security and authentication




```
root@topi:/etc# more shadow
root:$6$1z2.CqoJ$bIb7HOC7ByvSVcLmpc1C5F/H.gAddflg1xa2fQKnMAOabwZI1YSLDiK2gIKuEbeo
uGj33w8H4QDiWYvamlfIj2eu.:15138:0:99999:7:::
daemon:*:15040:0:99999:7:::
keithwork:$6$CRDEfvR2Q$B8.0J5P/7TvualkFfAFfe5a234.GgnFBGRfHKb6.jpTN223ZMja0ILte
1FoE6vzlf7Rt/eiNBSqfeegEVxs33fe#f7x0:15135:0:99999:7:::
mysql:!:15087:0:99999:7:::
httpd:!:15133:0:99999:7:::
keithbackup:$6$whkE4GJT$yUMfE4gYwhp656rNqv/7see8y5aF/Vgra3FUe.g4Facg4Iug4vyJLg4F
bgeZW0i7feqMPCHQpBsJi/:15164:0:99999:7:::
```

Overview

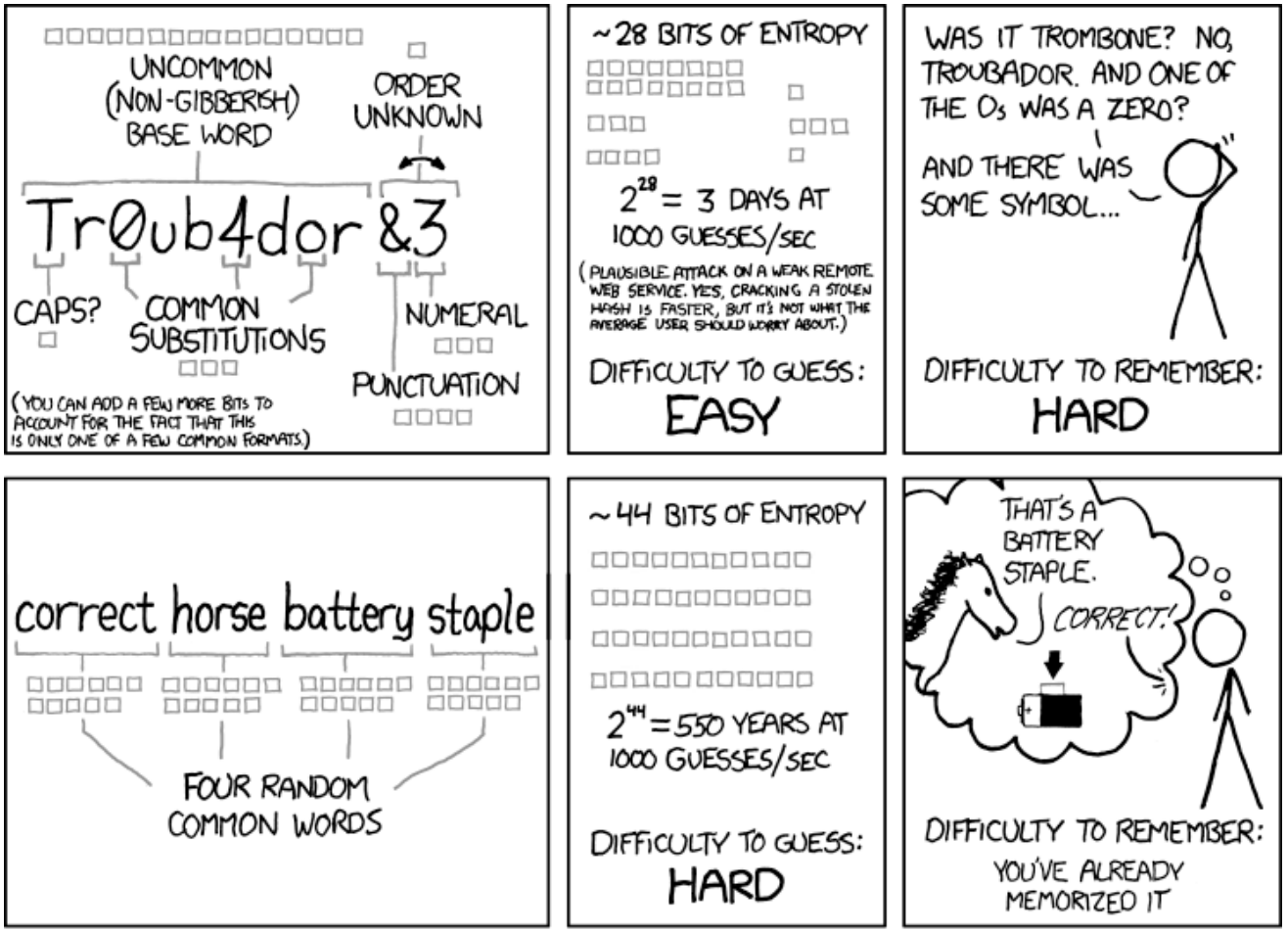
- Authentication
  - Passwords
    - One-way hashing
    - Salting passwords
  - Other forms: tokens, biometrics
  - Digital signing
    - Public key based signing
    - PKI, CA
- Pretty Good Privacy (PGP)
- Securing web commerce
  - SSL / TLS
  - https

Authentication

- Proving your identify
  - Something you know: password, PIN, pet's name
  - Something you possess: a key, smart card
  - Something you are: fingerprints, retina, face
  - Something you do: voice print, handwriting, typing rhythm
- Means of authentication
  - Password
  - Token-based
  - Biometric

Password authentication

- Users choose some secret password
  - Differing levels of required complexity/annoyance



https://xkcd.com/936/

Password storage

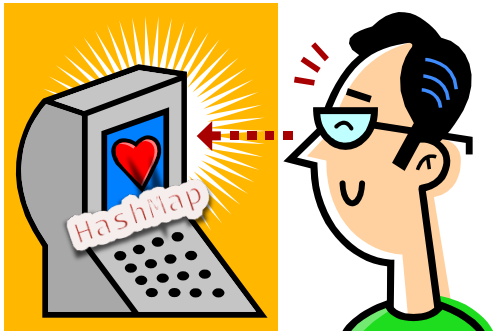
- User ID and password
  - Must be stored somewhere, e.g. `/etc/passwd`
  - Shadow password file, e.g. `/etc/shadow`
    - Reachable only by privileged users

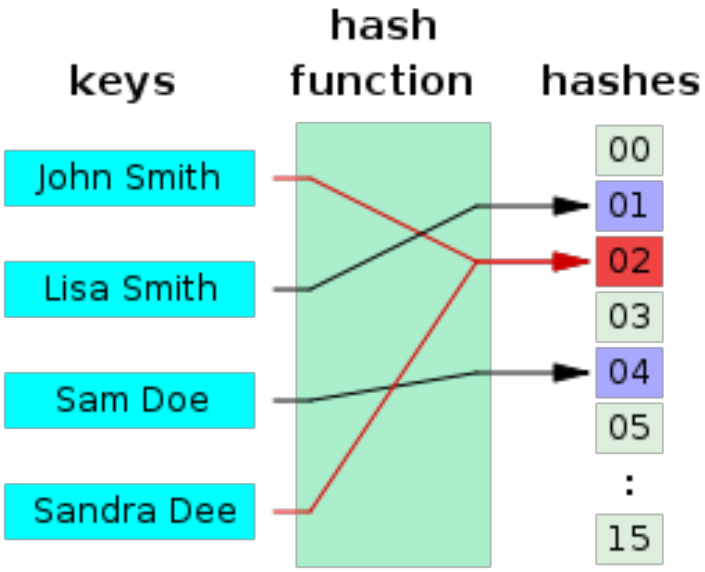
```
% more passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
work:x:1000:1000::/home/work:/bin/sh
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
httpd:x:1001:1001::/home/httpd:/bin/sh
backup:x:1005:1005::/home/backup:/bin/sh
```

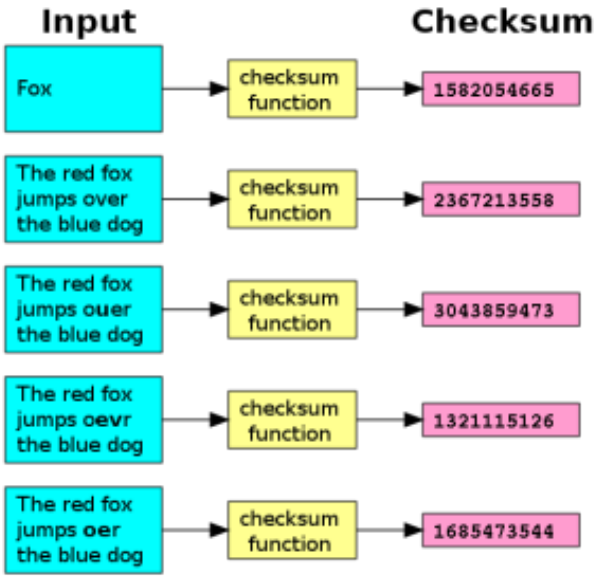
```
% more shadow
root:$6$1z2.CqoJ$bIb7HOC7ByvSVcLmpc1C5F/H.gAddflg1xa2fQKnMAOabwZI1YSLDiK2gIKuEbeo
uGj33w8H4QDiWYvamlfIj2eu.:15138:0:99999:7:::
daemon:*:15040:0:99999:7:::
work:$6$CRDEfvR2Q$B8.0J5P/7TvualkFfAFfe5a234.GgnFBGRfHKb6.jpTN223ZMja0ILte
1FoE6vzlf7Rt/eiNBSqfeegEVxs33fe#f7x0:15135:0:99999:7:::
mysql:!:15087:0:99999:7:::
httpd:!:15133:0:99999:7:::
backup:$6$whkE4GJT$yUMfE4gYwhp656rNqv/7see8y5aF/Vgra3FUe.g4Facg4Iug4vyJLg4F
bgeZW0i7feqMPCHQpBsJi/:15164:0:99999:7:::
```

Hashing

- Normal hash functions:
  - Key: large data set of variable length
  - Value: small data set of fixed length
  - Examples:
    - Error checking: checksum, CRC
    - Constant time data structures: Java HashMap

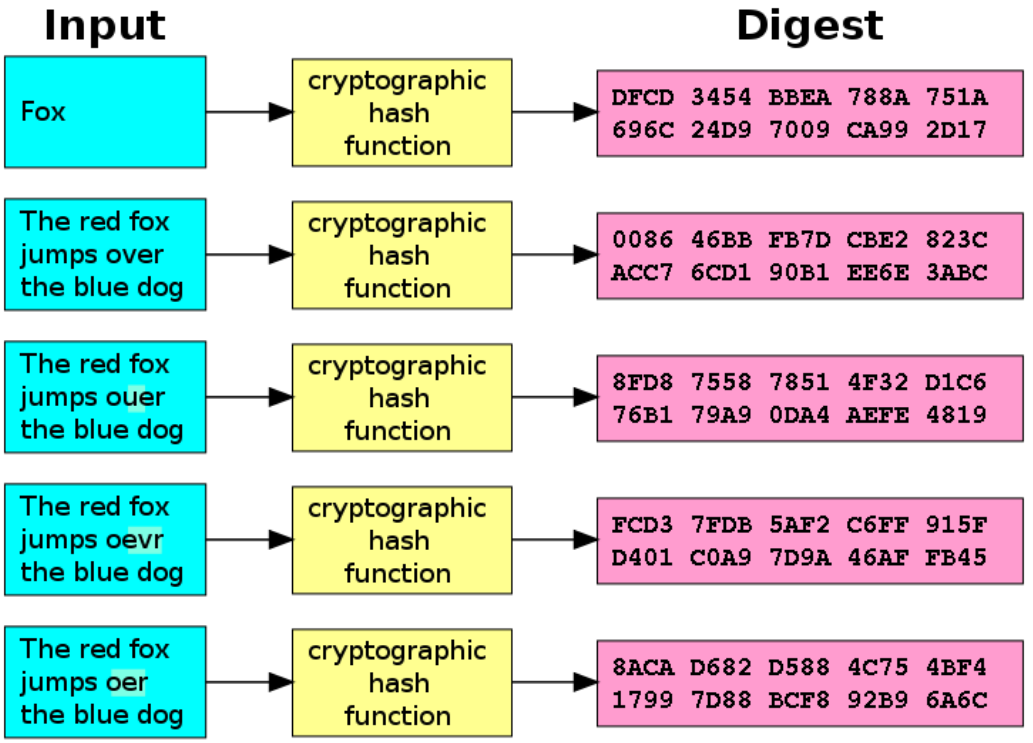






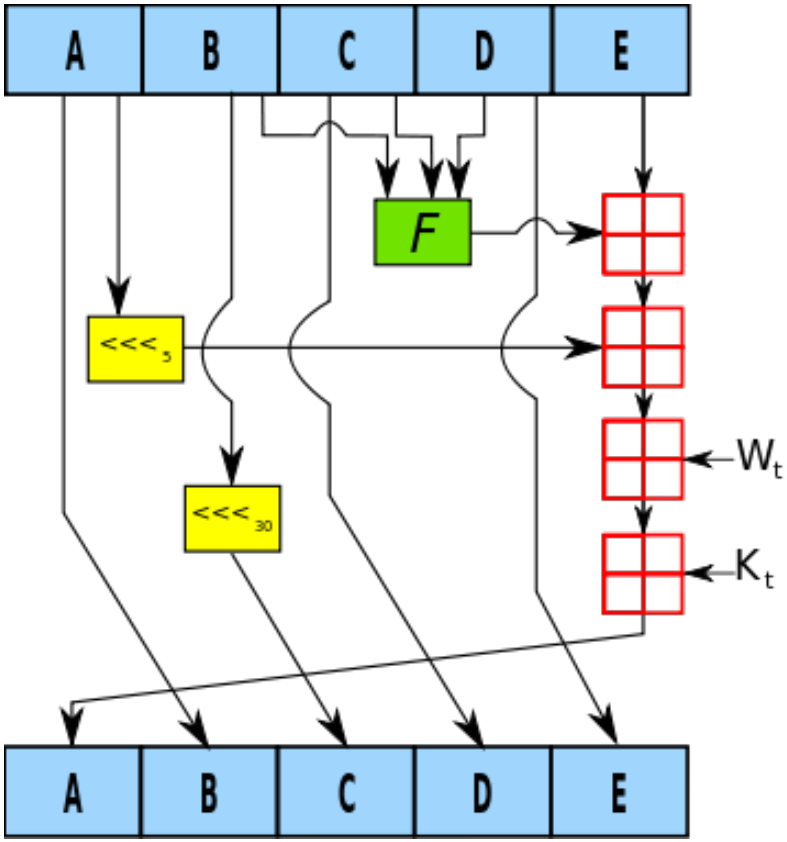
Secure hashing

- Secure hash functions:
  - Hash H(x) easy to compute for x
  - One-way: given h=H(x) intractable to find x
    - e.g. MD5 (128 bits), SHA-1 (160 bits), SHA-256 (256 bits), SHA-512 (512 bits)



Secure hashing

- Desirable properties
  - Pre-image resistant, one-way
    - For any code h, intractable to find x s.t. H(x) = h
  - 2nd pre-image resistant, weak collision resistant
    - For any block x, intractable to find y ≠ x s.t. H(y) = H(x)
  - Strong collision resistant
    - Intractable to find any pair (x, y) s.t. H(x) = H(y)
- Uses:
  - One-way encryption of passwords: store only hash
  - Intrusion detection: detect changes in file
  - Digital signing of messages



History:

- 1991: MD5 by Ron Rivest
- 1993: SHA-0
- 1995: Revised version, SHA-1
- 1996: Flaw found in MD5
- 2005: Attack found on SHA-1
- 2010: NIST required federal agencies to move to SHA-2
- 2012: NIST selects Keccak for SHA-3

http://blogs.technet.com/b/srd/archive/2012/06/06/more-information-about-the-digital-certificates-used-to-sign-the-flame-malware.aspx

http://www.win.tue.nl/hashclash/rogue-ca/

One iteration within the SHA-1 compression function:

- A, B, C, D and E are 32-bit words of the state
- F is a nonlinear function that varies
- $<<<_n$ denotes a left bit rotation by n places
- n varies for each operation
- W_t is the expanded message word of round t
- K_t is the round constant of round t
- Box with plus denotes addition modulo 2^{32}

Attacking passwords

- If hashed passwords compromised, attacker:
  - Knows users with same password
  - Can tell if user has same password on multiple systems (if using same hash function)
  - Can use an offline dictionary attack
- Dictionary attack:
  - Precompute hash value for:
    - All sequences of a given (short) length
    - Common words
  - Check for match against hash in password file

Salt

- Salting passwords
  - On account creation, assign salt value:
    - Timestamp, random value, ...
  - Salt stored unencrypted, associated with user ID
  - Hash computed from salt plus user's password
  - Makes dictionary attack much more expensive

```
% more shadow
root:$6$1z2.CqoJ$bIb7HOC7ByvSVcLmpc1C5F/H.gAddflg1xa2fQKnMAOabwZI1YSLDiK2gIKuEbeo
uGj33w8H4QDiWYvamlfIj2eu.:15138:0:99999:7:::
daemon:*:15040:0:99999:7:::
work:$6$CRDEfvR2Q$B8.0J5P/7TvualkFfAFfe5a234.GgnFBGRfHKb6.jpTN223ZMja0ILte
1FoE6vzlf7Rt/eiNBSqfeegEVxs33fe#f7x0:15135:0:99999:7:::
mysql:!:15087:0:99999:7:::
httpd:!:15133:0:99999:7:::
backup:$6$whkE4GJT$yUMfE4gYwhp656rNqv/7see8y5aF/Vgra3FUe.g4Facg4Iug4vyJLg4F
bgeZW0i7feqMPCHQpBsJi/:15164:0:99999:7:::
```

$6 = SHA-512

Improving password security

- Reactive password checker:
  - System attacks itself, revokes guessed passwords
    - But system has to do an expensive amount of work
- Proactive password checker:
  - Users selects a candidate password
  - System checks if allowable
  - Hopefully guide users to secure choice
    - Without *too* much annoyance
  - But different systems have:
    - Different min/max lengths, allowed symbols, case rules, number rules, ...

Improving password security

- User education
  - Encourage/force longer more complex passwords
    - e.g. Users often mistakenly believe reversing word makes password unguessable
  - Use first letter of personal phrase
    - "My dog's first name is Rex" -> "MdfniR"
  - Use random collection of words
    - "correcthorsebatterystaple"
- Computer-generated passwords
  - Normally low acceptance, users write them down
  - Generate pronounceable syllables, FIPS PUB 181

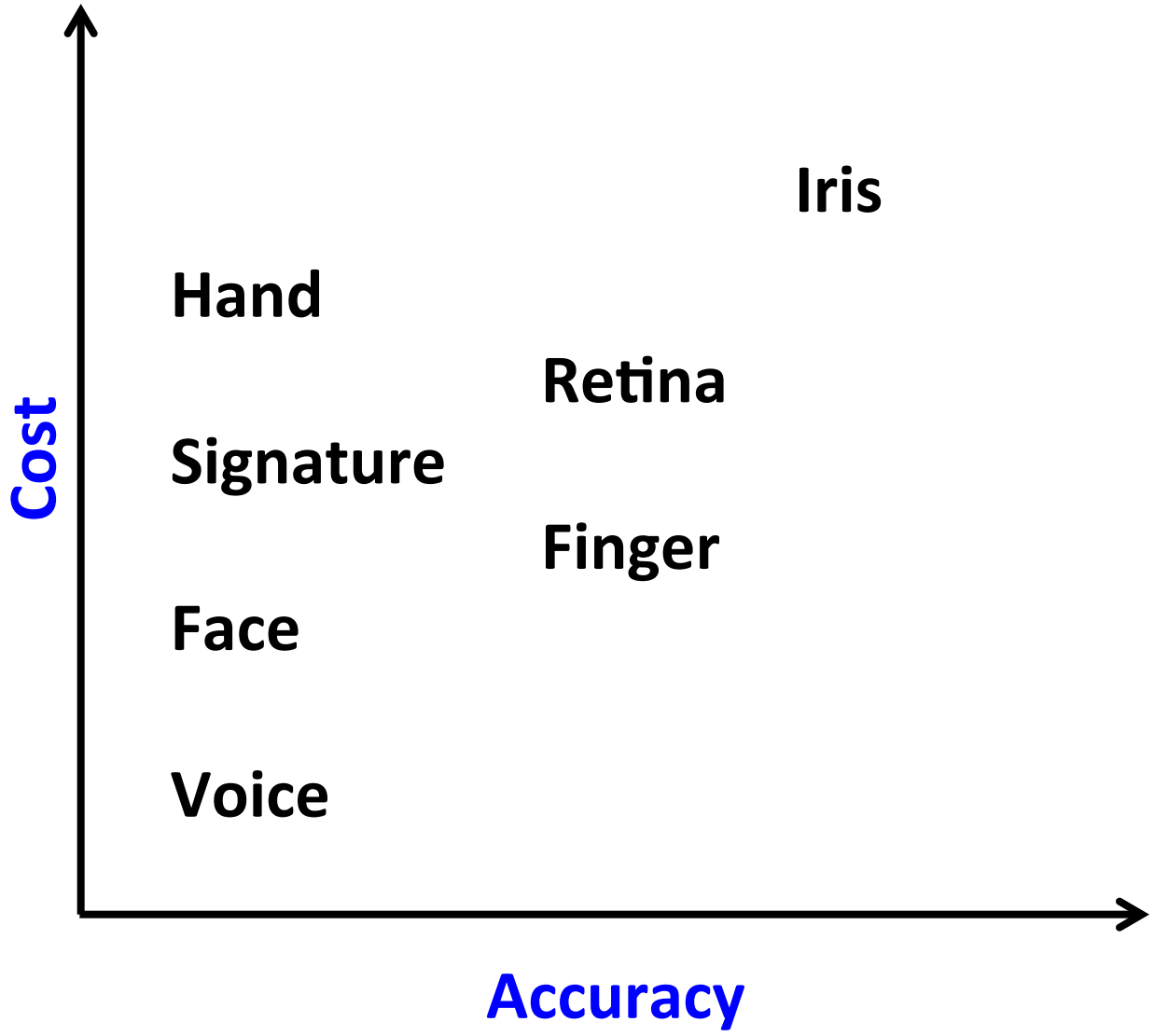
Token-based authentication

- Require users possess some object
  - Unique ID based
    - Magnetic strip, embedded microprocessor, ...
    - e.g. ATM card, mobile phone (two factor)
- Often in combination with user knowledge
  - e.g. ATM PIN

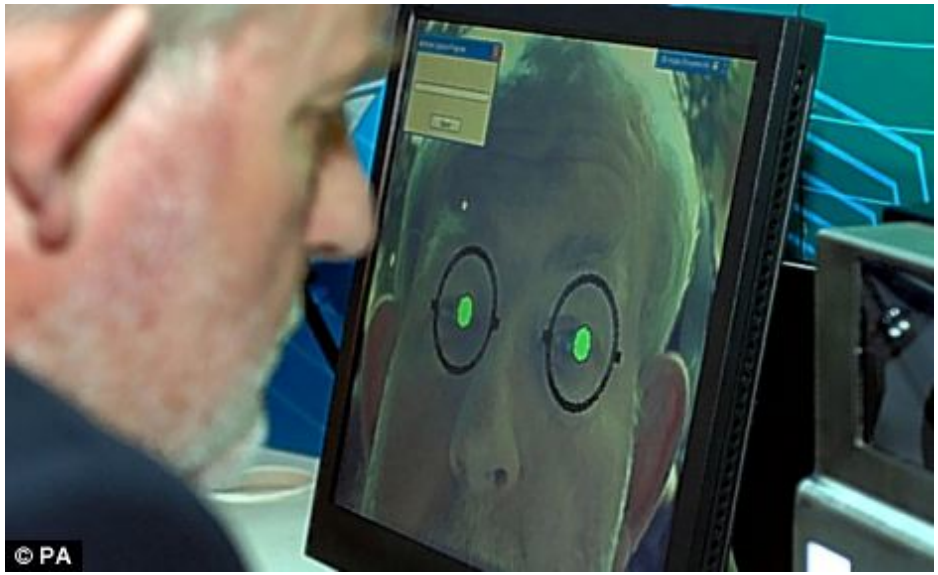
Biometric authentication

- Pattern recognition based
  - Facial recognition: location of facial features
  - Fingerprints: ridges and furrows on fingertip
  - Hand geometry: shape, length, width of fingers
  - Retinal: veins beneath retinal surface
  - Iris: structure of the iris
  - Signature: style of handwriting
  - Voice: patterns in speech signal
- Verification: proving you are who you say
- Identification: find out who you are

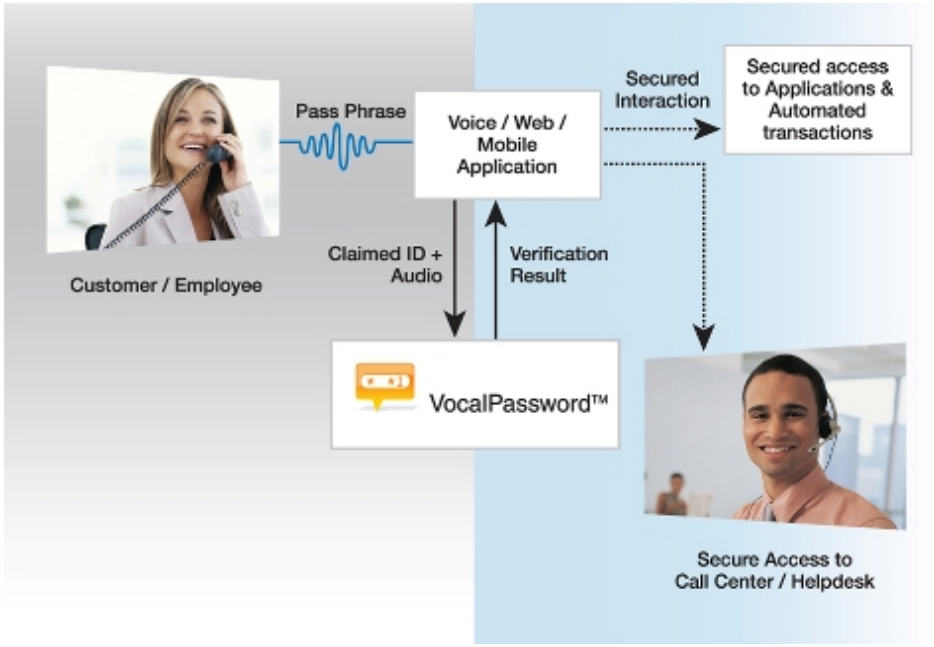
Biometric characteristics



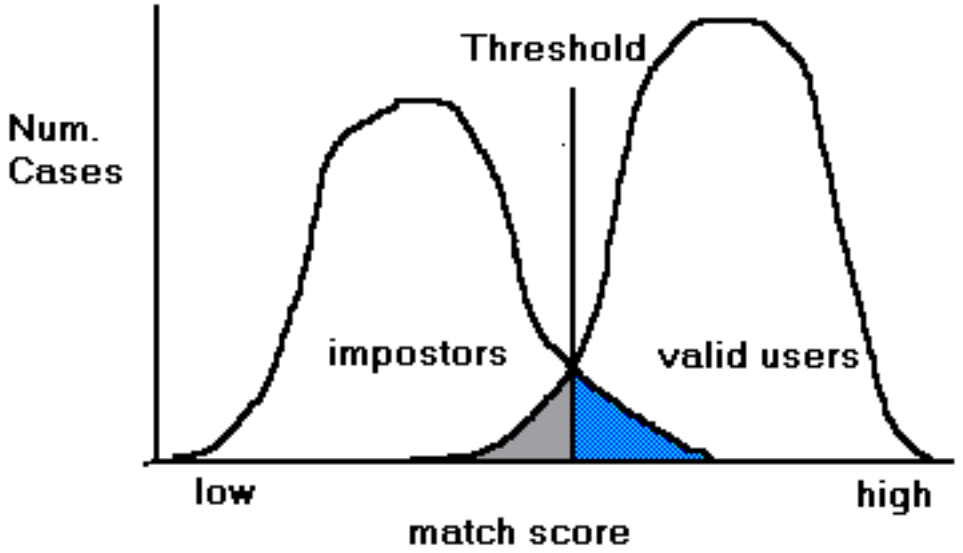
http://www.cl.cam.ac.uk/~jgd1000/UK_IRIS.png



http://i.dailymail.co.uk/i/pix/2012/02/17/article-0-11C5E3C4000005DC-346_468x286.jpg

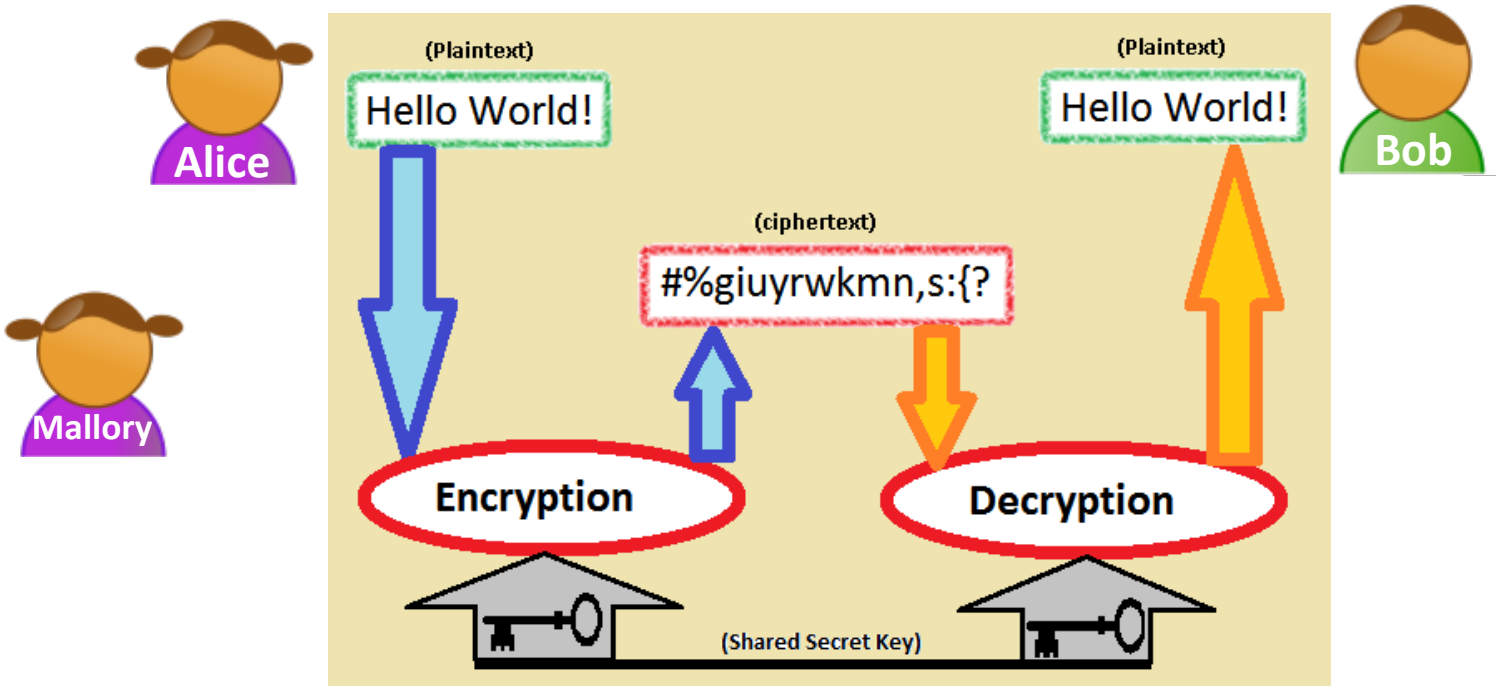


http://www.nuance.com/for-business/by-solution/customer-service-solutions/solutions-services/inbound-solutions/voice-authentication-biometrics/vocal-password/index.htm



http://www.cs.cmu.edu/afs/cs/Web/People/jeongue/jeon/Speaker%20Verification.htm

Digital signing

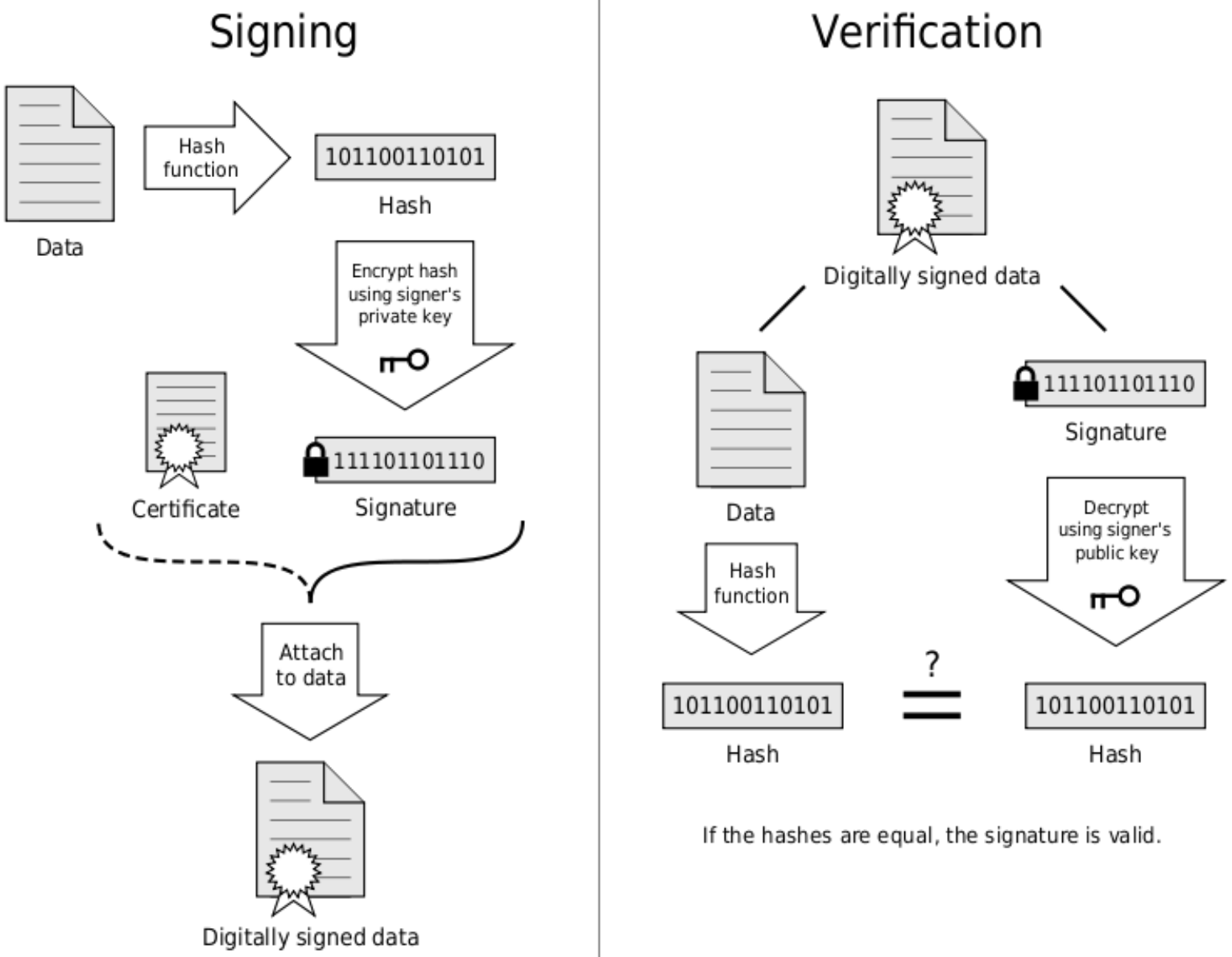


- Problem: impersonation in public-key crypto
  - Mallory encrypts message with Bob's public key
  - Only Bob can decrypt using his private key
  - Message is a love letter claiming to be from Alice

Digital signing

- Digital signing via public key crypto
  - Alice encrypts message with her private key
    - Everybody can decrypt using Alice's public key
    - But it proves it came from Alice since no one else has her private key
  - Encrypt result with Bob's public key
    - Only Bob can decrypt using his private key
  - Problem: asymmetric crypto can be expense
    - Hash the message
    - Encrypt just the hash

Hash-based digital signing

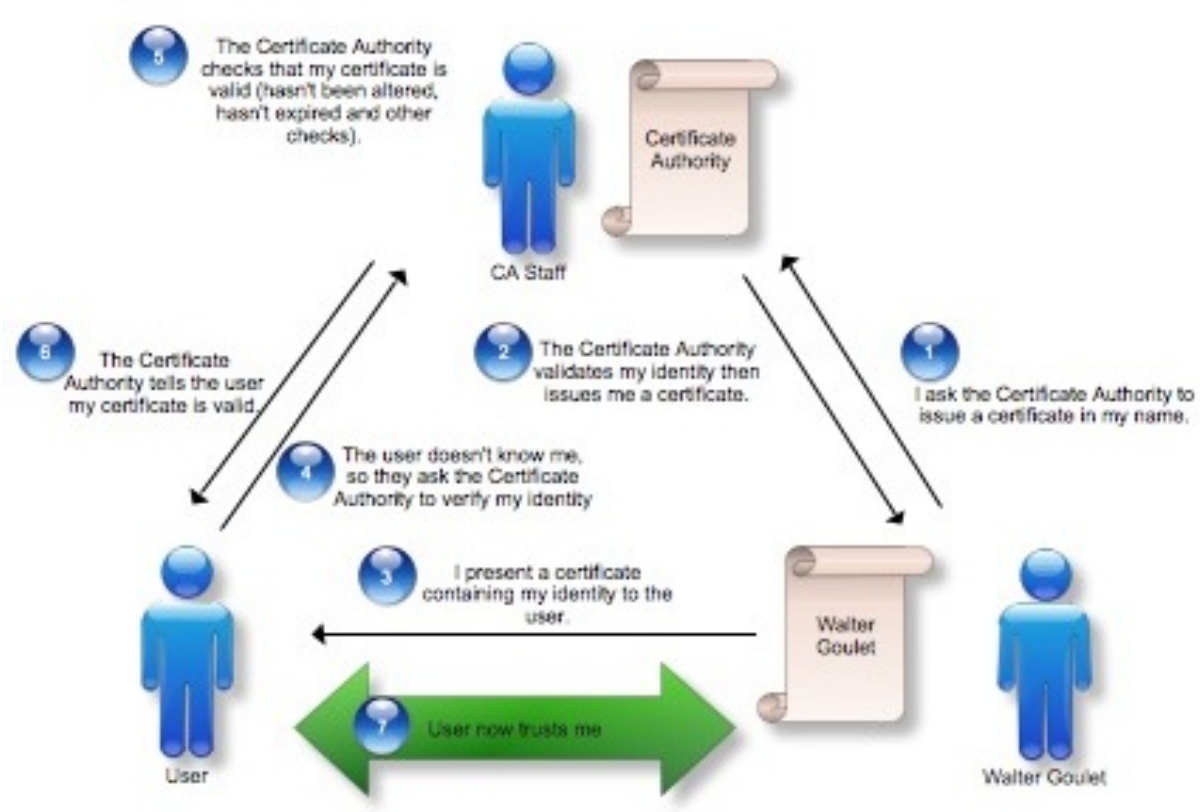


Distributing public keys

- Alice has to find Bob's public key
  - How does she know it is really Bob's key?
  - Someone else could impersonate Bob
    - Mallory fools Alice into using her fake Bob public key
    - Mallory decrypts using fake Bob's private key
    - Mallory reads message
    - Reencrypts using Bob's real public key and sends on
- Problems:
  - How do we distribute public keys?
  - How to establish trust of keys?

PKI

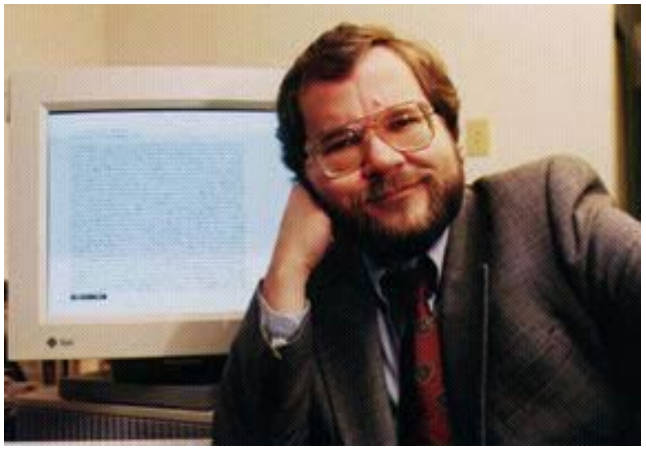
- Public Key Infrastructure (PKI)
  - Certificate Authority (CA)
    - Verifies user is who they say they are
    - Digitally signs the user's public key
    - e.g. VeriSign



http://blog.securism.com/2009/01/summarizing-pki-certificate-validation/

PGP

- Pretty Good Privacy (PGP)
  - 1991 Phil Zimmermann



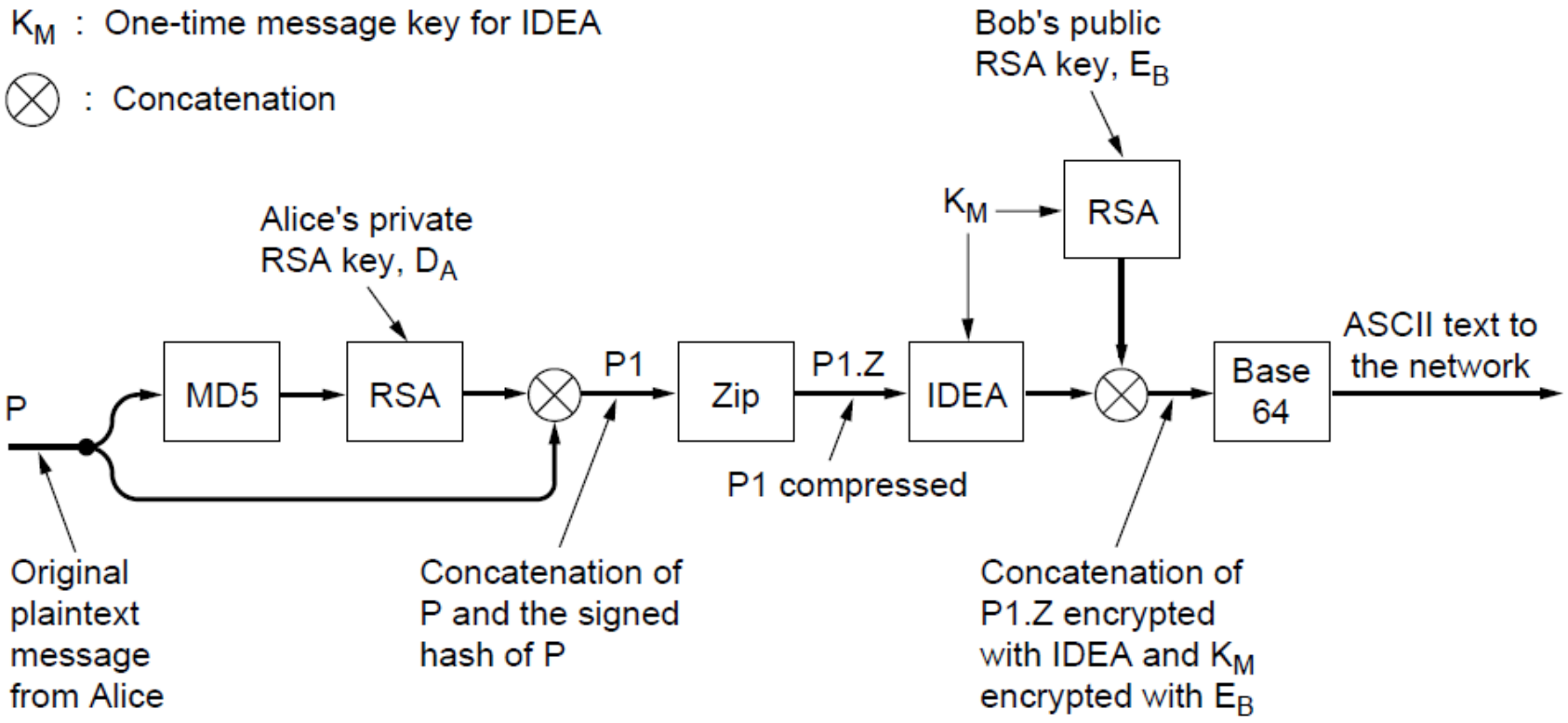
"In the past, if the Government wanted to violate the privacy of ordinary citizens, it had to expend a certain amount of effort to intercept and steam open and read paper mail, and listen to and possibly transcribe spoken telephone conversation. This is analogous to catching fish with a hook and a line, one fish at a time. Fortunately for freedom and democracy, this kind of labor-intensive monitoring is not practical on a large scale.

Today, electronic mail is gradually replacing conventional paper mail, and is soon to be the norm for everyone, not the novelty it is today. Unlike paper mail, E mail messages are just too easy to intercept and scan for interesting keywords. This can be done easily, routinely, automatically, and undetectably on a grand scale. This is analogous to driftnet fishing-- making a quantitative and qualitative Orwellian difference to the health of democracy."

-Philip Zimmermann, testimony to Congress

PGP

- Pretty Good Privacy (PGP)
  - Focus on efficiency:
    - RSA for symmetric key exchange
    - Symmetric cipher (IDEA) for bulk of encryption
  - Focus on ease of use:
    - Allow average Joe to use strong cryptography
    - User clicks to encrypt/sign an email
  - First widely available public-key crypto
    - Released via friend to the Usenet
  - Problems:
    - RSA was patented by RSA Data Security, Inc.
    - Strong encryption considered a munition by US



- Key length
  - `384 bits` = casual, broken easily today
  - `512 bits` = commercial, breakable by 3-letter orgs
  - `1024 bits` = military, not breakable on earth
  - `2048 bits` = alien, unbreakable on other planets

Securing web commerce

- Customer fills out order with credit card #
  - Problem 1: Keep data secure from customer's browser to the web server
  - Problem 2: Keep data secure on server or in transit to order fulfillment

SSL

- Secure Sockets Layer (SSL) / Transport Layer Security (TLS)
  - Client requests secure connection from server
  - Client sends list ciphers/hash functions supported
  - Server picks the strongest mutual cipher/hash
  - Server sends back digital certificate
    - Name of itself, trusted Certificate Authority (CA), public encryption key
  - Client contacts CA to confirm key belongs to site
  - Client generates session key by encrypting random number with server's public key
  - Client and server continue using symmetric cipher

HTTPS

- Hypertext Transfer Protocol Secure (HTTPS)
  - https://
  - Typically running on port 443

Application (HTTP)
Security (SSL)
Transport (TCP)
Network (IP)
Data link (PPP)
Physical (modem, ADSL, cable TV)

Summary

- Proving who you are
  - Passwords, tokens, biometrics
  - Digital signing using public key crypto
- Secure hash functions
  - Digital signing, storage of passwords, detecting changes in files
- PGP
  - Popular application of public key crypto
- Secure web commerce
  - SSL/TLS