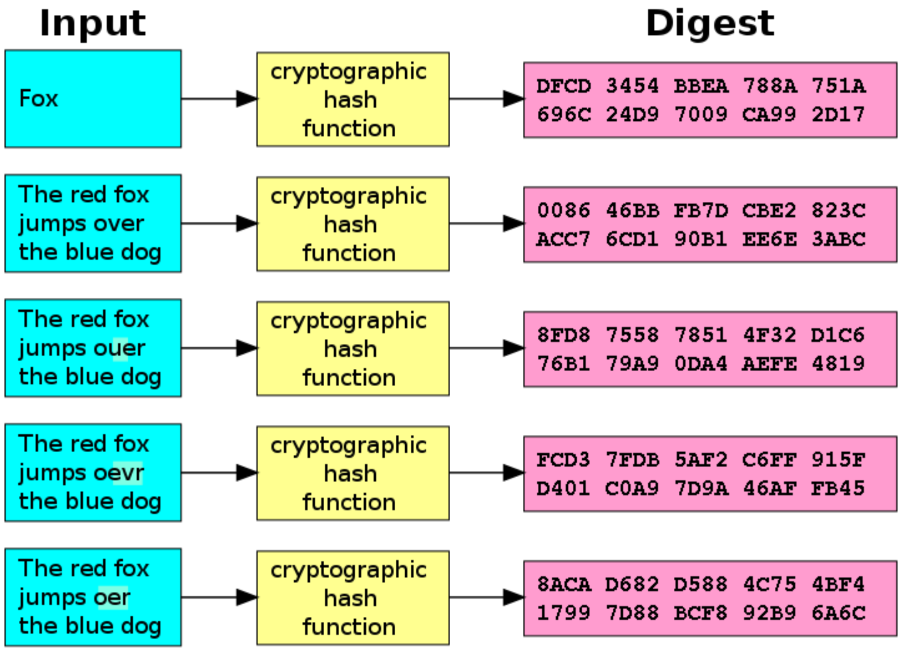
Security and authentication



Input

Digest

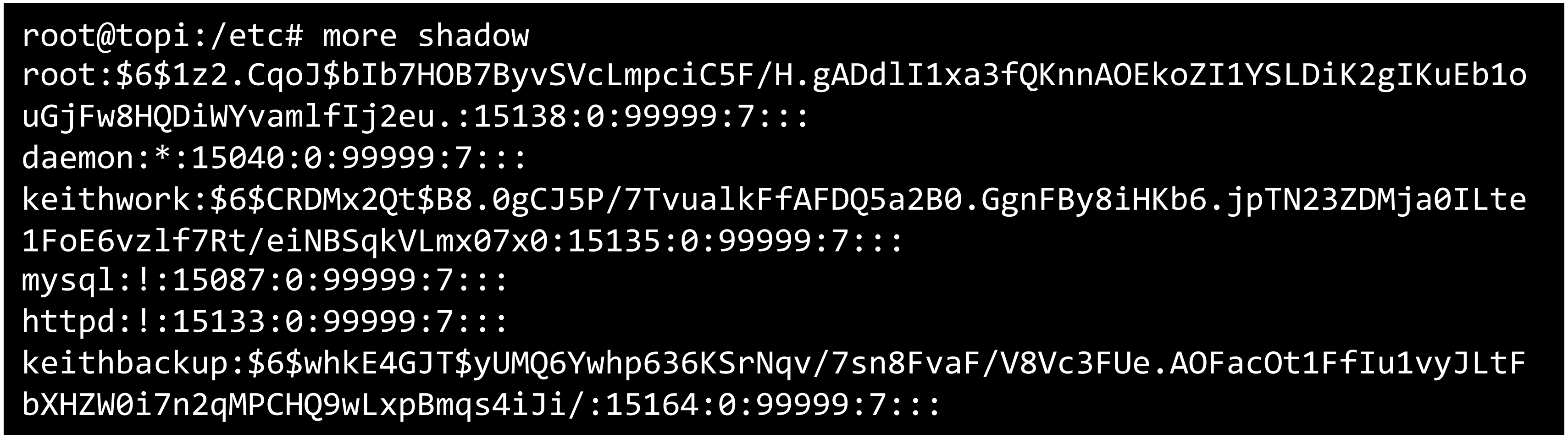
Fox → cryptographic hash function → DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17

The red fox jumps over the blue dog → cryptographic hash function → 0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC

The red fox jumps ouer the blue dog → cryptographic hash function → 8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819

The red fox jumps oevr the blue dog → cryptographic hash function → FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45

The red fox jumps oer the blue dog → cryptographic hash function → 8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C



```
root@topi:/etc# more shadow
root:$6$1z2.CqoJ$bIb7HOB7ByvSVcLmpciC5F/H.gADdlI1xa3fQKnnAOEkoZI1YSLDiK2gIKuEb1o
uGjFw8HQDiWYvamlfIj2eu.:15138:0:99999:7:::
daemon:*:15040:0:99999:7:::
keithwork:$6$CRDMx2Qt$B8.0gCJ5P/7TvualkFfAFDQ5a2B0.GgnFBy8iHKb6.jpTN23ZDMja0ILte
1FoE6vzlf7Rt/eiNBSqkVLmx07x0:15135:0:99999:7:::
mysql:!:15087:0:99999:7:::
httpd:!:15133:0:99999:7:::
keithbackup:$6$whkE4GJT$yUMQ6Ywhp636KSrNqv/7sn8FvaF/V8Vc3FUe.AOFacOt1FfIu1vyJLtF
bXHZW0i7n2qMPCHQ9wLxpBmqs4iJi/:15164:0:99999:7:::
```

Overview

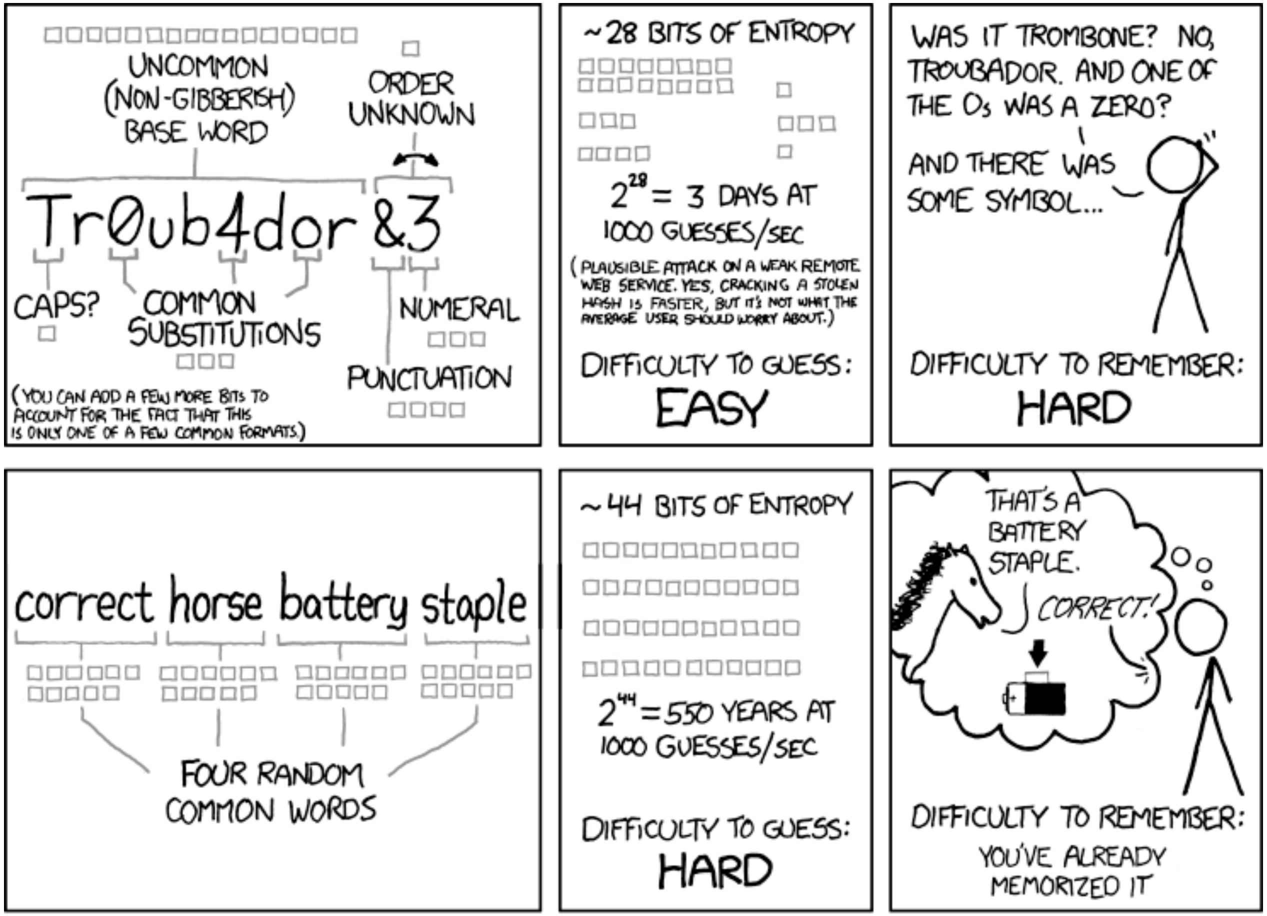
- Authentication
  - Passwords
    - One-way hashing
    - Salting passwords
  - Other forms: tokens, biometrics
  - Digital signing
    - Public key based signing
    - PKI, CA
- Pretty Good Privacy (PGP)
- Securing web commerce
  - Secure Socket Layer (SSL), Transport Layer Security (TLS)
  - https

Authentication

- Proving your identify
  - Something you know
    - e.g. password, PIN, favorite pet
  - Something you possess
    - e.g. a key, smart card
  - Something you are
    - e.g. fingerprints, retina, face
  - Something you do
    - e.g. voice pattern, handwriting, typing rhythm
- Means of authentication
  - Password
  - Token-based
  - Biometric

Password authentication

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


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

https://xkcd.com/936/

Password storage

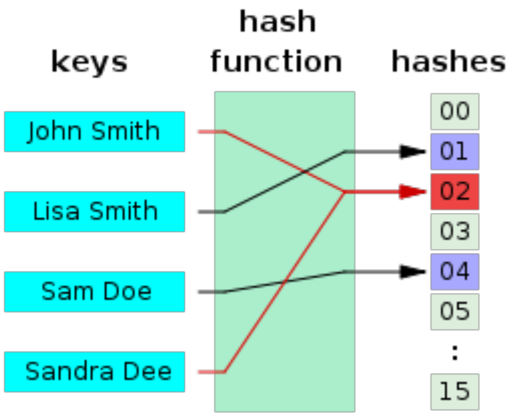
- User ID and password
  - Must be stored somewhere, e.g. /etc/passwd
  - Shadow password file, e.g. /etc/shadow
    - Reachable only by privileged users

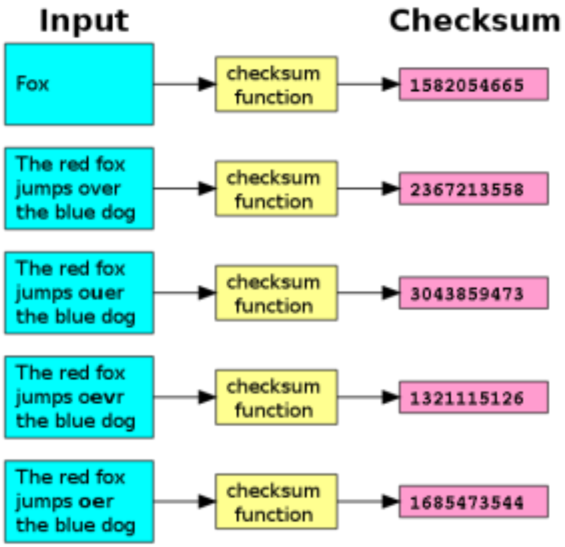
```
$ more passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
keithwork:x:1000:1000::/home/keithwork:/bin/sh
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
httpd:x:1001:1001::/home/httpd:/bin/sh
keithbackup:x:1005:1005::/home/keithbackup:/bin/sh
```

```
root@topi:/etc# more shadow
root:$6$1z2.CqoJ$bIb7HOB7ByvSVcLmpciC5F/H.gADdlI1xa3fQKnnAOEkoZI1YSLDiK2gIKuEb1o
uGjFw8HQDiWYvamlfIj2eu.:15138:0:99999:7:::
daemon:*:15040:0:99999:7:::
keithwork:$6$CRDMx2Qt$B8.0gCJ5P/7TvualkFfAFDQ5a2B0.GgnFBy8iHKb6.jpTN23ZDMja0ILte
1FoE6vzlf7Rt/eiNBSqkVLmx07x0:15135:0:99999:7:::
mysql:!:15087:0:99999:7:::
httpd:!:15133:0:99999:7:::
keithbackup:$6$whkE4GJT$yUMQ6Ywhp636KSrNqv/7sn8FvaF/V8Vc3FUe.AOFacOt1FfIu1vyJLtF
bXHZW0i7n2qMPCHQ9wLxpBmqs4iJi/:15164:0:99999:7:::
```

Hashing

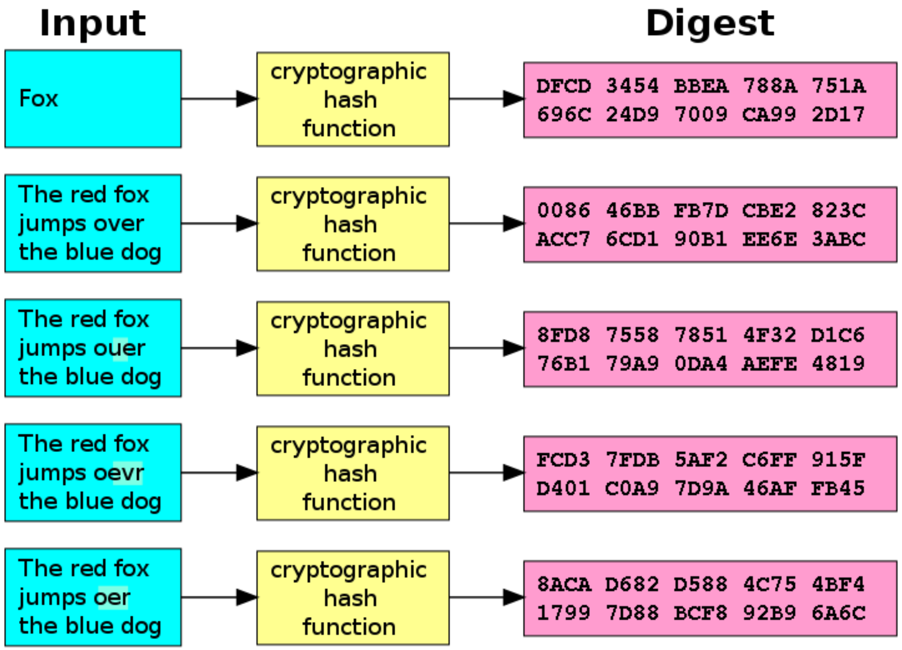
- Normal hash functions:
  - Key: large data set of variable length
  - Value: smaller data set of fixed length
  - e.g. checksum, CRC





Secure hashing

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



Input

Digest

Fox → cryptographic hash function → DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17

The red fox jumps over the blue dog → cryptographic hash function → 0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC

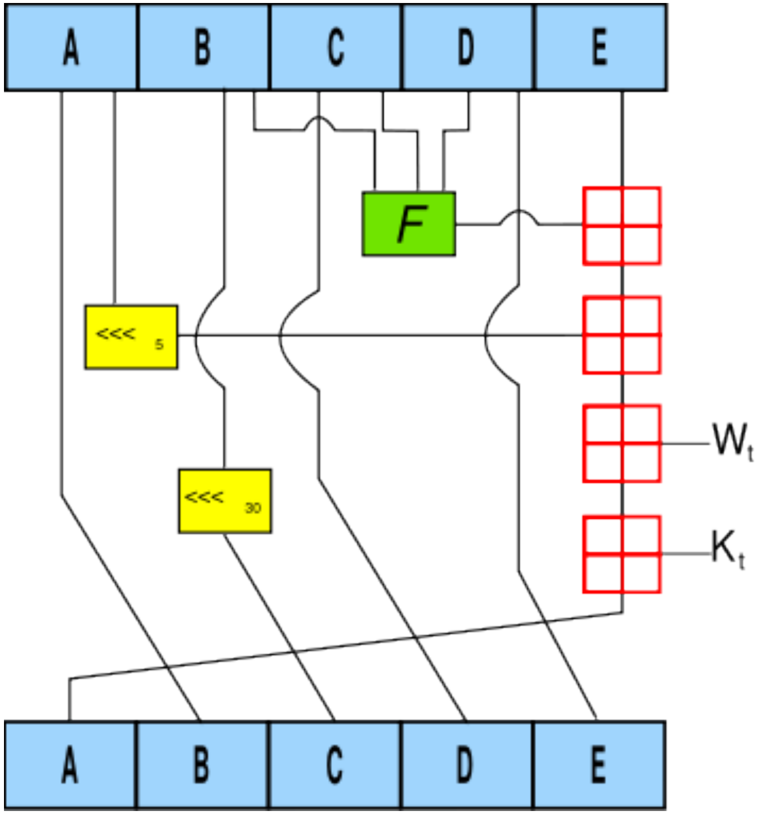
The red fox jumps ouer the blue dog → cryptographic hash function → 8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819

The red fox jumps oevr the blue dog → cryptographic hash function → FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45

The red fox jumps oer the blue dog → cryptographic hash function → 8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

Secure hashing

- Desirable properties
  - Preimage resistant, one-way
    - For any code h, intractable to find x s.t. H(x) = h
  - Second preimage 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 the hash, not the encrypted plaintext
  - Intrusion detection
    - Detect changes to a file
  - Digital signing of messages (stay tuned)



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;

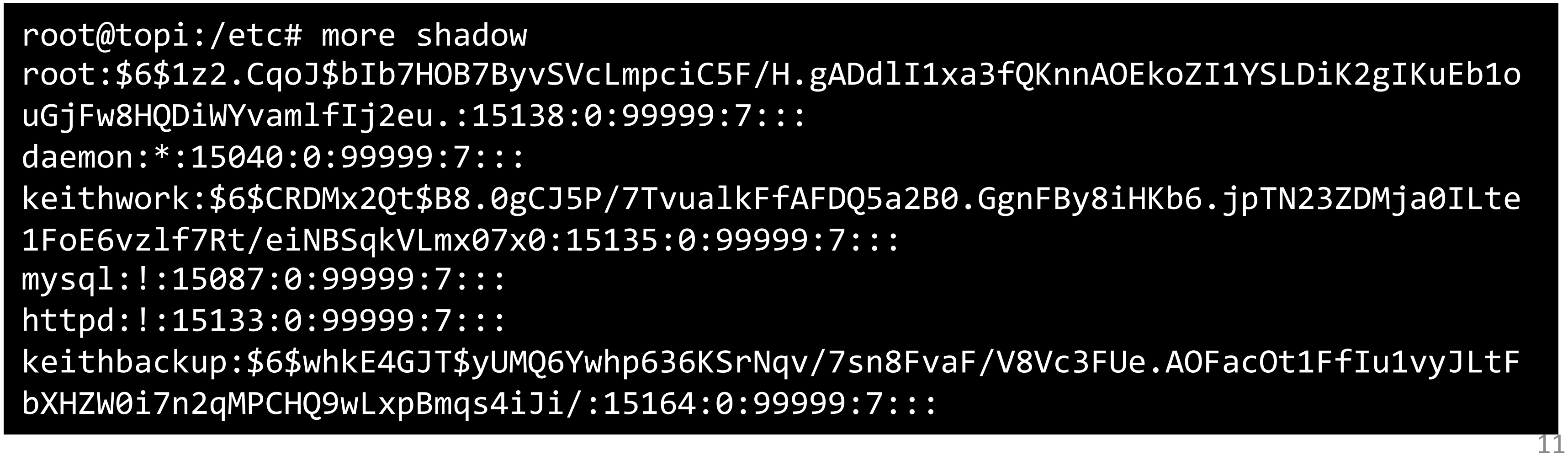
Box with plus denotes addition modulo 2^{32}.

Attacking passwords

- If hashed password file compromised
  - Attacker knows users with same password
  - Attacker can tell if user has same password on multiple systems
  - Attacker can use an offline dictionary attack
- Dictionary attack
  - Precompute hash value for
    - All sequences of a given (shortish) length
    - Common words
  - Check for match against hash value in password file

Salt

- Salting passwords
  - On account creation, system chooses fixed-length salt value
    - Timestamp
    - Random value
  - Salt value stored unencrypted associated with user ID
  - Stored hash computed from salt plus user password
  - Makes dictionary attack much more expensive



```
root@topi:/etc# more shadow
root:$6$1z2.CqoJ$bIb7HOB7ByvSVcLmpciC5F/H.gADdlI1xa3fQKnnAOEkoZI1YSLDiK2gIKuEb1o
uGjFw8HQDiWYvamlfIj2eu.:15138:0:99999:7:::
daemon:*:15040:0:99999:7:::
keithwork:$6$CRDMx2Qt$B8.0gCJ5P/7TvualkFfAFDQ5a2B0.GgnFBy8iHKb6.jpTN23ZDMja0ILte
1FoE6vzlf7Rt/eiNBSqkVLmx07x0:15135:0:99999:7:::
mysql:!:15087:0:99999:7:::
httpd:!:15133:0:99999:7:::
keithbackup:$6$whkE4GJT$yUMQ6Ywhp636KSrNqv/7sn8FvaF/V8Vc3FUe.AOFacOt1FfIu1vyJLtF
bXHZW0i7n2qMPCHQ9wLxpBmqs4iJi/:15164:0:99999:7:::
```

Improving password security

- Reactive password checking
  - System periodically attacks itself, revokes passwords it guesses
- Proactive password checker
  - Users selects a candidate password
  - System checks to see if allowable
  - Hopefully guide users to secure choice without annoying them too much

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"
- Computer-generated passwords
  - Normally low acceptance, users write them down
  - Generate pronounceable syllables, FIPS PUB 181

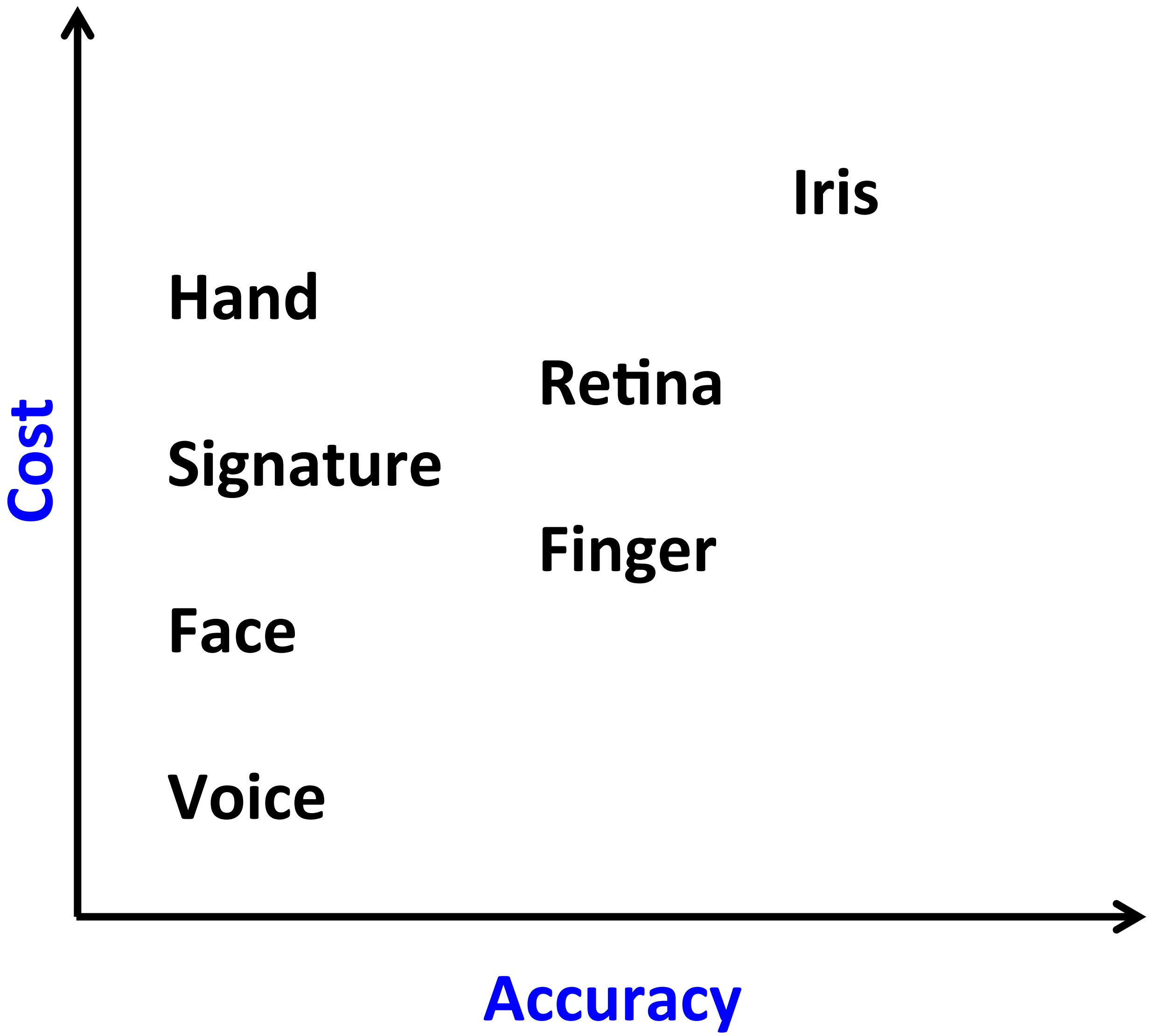
Token-based authentication

- Require users possess some object
  - Unique ID based on magnetic strip, embedded microprocessor
  - e.g. ATM card
- Often in combination with user knowledge
  - e.g. ATM PIN

Biometric authentication

- Pattern recognition based
  - Facial recognition: relative 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



Iris

Hand

Retina

Signature

Finger

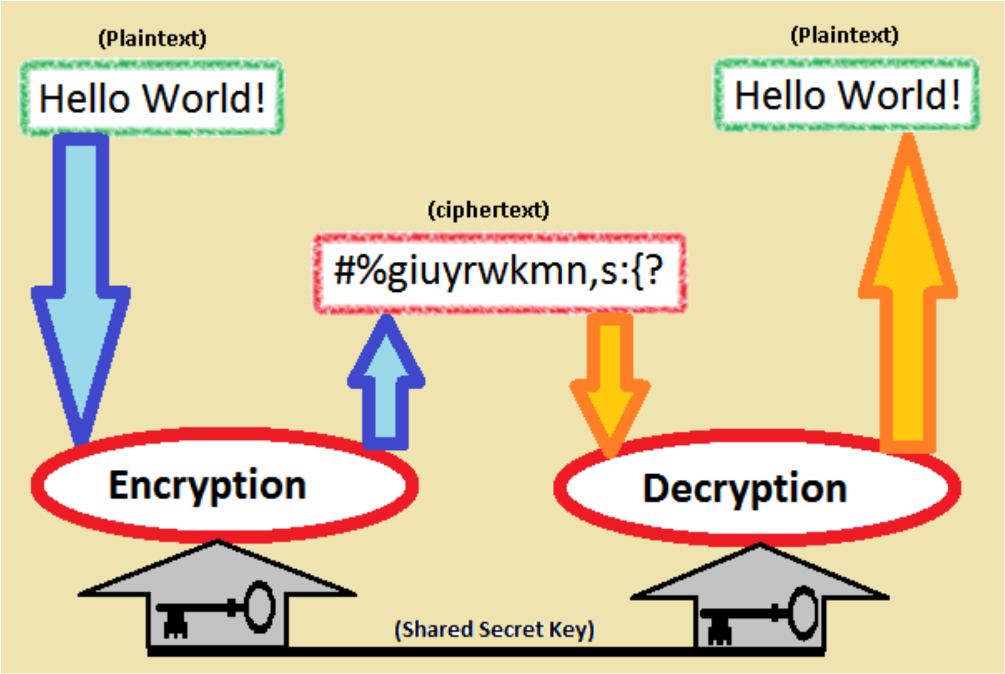
Face

Voice

Cost

Accuracy

Digital signing

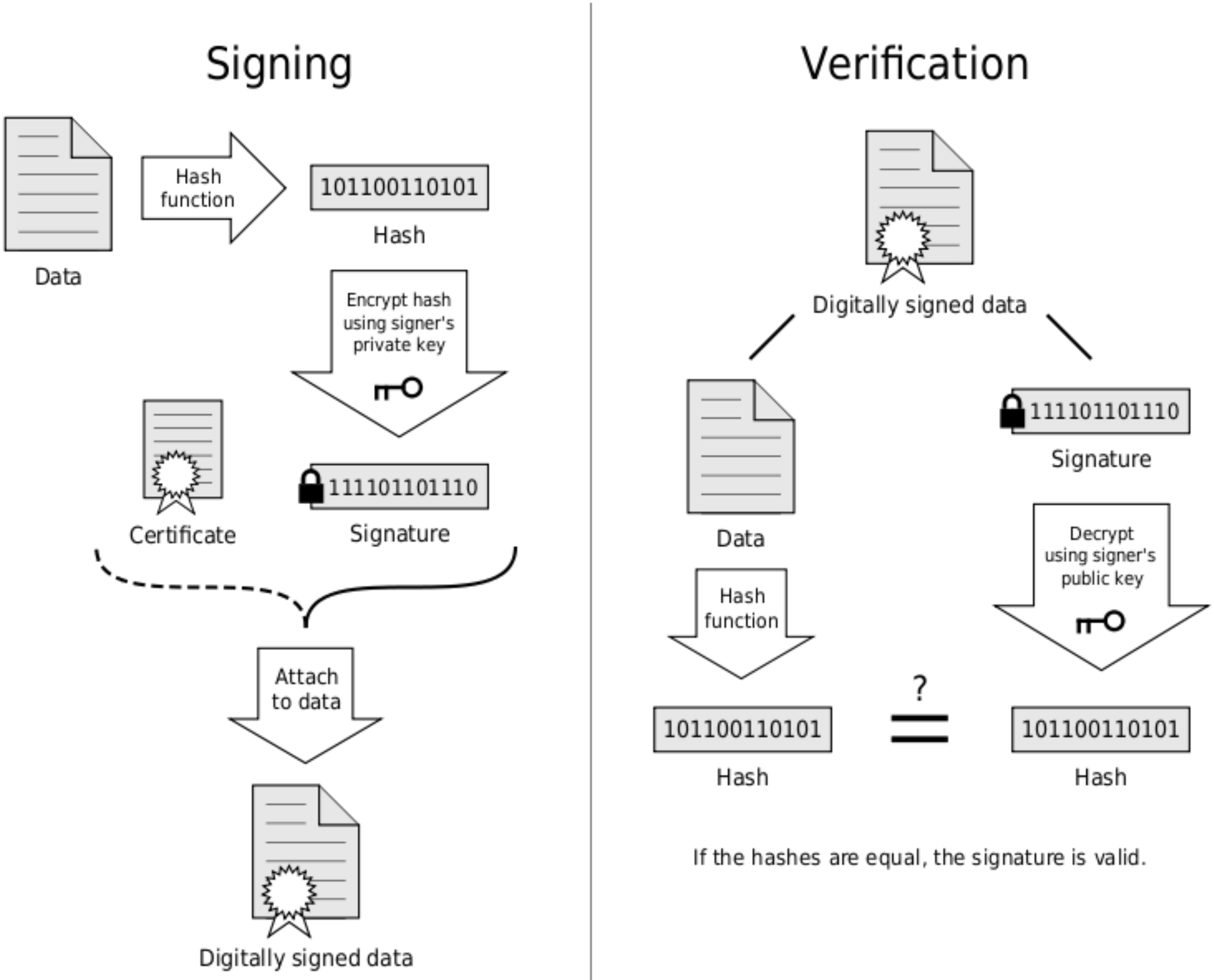


- Normal public-key encryption
  - Alice encrypts message with Bob's public key
  - Bob is the only one to decrypt using his private key
  - Message is a love letter claiming to be from Alice

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
  - Asymmetric crypto on entire message can be expense
    - Hash the message
    - Encrypt just the hash

Hash based digital signing



Signing

Data

Hash function

101100110101

Hash

Encrypt hash using signer's private key

Certificate

111101101110

Signature

Attach to data

Digitally signed data

Verification

Digitally signed data

Data

111101101110

Signature

Hash function

Decrypt using signer's public key

101100110101

Hash

?
=

101100110101

Hash

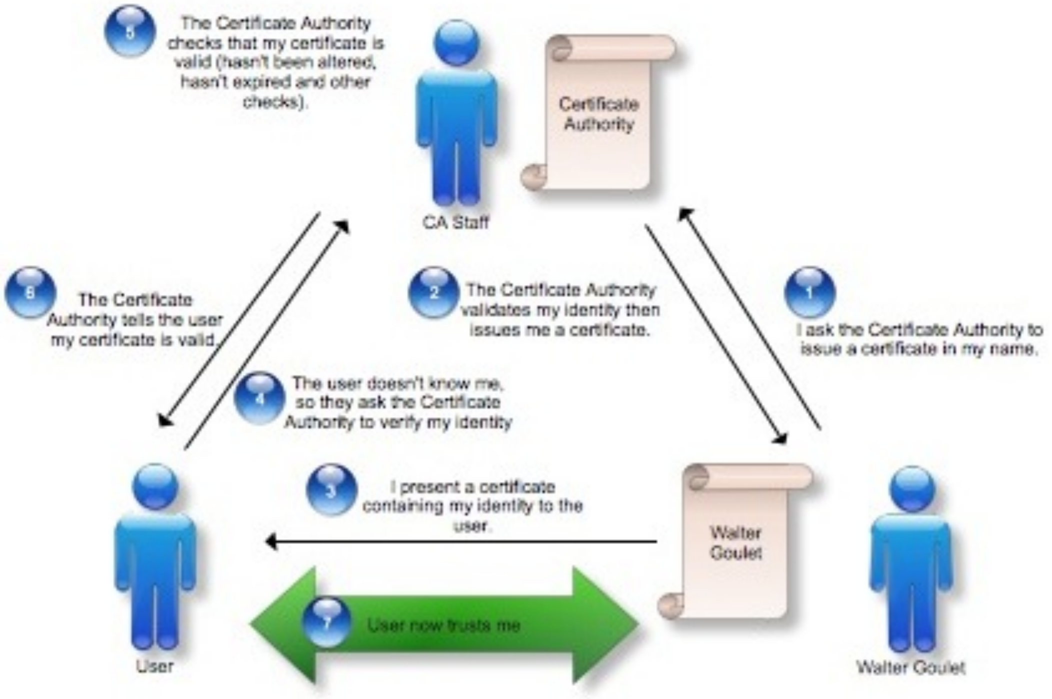
If the hashes are equal, the signature is valid.

Distributing public keys

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



5 The Certificate Authority checks that my certificate is valid (hasn't been altered, hasn't expired and other checks).

6 The Certificate Authority tells the user my certificate is valid.

2 The Certificate Authority validates my identity then issues me a certificate.

1 I ask the Certificate Authority to issue a certificate in my name.

4 The user doesn't know me, so they ask the Certificate Authority to verify my identity

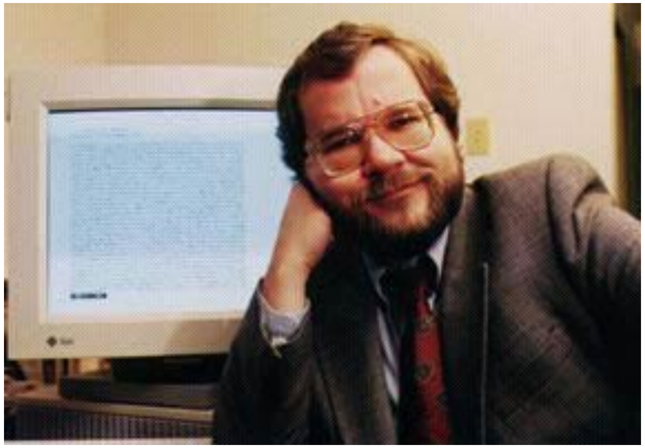
3 I present a certificate containing my identity to the user.

7 User now trusts me

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

PGP

- Pretty Good Privacy (PGP)
  - 1991 Phil Zimmermann



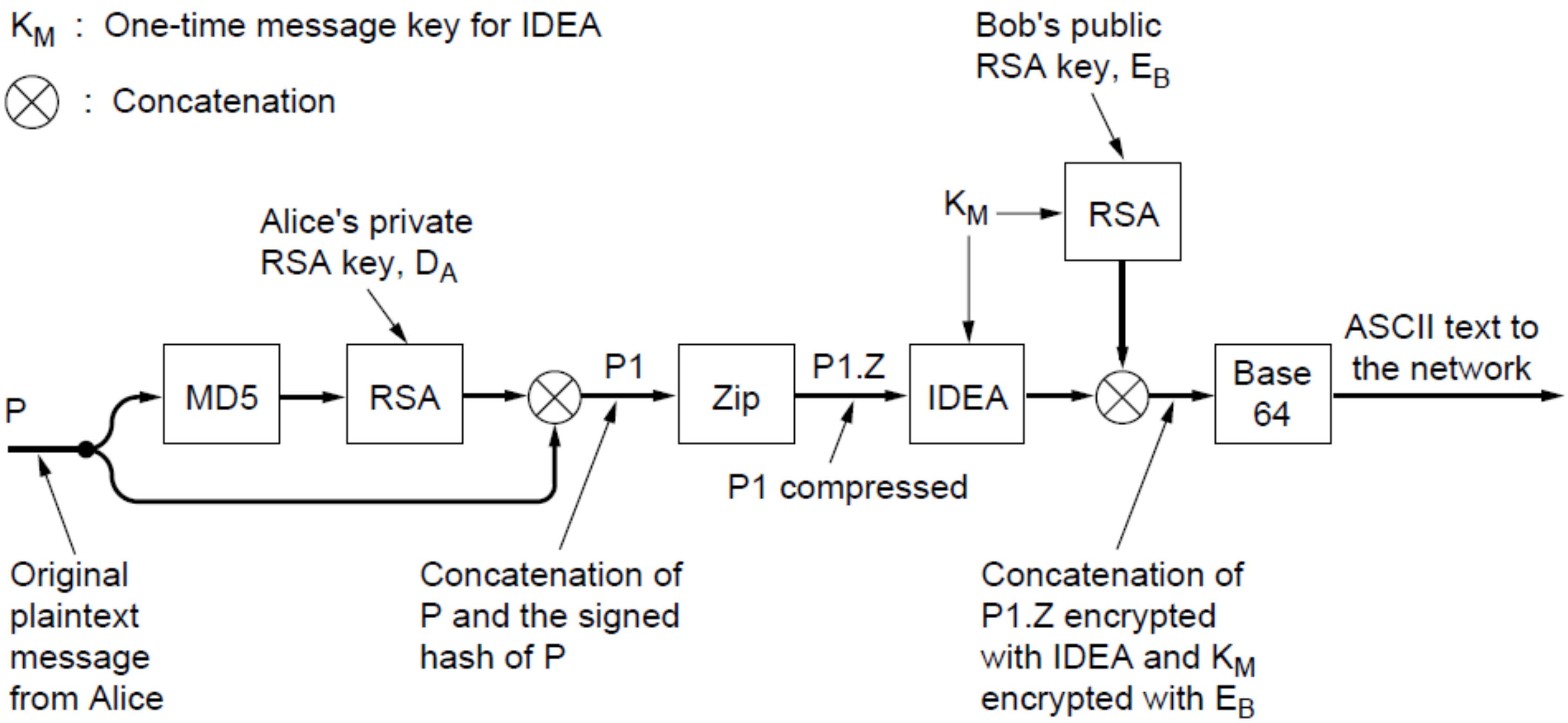
"In the past, if the Government wanted to violate the privacy of ordinary citizens, it had to expend a certain amount of effort to intercept and steam open and read paper mail, and listen to and possibly transcribe spoken telephone conversation. This is analogous to catching fish with a hook and a line, 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 is 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
    - Key idea:
      - RSA for key exchange
      - Faster symmetric cipher (IDEA) for bulk of data 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


K_M : One-time message key for IDEA
⊗ : Concatenation
Bob's public RSA key, E_B
Alice's private RSA key, D_A
K_M
RSA
MD5
RSA
P1
Zip
P1.Z
IDEA
Base 64
ASCII text to the network
P
P1 compressed
Original plaintext message from Alice
Concatenation of P and the signed hash of P
Concatenation of P1.Z encrypted with IDEA and K_M encrypted with E_B

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

Securing web commerce

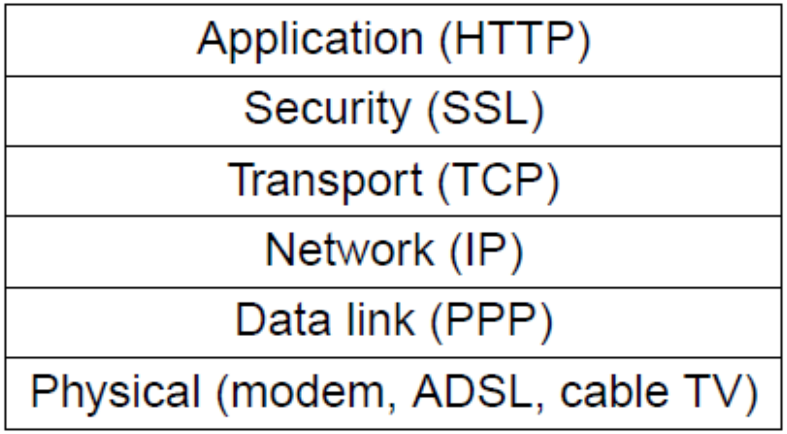
- Customer filling out order form with credit card #
  - Problem 1: Keep data secure from customer's browser to the web server
  - Problem 2: keep data secure on server or in transit to order fulfillment

SSL

- Secure Sockets Layer (SSL) / Transport Layer Security (TLS)
  - Client requests secure connection from server
  - Client sends list of ciphers and hash function 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 public 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