Encryption and Digital Signatures

A Yie

Motivations

We want to safeguard data!

- Authentication - ssh
- Privacy - email
- Digital Signatures - safe electronic transactions

Assumptions

- Expect attacks
- Underlying insecure network
- Goal of reliable, private communication

The Popek paper also assumes that the machines being used support some fast method of encryption, possibly in hardware, and that the system must support many logical channels.

Attacks

- Sniffing / tapping
- False messages
- Extraneous copies of valid messages
- Network disruption

Encryption

- Conventional / symmetric key

  E = F(D, K) and D = F'(E, K)

- Public key

  E = F(D, K) and D = F'(E, K')s

- Predicated on difficulty of a hard math problem, such as computing the inverse of a function

Limitations

- Processing in cleartext

- Overhead:

  computation

  storage

  protection

  revocation

  initial distribution

Placement

- Node or network hardware
- Operating system
- Application level

In general, higher level integration yields better security but reduces convenience.

Management

- Goal: minimum number of trusted participants
- Distributed management
- Centralized management
- Similar schemes for public-key and conventional encryption

Digital Signatures I

- Unforgeable
- Easy to verify authenticity
- Difficult to disavow
- Convenient

Digital Signatures II

- Network registry: inject a layer between the user and his or her keys

- Notary: send messages to a third party who also tags the message

Hash functions

- One-way
- Second-preimage resistant
- Collision resistant

Building an encrypted message

- Cipher: stream or block

- Compression

  Use the Merkle-Damgård construction!

  Allows us to handle longer messages, but introduces other structure.

Attacks

- Collisions
- Near collisions
- Pseudo-collisions (compression)
- Free-start collision (compression)

RSA Idea

Take two large primes, p and q. Let $n = pq$.

Choose $e < n$, e coprime to $(p-1)(q-1)$.

Take d such that $(p-1)(q-1)$ divides $ed-1$.

Resulting public key: (n, e)

Resulting private key: (n, d)

RSA Encryption

Given secret key (n, d), public key (n, e), and message M, we produce ciphertext C:

Encryption: $C = M^e \mod n$

Decryption: $M = C^d \mod n$

Since only the recipient knows d, only the recipient can decrypt the message.

RSA: Signatures

Given secret key (n, d), public key (n, e), and message M, we produce signature S:

Creation: $S = M^d \mod n$

Verification: $M = S^e \mod n$

Through the entire process, each participant uses only his or her private key and the public key of the other participant.

DES Intro

- Symmetric key algorithm
- Block cipher: takes 64 bits of plaintext and uses a 56 bit key to generate 64 bits of ciphertext

DES Feistel

DES consists of 16 rounds on each block.

Before each round, 64 bit block broken into two 32 bit blocks. Alternate the order that algorithm processes these 32-bit blocks.

This allows encryption and decryption to be performed very quickly and with the same keys.

DES: Steps in a round

- Expand a 32 bit block into a 48 bit block.
- xor the new 48 bit block with a 48 bit subkey from the 56 bit key.
- Break the 48 bit block into eight 6-bit pieces. Use a nonlinear function to generate 4 output bits from each of these 6-bit chunks.
- Permute the resulting 32 bits. This becomes the input for the next round.

Conclusion

You've got to trust someone at some point.
The major tradeoff in the security literature:

Security vs. practicability & ease of use