Class Note: April 10, 2007

Public Key cryptography

Each person has a public key, which can be given to anyone, and a secret key which is kept secret.

Example/

Alice has P_A (public key) and S_A (private key)

To encrypt a message to Alice:

$C = E(P_A, M)$, where C is cipher text, and M is message

To read message:

$M = D(S_A, C)$

RSA:

To generate Public/Private keys, Alice does:

1. Pick primes p, q
2. $N = p * q$, $\Theta(N) = (p-1)(q-1)$
3. Find e and d such that $e*d=1 \bmod \Theta(N)$
4. $P_A = (e, N)$, $S_A = (d, N)$

RSA encryption & decryption

$E((e, N), M) = m^e \bmod N$

$D((d, N), C) = c^d \bmod N = m^{ed} \bmod N = M$ (Fermat's Little Theorem)

Example/

Let p = 7, q = 5 then N = 35, $\Theta(N) = 24$

Let e = 7, d = 7 (In real system, e & d will never be the same)

$P_A = (7, 35)$, $S_A = (7, 35)$

$E(P_A, 2) = 23$

$D(S_A, 23) = 2$

Public Key Signatures:

Each user has a private signing key and a public verification key, S_A and P_A, respectively.

$S = sig(S_A, M)$

$Vrfy(P_A, M, S) = valid/invalid$

Sig((d, N), M) = m^d mod N

Vrfy((e, N), M, S) = valid iff s^e = M mod N

Signing Long Messages

Use hash function

S = sig(SA, h(M))

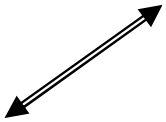
h: {0, 1}* → $\{0, 1\}^n$, where n is number of bits

A hash function is strong collision resistance if it is hard to find x =/= y such that h(x) = h(y)

Certificates

To verify Alice's signature, Bob needs to know P_A.

Suppose Alice & Bob trust Charlie



Charlie

S_C

Alice

P_A, S_A, P_C

Bob

P_C

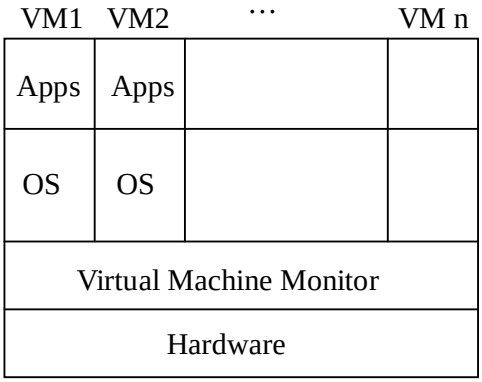
Obtains

C_A = sig(S_C, "Alice's public is P_A")

Terra, TVMM

- The owner of the computer is malicious
- Need "root_secure" system
- Applications
  - Network games
  - Movie/music players (aka DRM)
  - Reverse engineering

Trusted Virtual Machine Monitor



-VMM exposes hardware interface to guest VMs

- *Strong isolations* – each VM is completely isolated, such as memory, CPU, disk space, networks, etc…

Remote Attestation

Goal: Prove to remote party that I am running a certain set of software: Apps, OS, VMM, Boot loader, BIOS.