

# Outline

- ▶ Chapter 18: Protection
- ▶ Chapter 19: Security
- ▶ A Method for Obtaining Digital Signatures and Public-Key Cryptosystems Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman.  
Communications of the ACM 21,2 (Feb. 1978)
  - RSA Algorithm – First practical public key crypto system
- ▶ Authentication in Distributed Systems: Theory and Practice, Butler Lampson, Martin Abadi, Michael Burrows, Edward Wobber



# Protection

- ▶ Protect computer resources from being accessed by processes that should not have access
  - Access right: Operations allowed on an object
  - Domain: Set of all access rights
- ▶ UNIX: domain is userid, setuid bit in file switches domains
- ▶ Multics: rings, tasks can get access based on entry points
- ▶ Access Matrix defines protection: rows represent domains & columns represent objects
  - Global table
  - Access list for objects: easier to program
  - Capability list for domains/users:
  - Hybrid: lock-key mechanism
- ▶ Revocation of rights:
  - Immediate vs delayed, selective vs general, partial vs total, temporary vs permanent



# Revocation

- ▶ *Access List* – Delete access rights from access list.
  - Simple
  - Immediate
- ▶ *Capability List* – Scheme required to locate capability in the system before capability can be revoked.
  - Reacquisition
  - Back-pointers
  - Indirection
  - Keys



# Compiler/language based mechanism

- ▶ Compiler based enforcement
  - Specification of protection in a programming language allows the high-level description of policies for the allocation and use of resources
- ▶ Java VM
  - Multiple threads within a single JVM have different access rights
- ▶ A class is assigned a protection domain when it is loaded by the JVM. The protection domain indicates what operations the class can (and cannot) perform
  - Protection enforced using stack inspection



# Security

- ▶ Security problem: protection from unauthorized access, malicious modification or destruction
- ▶ User authentication:
  - Passwords
    - Encrypted passwords
      - Encrypted form should be secret because attacker can check offline
    - One-time passwords
    - Biometrics
- ▶ Threats:
  - Trojan horse
  - Trap door/stack and buffer overflow
  - Worms/viruses
  - Denial of service
  - Intrusion and detection



# Risk analysis

- ▶ Important to understand threat and perform risk analysis
  - No system is “secure”, systems usually trade security for performance, ease of use etc.
  - If information is worth  $x$  and it costs  $y$  to break into system and if  $(x < y)$ , then not worth encryption
  - Wasteful to build a system that is more secure than is necessary
    - Ssh in CSE dept – good
  - Palm pilots may not require powerful encryption systems if they are expected to be physically secure



# Security classification

- ▶ U.S. Department of Defense outlines four divisions of computer security: A, B, C, and D
  - D – Minimal security
  - C – Provides discretionary protection through auditing. Divided into C1 and C2. C1 identifies cooperating users with the same level of protection. C2 allows user-level access control
  - B – All the properties of C, however each object may have unique sensitivity labels. Divided into B1, B2, and B3
  - A – Uses formal design and verification techniques to ensure security
- ▶ Windows NT: Configurable security from D to C2
- ▶ SuSE Linux Enterprise Server 8 on IBM eServer xSeries - Evaluation Assurance Level 2+ certification (EAL2)
- ▶ <http://www.radium.ncsc.mil/tpep/epl/historical.html>



# Security Attacks

## ▶ Social engineering attacks

- Preys on people gullibility (good nature), hardest to defend
  - E.g. I once got an unlisted number from a telephone operator because I sounded desperate (I was, but that was not the point)
  - E.g. Anna kour\*va virus, Nigerian email scam, MS update scam
  - E.g. If I walk in with coupla heavy looking boxes into the elevator to go to Fitz 3<sup>rd</sup> floor (at night) would you let me in? You can get into “secure” companies by looking like you “belong” there

## ▶ Denial of service attacks

- Network flooding, Distributed DOS, holding resources, viruses





# Common technology - firewalls

- ▶ Firewalls are used to restrict the kinds of network traffic in/out of companies
  - Application level proxies
  - Packet level firewalls
- ▶ Does not prevent end-to-end security violations
  - People sometimes email list of internal computer users outside firewall to scrupulous “researchers”
  - Emails viruses exploit certain vulnerabilities in VBS to get around firewalls



# Intrusion detection

- ▶ Detect attempts to intrude into computer systems.
- ▶ Detection methods:
  - Auditing and logging
  - Tripwire (UNIX software that checks if certain files and directories have been altered – I.e. password files)
- ▶ System call monitoring



# RSA Paper - Encryption

- ▶ Properties of good encryption technique:
  - Relatively simple for authorized users to encrypt and decrypt data.
  - Encryption scheme depends not on the secrecy of the algorithm but on a parameter of the algorithm called the encryption key.
  - Extremely difficult for an intruder to determine the encryption key.



# Strength

- ▶ Strength of crypto system depends on the strengths of the keys
- ▶ Computers get faster – keys have to become harder to keep up
- ▶ If it takes more effort to break a code than is worth, it is okay
  - Transferring money from my bank to my credit card and Citibank transferring billions of dollars with another bank should not have the same key strength



# Encryption methods

## ▶ Symmetric cryptography

- Sender and receiver know the secret key (apriori )
  - Fast encryption, but key exchange should happen outside the system

## ▶ Asymmetric cryptography

- Each person maintains two keys, public and private
  - $M \equiv \text{PrivateKey}(\text{PublicKey}(M))$
  - $M \equiv \text{PublicKey}(\text{PrivateKey}(M))$
- Public part is available to anyone, private part is only known to the sender
- E.g. Pretty Good Privacy (PGP), RSA



# My Public Key

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: PGPfreeware 7.0.3 for non-commercial use <<http://www.pgp.com>>

```
mQGiBDqtLPwRBADnG0+9IkDvI8t/3wdL3CSO4DyteEH0NjrNwAYYIaewp3MklsxkP
p6iVblwiiCH4T4Nqkaru+kaEQ1hSTa7E/F9yQCWN5J0u1U7mtgTKFyt7VG0txAVx
tV7TuyxNogJkpm2BqoKqqUdCdbm+GurX/G2ynbINjEOvhcy0i1ttxgyDrwCg/8HZ
tM0i06VVNcR/QCmA+JdHGwMEAIjXLVV97huEtpuWDiq4J53ecV3HXQm6XoUZq4Sc
n+nsvXe4UD+61dub/riOqBy22fBBAKhUsM3lGFgr7h19X3RGdw/yBVox+BLajpW+
F+ddjJAVSFeTvNanhnXL9a3nwcThb4aEUTdD61kgoUWJl2BnsK1DUSo2X6AsZY+
GknOA/92dUNYUzspPLkXvPjOo+uJErZA4an+UYsJwD3AlYugVLkc3nQBQySO4bAR
XitjnN0DA6Kz/j6e+cqReCyEuBnPtay/Nn/dAnllgUlJ/EtKQ9J4krI3+RxRmlpY
UtWyTaakV/QCXkB/yB9i6iAfsCprlcrSpmZAGuNXr+pHTHB0ILQmU3VyZW5kYXIG
Q2hhbmRyYSA8c3VyZW5kYXJAY3MudWdhLmVkdT6JAFgEEBECABgFAjqtLPwICwMJ
CAcCAQoCGQEFgwMAAAACgkQ1U7dFVWfeisqTACfXxU9a1mbouW2nbWdx6MHatQ6
TOgAoM9W1PBRW8Iz3BIgcnSsZ2UPNJHduQINBDqtLPwQCAD2Qle3CH8IF3Kiutap
QvMF6PlTETlPtvFuuUs4INoBp1ajFomPQFXz0AfGy0OplK33TGSgsfgMg7116RfU
odNQ+PVZX9x2Uk89PY3bzpnhV5JZzf24rnRPxfx2vIPFRzBhzJZv8V+bv9kV7H
AarTW56NoKVyOtQa8L9GAFgr5fSI/VhOSdvNILSd5JEHNmszbDgNRR0PfiizHHxb
LY7288kjwEPwpVsYjY67VYy4XTjTNP18F1dDox0Ybn4zISy1Kv884bEpQBgrjXyE
pwpY1obEAxnIByl6ypUM2Zafq9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6q6JewlXp
Mgs7AAIcCACLxNC3Vth553Y90JCVyM9mPWzvrkjEGfBiCFDZ0HONW81ywUyV6jT
O/1sUsgr7jGB26XBsnIY96a9WTpUoI+20YstFLRj8sXOVXuaP/YTmgSLv8206SWd
Bze1S0YJcU31/zdCftsz67UWT8vg39yeGyQ5KQP83p9DKpi4Z5K4M29p8eCt9BY+
kid94h9+16ZT8JLF0iEwGapZvpaTucCNoC8t6CKPto0dGpkYp7uBYoSzLgNvUh2n
BjGVEmluiobqbOaomDERITY2iNcW3CCgjjYvvg/Hnu7HB2xKzuVUN1NTGogcuNI
Yx88mi+d/HxTY6YNr9xNW0f0pWkZDVB0iQBMBBgRagAMBQI6rSsz8BRsMAAAAAAoJ
EJVO3RVVn3orYhIAoIQPxGvHmX8c6kaAZqkolzYCeixcAJ9tp5h/KQZrIN/BpyTW
9Xgv4qxKEA==
=Pv50
```

-----END PGP PUBLIC KEY BLOCK-----



# Public Key Infrastructure (PKI)

- ▶ Process of issuing, delivering, managing and revoking public keys
- ▶ E.g. Secure Socket Layer (SSL)
  - Client C connects to Server S
    1. C requests server certificate from S
    2. S sends server certificate with  $S_{public}$  to C
    3. C verifies validity of  $S_{public}$
    4. C generate symmetric key for session
    5. C encrypts  $C_{symmetric}$  using  $S_{public}$
    6. C transmits  $C_{symmetric}(data)$  and  $S_{public}(C_{symmetric})$  to S



# Authentication

- ▶ Identification verification process
  - E.g. kerberos certificates, digital certificates, smart cards
- ▶ Used to grant resources to authorized users





# RSA

- ▶ Named after Rivest, Shamir and Adleman
  - Only receiver receives message:
    - Encode message using receivers public key
  - Only sender could've sent the message
    - Encode message using sender's private key
  - Only sender could've sent the message and only receiver can read the message
    - Encode message using receivers public key and then encode using our private key



# Practical Public Key Cryptosystem

1.  $\text{Decrypt}(\text{Encrypt}(\text{Message})) = \text{Message}$
  2.  $\text{Encrypt}()$  and  $\text{Decrypt}()$  are easy to compute
  3.  $\text{Encrypt}()$  does not reveal  $\text{Decrypt}()$
  4.  $\text{Encrypt}(\text{Decrypt}(\text{Message})) = \text{Message}$
- ▶ Function satisfying 1-3: Trap-door one-way function
- One way: easy to compute in one direction, difficult in the other direction
  - Trap-door: Inverse functions are easy to compute once certain private “trap-door” information is known.



# Signature

- ▶ Encrypt using private key of sender. Anyone can decrypt using the public key of sender to verify signature

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Hello world!!

-----BEGIN PGP SIGNATURE-----

Version: PGPfreeware 7.0.3 for non-commercial use  
<<http://www.pgp.com>>

iQA/AwUBOq8LO5VO3RVVn3orEQLFZwCdGi9AWvIhollaYmr9TP  
vtdbKoe20AoLLr  
vbJ8SgklZ73lCy6SXDl91osd  
=L3Sh  
-----END PGP SIGNATURE-----



# Privacy

- ▶ Encrypt with receivers public key

-----BEGIN PGP MESSAGE-----

Version: PGPfreeware 7.0.3 for non-commercial use <<http://www.pgp.com>>

```
qANQR1DBwU4D30m79rqmjHMQB/4q1mu3IP8AsMBYSUW6udXZnF0/LVL51eYzVnAW
lxbxhHmBoZf9YEItoXw82gkgVebz+3Xfj6T5mLNy5FA6cgKKw57AY9BI3aEKIJK
/nV5qR8E/VZOhaPoog8dtV1Hpi5Z0vNCI7s5lbp3C2tlrgYtyYfe86bqCNe3yAl
btTUT+bA9HL3pXqhOoWIRB+N58T9ybn/9FyonYYfGuPdMTj+ZciK37R+ezWg5YmZ
jdDMf/CxgIIMF/Tv2jQ8KgmRkIyi6gWQmEtWzFUIAPgdpOC7TQC3sQqVjK4GyOY6
WnrXiWqO3895ukBGyHzqyssUTJFe5qncIkrmCvA3tph+uc7pCACKrYaGLSWWoQSB
L6zch2GnhG4+JpDCVKF/poJ1URkB2Odd9/OCReR0sFXZFW14IJQznu3HOhtA+y
g7Nn736fqMD9jpBZfUtKv/v4JMyWcRdp3R3icm03zi24n+244r1DQj+cVIFYPfd
zRAGTLORVjXH2amGqilKyxqMU7ZYXIMI43bFlviu4tabKYnZJxpM8keUKA3u+vPs
X9ksSoBSiT6Kow3Lac2t3Qo5TimYIS5ODFnC6Pp9aRZzNcBOKmiYO4IldFH2jta
RbcmesEjH5RpbDV4BfcOMdm2UGWZe6kAaKkSdxHIUVZAJnesbT+IQf4AZjXkmsOM
8qnBKi5xyS/wrhS4zamV/Mp+5qIGNASXUHPsp3rukovaZANdZ/Y6zNQQVim0kphd
5ECybmVrHQ==
=S9ph
-----END PGP MESSAGE-----
```



# Algorithm

- ▶ To break their algorithm requires that you factor a large prime
  - Computationally very hard. Can't be “proven” yet
  - With present technology, 512 bit key takes a few months to factor using “super computers”, 1024 takes a long time and 2048 takes a very long time
  - Takes 2 seconds to generate a 2048 bit key on a 933 Mhz Pentium, 1 seconds in a 2.4 GHz Xeon
  - Algorithm has remained secure for the past ~20 years
  - One of the most successful public key system



# Authentication

- ▶ Method for obtaining the source of the request
  - Who said this?
- ▶ Interpreting the access rule – authorization
  - Who is trusted to access this?
  - Access control list (ACL)
- ▶ Easier in central servers because the server knows all the sources

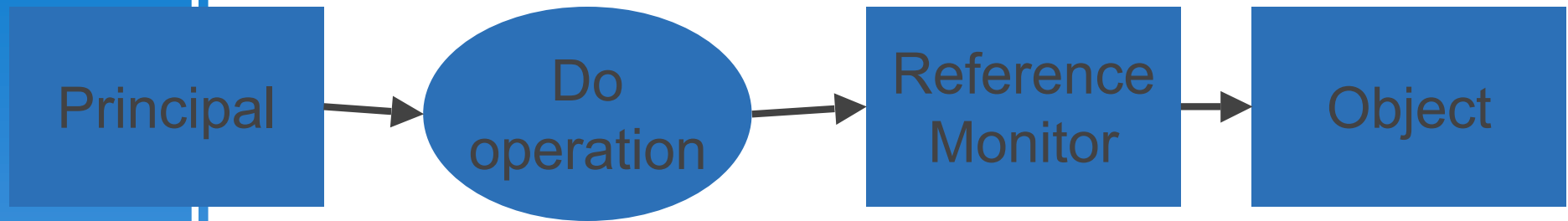


# Distributed authentication

- ▶ **Autonomy:** Request might come through a number of untrusted nodes
- ▶ **Size:** Multiple authentication sources
- ▶ **Heterogeneity:** Different methods of connecting
- ▶ **Fault-tolerance:** Parts of the system may be broken



# Access Control Model



- ▶ Principal: source for requests
- ▶ Requests to perform operations on objects
- ▶ Reference monitor: a guard for each object that examines each request for the object and decides whether to grant it
- ▶ Objects: Resource such as files, processes ..



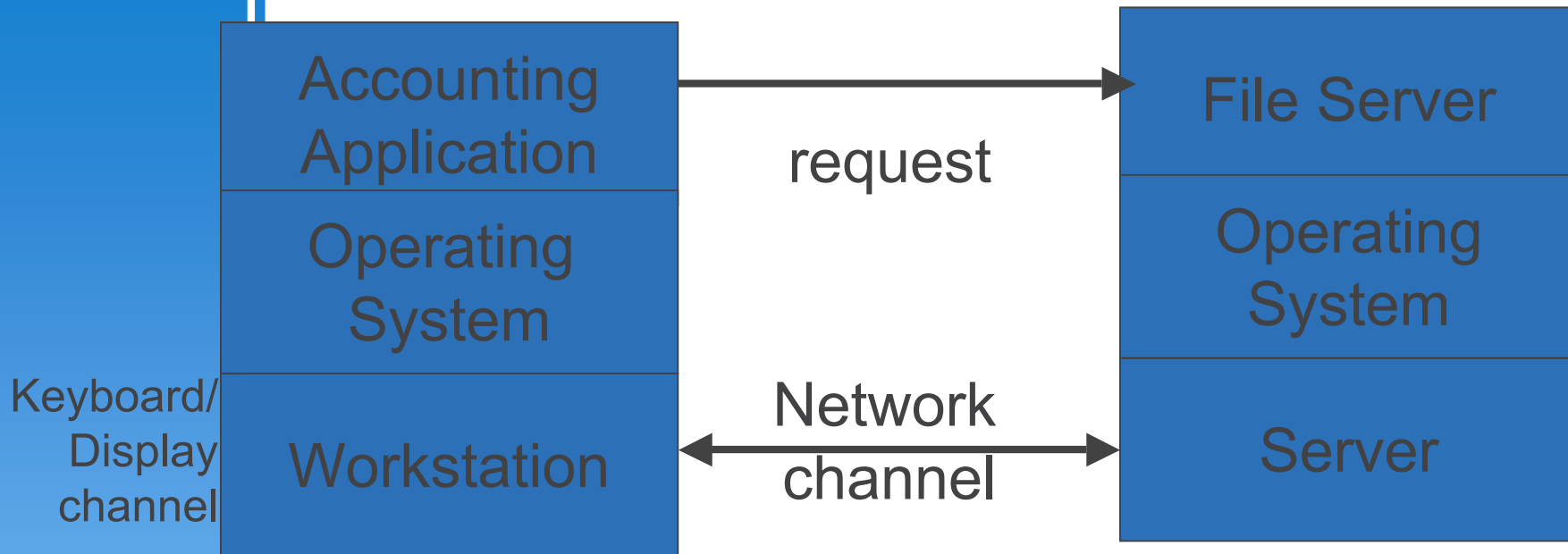


# Trusted Computing Base

- ▶ A small amount of software and hardware that security depends upon
  - You have to trust something
- ▶ Possible to obtain trusted statements from untrusted source
  - end-to-end argument
- ▶ TCB typically includes:
  - Operating system
  - Hardware
  - Encryption mechanisms
  - Algorithms for authentication and authorization



# Example scenario



- ▶ One user, two machines, two operating systems, two subsystems, and two channels
- ▶ All communication over channels (no direct comm.)



# Encryption channels

- ▶ Shared vs public key cryptography
  - Shared is fast
  - Public key systems are easy to manage
  - Hybrids offer best of both worlds (e.g. SSL)
- ▶ Broadcast encryption channels
  - Public key channel is broadcast channel: you can send a message without knowing who will receive it
  - Shows how you can implement broadcast channel using shared keys
- ▶ Node-to-node secure channels



# Principals with names

- ▶ When requests arrive on a channel it is granted only if the channel speaks for one of the principals on the ACL
  - Push: sender collects A's credentials and presents them when needed
  - Pull: receiver looks up A in some database to get credentials for A



# Man in the middle attack

Principal (C)



Resource (S)

1. C requests server certificate from S
2. S sends server certificate with  $S_{public}$  to C
3. C verifies validity of  $S_{public}$
4. C generate symmetric key for session
5. C encrypts  $C_{symmetric}$  using  $S_{public}$
6. C transmits  $C_{symmetric}(data)$  and  $S_{public}(C_{symmetric})$  to S



# Man in the middle attack

Principal (C)

Intruder  
S1/C2

Resource (S)

- ▶ C requests server certificate from S
- ▶ S sends server certificate with  $S_{public}$  to C
- ▶ C verifies validity of  $S_{public}$
- ▶ C generate symmetric key for session
- ▶ C encrypts  $C_{symmetric}$  using  $S_{public}$
- ▶ C transmits  $C_{symmetric}(data)$  and  $S_{public}(C_{symmetric})$  to S
- ▶ Certification authorities



# Certification Authority

- ▶ Difficult to make system highly available and highly secure
  - Leave CA offline, endorse certificates with long timeout
  - Online agent highly available, countersign with shorter timeout
  - Cache while both timeouts fresh
  - Invalidation at cache granularity
- ▶ Simple Certification Authority
  - CA speaks for A and is trusted when it says that C speaks for A
    - Everyone trusts CA to speak for named principal
    - Everyone knows public key of CA
- ▶ Pathnames and Multiple authorities
  - Decentralized authority, parents cannot unconditionally speak for children



# Groups

- ▶ Each principal speaks for the group
- ▶ Group membership certificates
  - Impossible to tell the membership
- ▶ Alternate approach is to distribute certificates to all principals
  - Revocation?





# Roles and programs

- ▶ Role that a user play; a normal user or sysadmin?
- ▶ ACL may distinguish the role
  
- ▶ Delegation:
  - Users delegate to compute server



# Auditing

- ▶ Formal proof for every access control decision

