Security (Cryptography) Basics

Sang Shin
Michèle Garoche
www.javapassion.com

“Learning is fun!”

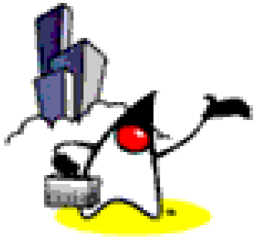
Agenda

- What is and Why network security?
- What security services do we need?
- Cryptographic process
- Public key vs. Secret key scheme
- Digital signing, Tamper-proofing & Encrypting
- Security (Cryptographic) technologies
- Key distribution and management
    - Kerberos, Certificate
- Security needs for E-commerce

Security/Cryptographic systems

- Focus of this talk
  - Network security
  - Distributed computing
  - Protection of network-based apps, data, resource
- Will not cover
  - Physical security
  - Stand-alone system security
  - Personnel issues
  - Policy issues

What is and Why Network Security?



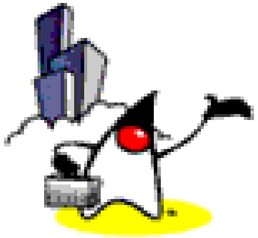
Why Network Security?

- for Distributed computing
  - Logical set of services distributed over the network
  - Physical security model (mainframe model) does not work anymore
- for Internet and Web
  - Increase of security threat in terms of both scale and frequency
  - More stringent security for E-commerce and B2B

Common Security Threats

- Identity interception
- Masquerading
- Replay attack
- Data interception and manipulation
- Repudiation
- Denial of service

What Security Services Do we need?



Security Needs of an Enterprise

- Single sign-on
  - Internet and intranet
- Controlled access to corporate information
- Secure business transaction over Internet
- Centralized, easy to use security admin tools
- Transparency of security features
  - end users should not be exposed to the underlying security schemes
- Interoperable security systems
  - Various PKI schemes, Kerberos

Network Security Needs

- Authentication (Identity verification)
- Access control (Authorization)
- Data confidentiality (Privacy)
- Data integrity (Tamper-proofing)
- Non-repudiation (Proof of transaction)
- Auditing

Authentication

- Verification of identity
  - Making sure that a user (organization, software entity) is who he claims to be (or what it claims to be)
  - Prevents Identity interception, Masquerading
- Schemes
  - In a non-networking environment, your driver license, with a picture, could be used to prove that you are who you claim to be
  - In a networking environment, digital signing is used to perform identity verification

Data Confidentiality (Privacy)

- Protects the information on the wire from prying eyes
- Schemes
  - Encrypting data by Cryptographic system
    - Clear text data + Key -> Encryption technology -> Cyphertext
    - Key could be either “shared (secret, symmetric) key” or “public (asymmetric) key”

Access Control (Authorization)

- Specifies which who can access what resources under what context
- Access control information can be maintained by either directory service or the resources themselves
  - File service, Database service (access control information is maintained by resources themselves)
- Schemes
  - ACLs- List of users and groups and their access rights in LDAP server
  - XACML

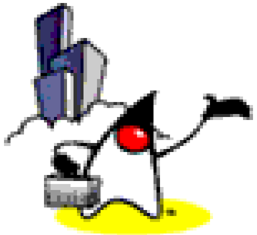
Data Integrity (Tamper-proofing)

- Prevents data tampering while data is on the wire
  - Making sure data received by the receiver is the same data sent by the sender
- Schemes
  - Digital hashing (Digital Checksum, Message Digest)
  - Usually this digital hash is used as base data for digital signing
    - message digest can be a small fixed size of data regardless of the size of original data

Non-repudiation

- Being able to prove to a 3rd-party that a transaction actually happened
  - Protects senders as well as recipients
- Schemes
  - In a non-networking environment, when you purchase merchandise using your credit card, the retailer can prove that you made a purchase
  - In a networking environment, digital signing is used

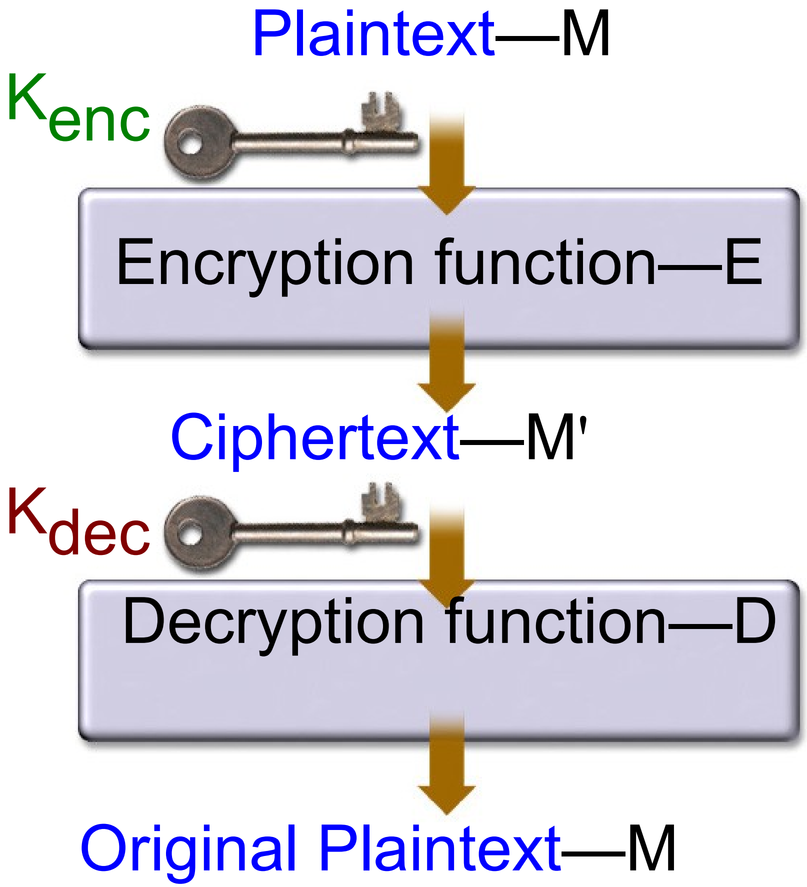
Cryptographic Process



Terminology

- Encrypt, Encipher, Encode: the process of converting plaintext to ciphertext
  - Encryption algorithm: a particular mathematical procedure of encrypting/decrypting
  - Key: information that is used to encrypt or decrypt information in a distinctive way
    - Secret Key (Symmetric, Shared)
    - Public Key (Asymmetric)
- Cryptography: mechanisms to protect information by applying “encryption” to it that are hard to reverse without secret knowledge

Cryptographic Process



M is the original message

K_{enc} is encryption key

M' is the scrambled message

K_{dec} is decryption key

It is "hard" to get M just by knowing M'

E and D are related such that

$E(K_{enc}, M) = M'$

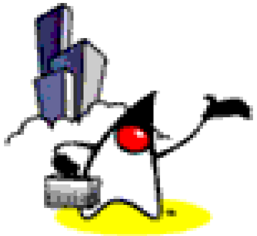
$D(K_{dec}, M') = M$

$D(K_{dec}, E(K_{enc}, M)) = M$

Cryptographic technologies (based on Security layers)

- Link layer encryption
- Network layer encryption
  - IPSEC, VPN, SKIP
- Transport layer
  - SSL, PCT(Private Communication Technology)
- Application layer
  - PEM (Privacy Enhanced Mail)
  - PGP (Pretty Good Privacy)
  - SHTTP

Public Key versus Secret key



Cryptographic Technologies - Secret key vs. Public key

- Key Management and distribution
  - Public key is easier to distribute than the secret key
- Encryption algorithms
- Key length
- Performance
  - Secret key scheme is much much faster
- Security services possible
  - Digital signing is only possible with public key
- Suitability to intranet or internet
  - Public key

Secret Key Encryption

- Sender and receiver share a secret key
  - Same secret key is used for both encryption and decryption
- Pros
  - Fast and efficient
- Cons
  - Secure distribution of keys is a problem: Not suitable for Internet

Public Key Encryption

- Uses a pair of keys: one public, the other private
  - Only private key needs to be kept secret
- The pair of keys is produced by a mathematical algorithm
  - It’s impossible to determine the value of the private key by knowing the public key
- One key is used for encryption and the other is used for decryption

Public Key Encryption (Cont.)

- Pros
  - Easier key management and distribution
    - No need to distribute secret key: More suitable for internet
  - Digital signing is possible
  - Broader ISV, products support
- Cons
  - Slower than secret key encryption
    - It is much more demanding on computing resources
  - Validation of public keys still needs to be done
    - Certificate Authority (CA)
  - Revocation of a public key is difficult

Public key and Secret key schemes are used together

- In real life the Public key and Secret key schemes are used in tandem
  - SSL is a good example
- Public key
  - Exchange of session specific secret keys (Session Key)
  - Easy key distribution, digital signing
- Secret key
  - Encryption of the user data
  - Performance

Comparison of Key Schemes

Public Key

- Encryption and decryption keys are different
- Key distribution is easier
- Public key cryptography is very slow
- Examples: RSA

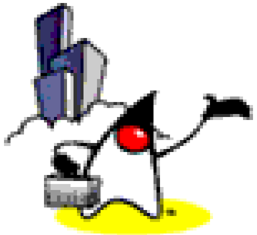
Secret Key

- Encryption and decryption keys are the same
- Key distribution is an issue
- Private key cryptography is faster
- Examples: DES, AES

Session Key

- Key negotiation and encryption are separate
- Best of both approaches
- Examples: SSL

Digital Signing, Tamper-proofing & Encrypting



Digital Signing

- Used for authentication (verifying an identity) and non-repudiation
- Uses public/private key pair
- Steps for digital signing
  - Sender creates message digest from the data
  - Sender enciphers the message digest with his private key
  - If receiver can decipher received message digest with the sender’s public key, the data must be from the sender

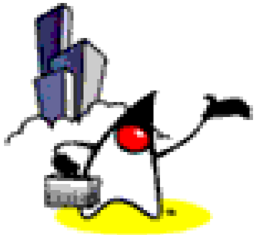
Encrypting

- Used for data confidentiality
- Can use either “public/private” key pair or secret (symmetric) key
- Steps for encrypting using “public/private” key pair
  - Sender encrypts data with receiver's public key.
  - Receiver then decrypts data with his private key. (Only he can decrypt it since only he knows his private key.)

Tamper-Proofing (Integrity)

- Performed as part of digital signing
  - Sender creates message digest from the data to be sent before signing
  - Receiver deciphers the signed message digest that he received from the sender (as part of authentication)
  - Receiver also creates his own message digest from the data it received
  - Receiver then compares the one that he received with the one that he created and sees if they match. If they match, then the data must not have been changed.

Security (Cryptographic) Technologies



Encryption Technology Issues for both Secret and Public keys

- Encryption Strength
  - Algorithm
  - Key length
  - Implementation
- Key distribution/management

Secret-key encryption algorithms (Symmetric algorithms)

- DES (Data Encryption Standard) - 56bit
- Triple DES-112bit
- IDEA (International Data Encryption Algorithm)
  - 128bit key
  - More complex (complete) than DES but the speed is comparable
  - Used in PGP
- RC2 and RC4
- Skipjack (Clipper)
  - Two-master keys

Public-key encryption algorithms (Asymmetric algorithms)

- Based on mathematical computations that are easy to compute in one direction but are practically impossible in the reverse direction
  - Diffie-Hellman(DH): Exponentiation is easy but computing discrete logarithms from the resulting value is practically impossible
  - RSA: Multiplication of two large prime numbers is easy but factoring the resulting product is practically impossible

Diffie-Hellman (DH) algorithm

- Private key and Public key generation
- Example between Alice and Bob
  - Each generates random number (private key), X & Y
    - X is private key of Alice
    - Y is private key of Bob
  - Each exponentiates the shared public data A with their private key, generates a public key
    - (A power of X) is the public key for Alice
    - (A power of Y) is the public key for Bob
  - From public key, (A power of X) for Alice and (A power of Y), it is impossible to guess private keys X and Y

Diffie-Hellman (DH) algorithm

- Generation of common secret key is possible
  - Alice has
    - Private key of herself, X
    - Public key of Bob, (A power of Y)
  - Bob has
    - Private key of himself, Y
    - Public key of Alice, (A power of X)
  - The common secret key can be computed if each exponentiate each other's public key with their private key and they are the same
    - Alice - (A power of Y) power of X
    - Bob - (A power of X) power of Y

RSA algorithm

- Used for authentication, data integrity, data privacy and non-repudiation
- Most widely used public key encryption algorithm
  - SSL, PGP, PEM, RSA digital signatures
- P * Q = N, Private key is computed from P and Q. The Public key is N
- Foundation of PKCS (Public Key Cryptography Standards)
  - Use of RSA and DES for strong authentication
  - Sun, Microsoft, Lotus endorsement

Encryption Algorithm strength

- Public key encryption has not, for all practical purposes, been broken yet
- RSA’s strength is based on the fact that it is not feasible, for all practical purposes, to factor numbers containing 150 or more digits

Key length

- Directly related encryption strength
- If encryption algorithm can’t be broken, the next best attack is to find the key by brute force
  - Algorithms are well-published
  - By “being broken”, I was referring to finding flaws in the algorithm
- Key’s protection rises exponentially with its length

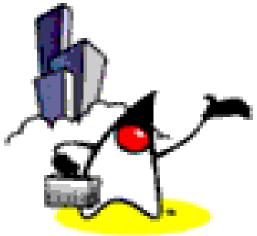
Key length (Cont.)

- Keys in public key encryption are longer than ones in secret key encryption
- Secret key encryptions
  - DES (56 bits)
  - Triple DES (112 bits)
  - Skipjack (80 bits)
  - IDEA (128 bits)
- Public key encryptions
  - Minimum 512 bits (150 decimal digits) up to 2048 bits
  - Requires serious computing power

Performance

- Using public key to encrypt entire messages or files is not practical from performance perspective
  - Public key encryption isn’t used to sign an entire message but rather only the message digest
- DES is 100 times faster than public key scheme using software and 1000 to 10,000 times faster using hardware
- This is the reason why public key is used to exchange the secret key, which is then used to encrypt actual data

Key Management & Distribution (Kerberos, Certificate)



Key Management & Distribution

- How keys are generated, stored, managed and revoked
- How keys are distributed
- This is an issue to both secret and public key encryption systems
  - Secret key: via Key Distribution Center (KDC), Kerberos
  - Public key: via Certificate (PKI)

Secret Key Management & Distribution Techniques

- Use public key encryption to exchange newly generated secret key
  - Diffie-Hellman (DH) key exchange or
  - Use RSA to send Secret key to the receiver
- Start out by using a previously agreed upon secret key
  - Immediately generate a new secret key, which is used for data encryption for a specific period of time and then generate a new secret key
- Key Distribution Center (KDC) - ANSI X9.17, Kerberos

Key Distribution Center (KDC)

- No need for a pair-wise key for every pair of hosts
- Each principal has a master key for communicating with KDC
- Scenario - Alice talking to Bob securely
  - Alice asks for Session key from KDC
  - KDC uses random number generator to generate a fresh Session key
  - KDC encrypts it with Alice’s and Bob’s master keys
  - KDC sends the encrypted Session keys to Alice
  - Alice sends the “encrypted Session key with Bob’s master key” to Bob
  - Now they have a common Session key

Kerberos

- Authenticates the identity of network principals
    - Strong authentication
        - Username/Current-time/encryption initial contact
        - “Shared secret key” between principals and KDC
        - Passwords never on the wire
        - Mutual authentication
- Single sign-on solution
- Cross-realm operation
- Delegation

Kerberos (Cont.)

- Holds a database of all principals and their master keys
- This database needs to be carefully protected
  - Server needs to be physically secured
  - The master keys in the database are all encrypted with the server’s own private master key
- Never maintains the session key internally
  - Session key is kept in the encrypted “ticket-granting-ticket” (TGT)
  - Immune to server crash

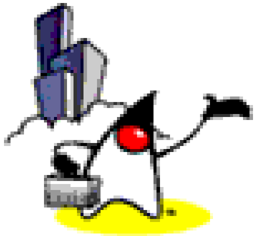
Kerberos drawbacks

- Each application needs to be kerberosized
- Single point(s) of Security risk/failure
  - KDC system (OS, file system) itself must be secure
  - Requires physically secure kerberos sever(s)
  - KDC master key needs to be highly protected
  - Potential performance bottleneck
- Kerberos v5 is not exportable (v4 is)

Public key, Certificate management/distribution

- There is no secret key distribution problem
- We still need a trusted 3rd-party (CA) to validate public keys
  - CA creates a Certificate for a certain user (Binding)
  - Certificate contains the user’s public key and ids
  - Public key is encrypted by CA’s private key (CA’s signature)
  - Users then validate the Certificate by CA’s public key
- Certificates can be transmitted over insecure network and stored in insecure storage

Certificates



Certificate Management issues (PKI Operations)

- Certificate generation
- Certificate lifetime management
- Certificate revocation (thorny issue)
- Certificate publishing
- Certificate storage
  - Directory server, DNS, NIS, NIS+, even plain files
- Certificate distribution
- Hierarchy of CA’s

Certificate formats

- X.509
  - Principal name
  - Public key
- PGP (Pretty Good Privacy)

Certificate distribution

- Transparent distribution
  - Directory service
    - X.500, X.509
    - LDAP
  - Key exchange
    - IPSEC key management protocols: SKIP, ISAKMP
    - SSL, PCT
- Interactive distribution
  - Email requests
  - Web sites
  - Finger requests

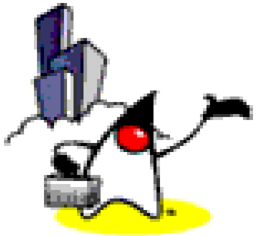
Certificate Authority (CA)

- Generates certificates
- Signs certificates with its own private key
- CA structures
  - Single centralized CA
    - Bottleneck
    - No flexibility to accommodate certificate policy
  - Multiple Cas
  - Hierarchy of CAs
    - Delegation of ‘certification generation” authority
    - Root CA signs certificates of next level CAs

PEM and PGP CA model

- PEM (RFC 1422)
  - One single global hierarchical structure
  - The root CA is the Internet Policy Registration Authority(IPRA)
  - The next level CA is the Policy Creation Authority(PRA)
  - The next level has the organizational Cas
  - Not much industry support
- PGP
  - Designed for individual users to authenticate each other
  - Each individual is his own CA

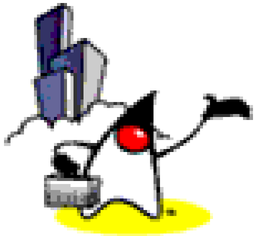
Server Authentication by Browser



Server authentication by Browser

- Server certificates are issued and signed by a commercial CA. For example, Verisign, Inc.
- The certificates of well-known CAs are pre-installed in every copy of browser
- You can add certificates of other CAs later on
- When the browser connects to a secure SSL server, the server will send its certificate to the browser client.
- The browser then validates it using the public key of the well known CA of which it has prior knowledge
- Transparent operation to end users

Cryptographic Technologies at the IP Layer



Cryptographic technology types - Location within a system

- Link layer encryption
- Network layer (IP layer) encryption
  - IPSEC, VPN, SKIP
- Transport layer
  - SSL, PCT(Private Communication Technology)
- Application layer
  - PEM (Privacy Enhanced Mail)
  - PGP (Pretty Good Privacy)
  - SHTTP

Requirements for IP layer security

- Cryptographic system designed specifically for TCP/IP
- Security services are between sites (or hosts) and not between individuals or apps
- Basis for VPN support
- Designed to work over public and insecure Internet
- Should accommodate existing TCP/IP apps
- Should accommodate existing Internet infrastructure – there should be no change in routers or ISPs

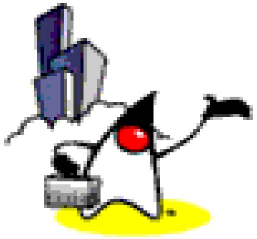
IPSEC (IP Security Protocol)

- Originally was part of IPv6, but adapted to IPv4
- Provides data integrity, data privacy services
  - Authentication Header (AH): Digital checksum (MD5)
  - Encapsulating Security Payload (ESP): Encryption (DES)
- Sender of IP packet specifies Security Association for each IP packet
  - Specification of the crypto method to be used
  - Keys to be used by the crypto methods
  - IP addresses of the sender and the receiver

IPSEC key management

- Manual keying
- Simple Key Interchange Protocol (SKIP)
  - Developed by Sun for VPN (SunScreen)
  - Designed for key exchange by special header
  - Special header (20 to 30 bytes) for every IP packet
  - Supports DH key exchange
- ISAKMP
  - Management of Security Associations as well as key exchange
  - Supports Oakley

Message Digest



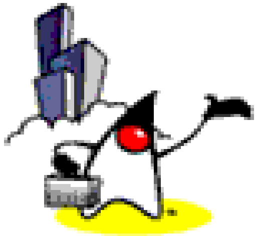
MD (Message Digest)

- Calculating a checksum using mathematical algorithms
- Properties
  - It is impossible to guess the original data from the message digest
  - Regardless of the size of the original data the resulting message digest can be a fixed size
    - This is the reason why it is used for digital signing
  - A change of a single bit in the original data will result in a different message digest
    - Possibility of generating same message digest is practically non-existent

MD (Message Digest) Standards

- MD4, MD5 (RFC 1320, 1321)
  - 128-bit digest from messages of any length
  - Developed by Ron Rivest
- SHA (Secure Hash Algorithm)
  - 160-bit digest
  - Developed by NIST
  - More secure but slower than MD4 and MD5

Security Needs for E-commerce



Secure Internet Communication

- Customer requirements
  - E-commerce
  - Business to business transaction
  - Secure access to corporate data
- Characteristics of Internet vs. Intranet
  - Millions of users with no prior contact
  - Data over insecure communication channel
  - No centralized controlling organization
- Functional requirement
  - Has to be fast and reliable

Thank you!

Sang Shin
Michèle Garoche
http://www.javapassion.com
“Learning is fun!”