CSC 4103 - Operating Systems
Spring 2007

Lecture - XXI

Protection and Security - II

Tevfik Koşar

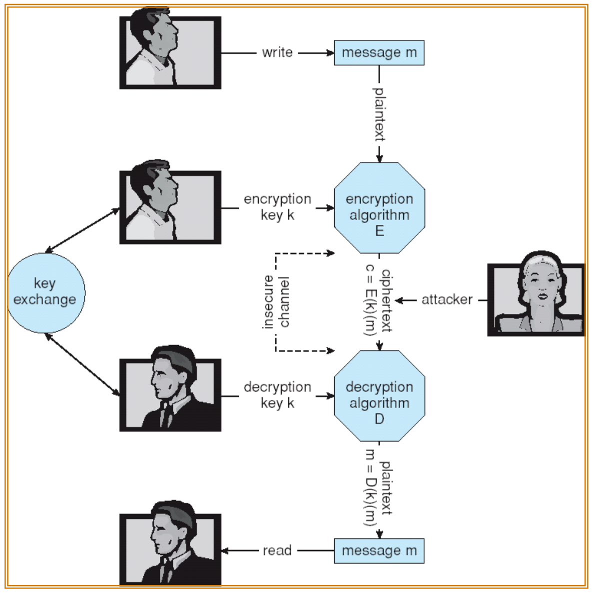
Louisiana State University

April 17th, 2007

Symmetric Encryption

- Same key used to encrypt and decrypt
  - *E*(*k*) can be derived from *D*(*k*), and vice versa
- **DES** is most commonly used symmetric block-encryption algorithm (created by US Govt)
  - Encrypts a block of data at a time (64 bit messages, with 56 bit key)
- **Triple-DES** considered more secure (repeat DES three times with three different keys)
- Advanced Encryption Standard (**AES**) replaces DES
  - Key length upto 256 bits, working on 128 bit blocks
- **Twofish, RC4, RC5 ..** other symmetric algorithms
- **RC4** is most common symmetric stream cipher (works on bits, not blocks), but known to have vulnerabilities
  - Encrypts/decrypts a stream of bytes (i.e wireless transmission, web browsers)
  - Key is a input to psuedo-random-bit generator
    - Generates an infinite **keystream**

Symmetric Encryption



Asymmetric Encryption

- Encryption and decryption keys are different
- Public-key encryption based on each user having two keys:
  - public key – published key used to encrypt data
  - private key – key known only to individual user used to decrypt data
- Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme
  - Most common is RSA *(Rivest, Shamir, Adleman)* block cipher

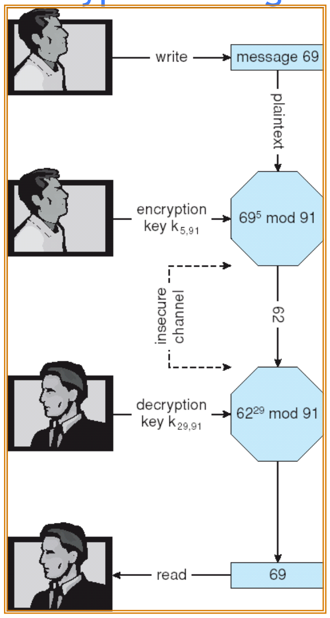
Asymmetric Encryption (Cont.)

- Formally, it is computationally infeasible to derive $D(k_d$ $, N)$ from $E(k_e , N)$, and so $E(k_e , N)$ need not be kept secret and can be widely disseminated
  - $E(k_e , N)$ (or just k_e) is the **public key**
  - $D(k_d , N)$ (or just k_d) is the **private key**
  - N is the product of two large, randomly chosen prime numbers p and q (for example, p and q are 512 bits each)
  - Select k_e and k_d, where k_e satisfies $k_e k_d \bmod (p-1)(q-1) = 1$
  - Encryption algorithm is $E(k_e , N)(m) = m^{k_e} \bmod N$,
  - Decryption algorithm is then $D(k_d , N)(c) = c^{k_d} \bmod N$

Asymmetric Encryption Example

- For example. choose $p = 7$ and $q = 13$
- We then calculate $N = 7*13 = 91$ and $(p-1)(q-1) = 72$
- We next select k_e relatively prime to 72 and< 72, yielding 5
- Finally,we calculate k_d such that $k_e k_d \bmod 72 = 1$, yielding 29
- We how have our keys
  - Public key, $k_{e,}$ N = 5, 91
  - Private key, k_d , N = 29, 91
- Encrypting the message 69 with the public key results in the cyphertext 62 ($E=69^5 \bmod 91$)
- Cyphertext can be decoded with the private key
  - Public key can be distributed in cleartext to anyone who wants to communicate with holder of public key

Encryption and Decryption using RSA Asymmetric



Cryptography (Cont.)

- Note symmetric cryptography based on transformations, asymmetric based on mathematical functions
  - Asymmetric much more compute intensive
  - Typically not used for bulk data encryption
  - Used for authentication, confidentiality, key distribution

Authentication

- Constraining set of potential senders of a message
  - Complementary and sometimes redundant to encryption
  - Also can prove message unmodified
- Algorithm components
  - A set K of keys
  - A set M of messages
  - A set A of authenticators
  - A function $S : K \rightarrow (M \rightarrow A)$
    - That is, for each $k \in K$, $S(k)$ is a function for generating authenticators from messages
    - Both S and $S(k)$ for any k should be efficiently computable functions
  - A function $V : K \rightarrow (M \times A \rightarrow \{\text{true, false}\})$. That is, for each $k \in K$, $V(k)$ is a function for verifying authenticators on messages
    - Both V and $V(k)$ for any k should be efficiently computable functions

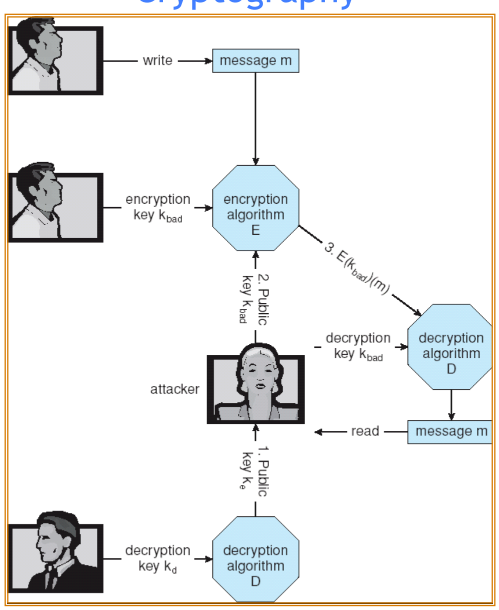
Authentication (Cont.)

- For a message m, a computer can generate an authenticator $a \in A$ such that $V(k)(m, a)$ = `true` only if it possesses $S(k)$
- Thus, computer holding $S(k)$ can generate authenticators on messages so that any other computer possessing $V(k)$ can verify them
- Computer not holding $S(k)$ cannot generate authenticators on messages that can be verified using $V(k)$
- Since authenticators are generally exposed (for example, they are sent on the network with the messages themselves), it must not be feasible to derive $S(k)$ from the authenticators

Key Distribution

- Delivery of symmetric key is huge challenge
  - Sometimes done **out-of-band,** via paper documents or conversation
- Asymmetric keys can proliferate – stored on **key ring**
  - Even asymmetric key distribution needs care – man-in-the-middle attack

Man-in-the-middle Attack on Asymmetric Cryptography



Digital Certificates

- Proof of who or what owns a public key
- Public key digitally signed a trusted party
- Trusted party receives proof of identification from entity and certifies that public key belongs to entity
- Certificate authority are trusted party – their public keys included with web browser distributions
  - They vouch for other authorities via digitally signing their keys, and so on

Encryption Example - SSL

- Insertion of cryptography at one layer of the ISO network model (the transport layer)
- SSL – Secure Socket Layer (also called TLS)
- Cryptographic protocol that limits two computers to only exchange messages with each other
  - Very complicated, with many variations
- Used between web servers and browsers for secure communication (credit card numbers)
- The server is verified with a **certificate** assuring client is talking to correct server
- Asymmetric cryptography used to establish a secure **session key** (symmetric encryption) for bulk of communication during session
- Communication between each computer then uses symmetric key cryptography

User Authentication

- Crucial to identify user correctly, as protection systems depend on user ID
- User identity most often established through *passwords*, can be considered a special case of either keys or capabilities
  - Also can include something user has and /or a user attribute
- A password can be associated with each resource (eg. File)
- Different passwords may be associated with different access rights
  - Eg. Reading, updating, and deleting files
- Passwords must be kept secret
  - Frequent change of passwords
  - Use of “non-guessable” passwords
  - Log all invalid access attempts
- Passwords may also either be encrypted or allowed to be used only once

Password Vulnerabilities

- Password length
  - A four digit password would take less than 5 seconds to crack
- Password combination
  - Should use combination of digits, upper and lower case letters, and other characters
- Never write your password somewhere, memorize it
- Periodically change your password
- Do not use the following in your password:
  - Name, lastname
  - Username
  - Date of birth, zipcode, other personal info
- Do not share your accounts with others

Encrypted Passwords

- How to keep a password secure within the computer?
- UNIX-type systems keep the password lists encrypted
  - Impossible to invert
  - Simple to compute

  ==> one-way encryption
- Comparison is performed between encoded passwords
- Another level of protection:
  - Encrypted password file is only readable to root

Biometrics

- Instead of passwords, use biomentric measures
  - Palm-readers
  - Finger-print-readers
  - Iris scanners
  - Voice recognition
- Multi-factor authentication
  - Use a combination of different authentication mechanisms

Implementing Security Defenses

- **Defense in depth** is most common security theory: using multiple layers of security
- Security policies
  - Eg. Policies on user passwords and accounts
- Vulnerability assessment compares real state of system / network compared to security policy
  - Eg. Assessment to passwords, network ports
- Intrusion detection endeavors to detect attempted or successful intrusions
  - **Signature-based** detection
    - Examine system input or network traffic for specific behavior patterns
  - **Anomaly detection**
    - Detect differences from normal behavior
    - Can also detect previously unknown methods of intrusion: **zero-day** attacks
  - **False-positives** (false alarms) and **false-negatives** (mussed intrusions) are problem
- Auditing, accounting, and logging of all or specific system or network activities

Any Questions?

Reading Assignment

- Read chapter 14 and 15 from Silberschatz.

Acknowledgements

- “Operating Systems Concepts” book and supplementary material by Silberschatz, Galvin and Gagne.