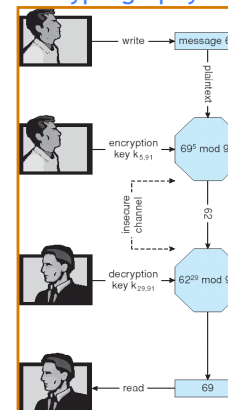


LECTURE - XXI PROTECTION AND SECURITY - II

Tevfik Koşar

Louisiana State University
April 22nd, 2008

Encryption and Decryption using RSA Asymmetric Cryptography



Authentication

- Constraining set of potential senders of a message
 - Complementary 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 $V : K \rightarrow (M \times A \rightarrow \{\text{true}, \text{false}\})$. 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 $V(k)(m, a) = \text{true}$ 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

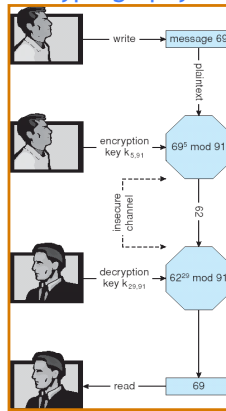
Constraining both Sender & Receiver

- generate an authenticator $a \in A$ such that $V(k)(m, a) = \text{true}$ only if it possesses $S(k)$
- Encrypt this authenticator with the public key of the targeted receiver
 - $E(k)(m, a) = C$

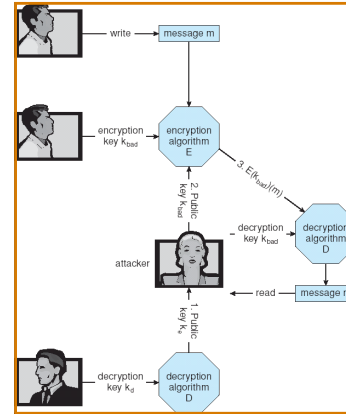
Key Distribution

- 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

Encryption and Decryption using RSA Asymmetric Cryptography



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
 - i.e. VeriSign, Comodo etc.

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 biometric measures
 - Palm-readers
 - Finger-print-readers
 - Iris scanners
 - Voice recognition
- Multi-factor authentication
 - Use a combination of different authentication mechanisms

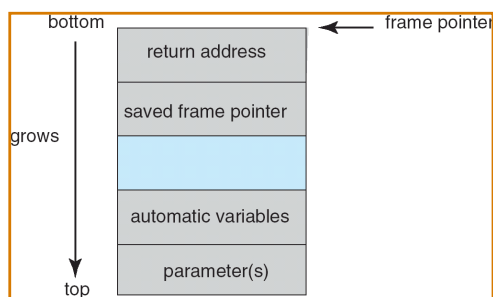
Program Threats

- Trojan Horse
 - Code segment that misuses its environment
 - Exploits mechanisms for allowing programs written by users to be executed by other users
 - Spyware, pop-up browser windows, covert channels
- Trap Door
 - Specific user identifier or password that circumvents normal security procedures
 - Could be included in a compiler
- Logic Bomb
 - Program that initiates a security incident under certain circumstances
- Stack and Buffer Overflow
 - Exploits a bug in a program (overflow either the stack or memory buffers)

C Program with Buffer-overflow Condition

```
#include <stdio.h>
#define BUFFER_SIZE 256
int main(int argc, char *argv[])
{
    char buffer[BUFFER_SIZE];
    if (argc < 2)
        return -1;
    else {
        strcpy(buffer, argv[1]);
        return 0;
    }
}
```

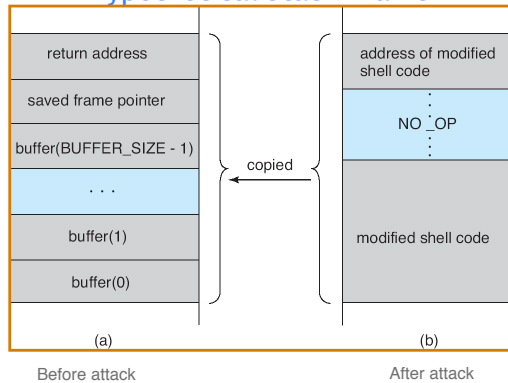
Layout of Typical Stack Frame



Modified Shell Code

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    execvp(`\bin\sh'', `bin\sh'', NULL);
    return 0;
}
```

Hypothetical Stack Frame



Program Threats (Cont.)

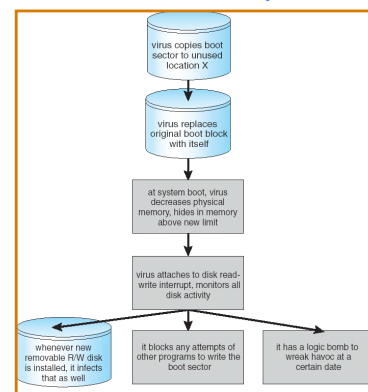
- Viruses
 - Code fragment embedded in legitimate program
 - Very specific to CPU architecture, operating system, applications
 - Usually borne via email or as a macro
- Visual Basic Macro to reformat hard drive

```
Sub AutoOpen ()
    Dim oFS
    Set oFS = CreateObject('Scripting.FileSystemObject')
    vs = Shell('c:\command.com /k format c:', vbHide)
End Sub
```

Program Threats (Cont.)

- Virus dropper inserts virus onto the system
- Many categories of viruses, literally many thousands of viruses:
 - **File** (appends itself to a file, changes start pointer, returns to original code)
 - **Boot** (writes to the boot sector, gets exec before OS)
 - **Macro** (runs as soon as document containing macro is opened)
 - **Source code** (modifies existing source codes to spread)
 - **Polymorphic** (changes each time to prevent detection)
 - **Encrypted** (first decrypts, then executes)
 - **Stealth** (modify parts of the system to prevent detection, eg read system call)
 - **Tunneling** (installs itself as interrupt handler or device driver)
 - **Multipartite** (can infect multiple parts of the system, eg. Memory, bootsector, files)
 - **Armored** (hidden and compressed virus files)
- Browser virus, keystroke logger .etc

A Boot-sector Computer Virus



System and Network Threats

- **Worms** - use spawn mechanism; standalone program
- Internet worm (*Robert Morris, 1998, Cornell*)
 - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs
 - **Grappling hook** program uploaded main worm program
- **Port scanning**
 - Automated attempt to connect to a range of ports on one or a range of IP addresses
- **Denial of Service**
 - Overload the targeted computer preventing it from doing any useful work
 - Distributed denial-of-service (**DDOS**) come from multiple sites at once

The Morris Internet Worm

