

# ISA 666 Internet Security Protocols

Duminda Wijesekera  
dwijesek@gmu.edu  
(703) 993-1578

Yours truly,

- **Name** = Duminda Wijesekera
- **Web** = <http://www.gmu.edu/~duminda>
- **Email** = [dwijesek@gmu.edu](mailto:dwijesek@gmu.edu)
- **Phone** = (703) 993-1578
- **Office** = Room 351 Science & Technology II
- **Office hours** = Tuesday 3.00-4.00 pm
  
- **Classroom** = ST II Room 7

## The teaching assistant

- **Name** = Min Xu
- **Web** = <http://www.gmu.edu/~duminda>
- **Email** = [mxu@gmu.edu](mailto:mxu@gmu.edu)
- **Phone** = (703) 993- need to get
- **Office** = Room 468 Science & Technology II
- **Office hours** = Tuesday 2.00-4.00 pm
- **Classroom** = FAB B106

## The Textbook

- **Title:** *Network Security – Private Communication in a public world*
- **Authors:** Charlie Kaufman, Radia Pearlman and Mike Speciner
- **Edition:** Latest (second)
- **ISBN:** 0-13-046019-2
- **Publisher:** Prentice Hall
- **Alternative Book**
  - William Stallng, *Network Security Essentials, 2nd Edition*, Prentice Hall.

## On-line Resources

- **WWW page:**
  - <http://www.ise.gmu.edu/~duminda/classes/fall06/ise666/index.html>
  - For course materials, e.g., lecture slides, homework files, papers, tools, etc.
  - Will be updated frequently. So check frequently.
- Will be using the teaching material from the last semester, with some changes

## Course Grades

- **Assignments** = 40% { four assignments }
- **Midterm exam** = 30%
- **Final exam** = 30%
  
- Grades curved.
  
- **GMU academic standards and honor code applies!**

## Prerequisites

- INFS 612 (Data Communication and Distributed Processing) or Equivalent
- INFS 601 (Operating Systems) or Equivalent
- Strong Programming
- Basic Knowledge and Skills in Discrete Mathematics

## Policies on Absences

- Excused from exams only under a university accepted condition.
- **Need to inform prior to absence!**
- One chance to take a makeup an exam for acceptable absences.
- No makeup for homework assignments.

## Policies on Late Submissions

- Homework deadlines are **hard**.
- Late homework will be accepted with a 10% penalty in for each day past due.
- Once a homework assignment is discussed in class or solution posted, no late submissions will be accepted.

## Academic Integrity

- All University, School, and Department Policies Against Academic Dishonesty Will Be Strictly Enforced.
- University Academic Policies at
  - <http://www.gmu.edu/catalog/apolicies/#Anchor12>
- ISE Department Honor System and Code
  - <http://www.ise.gmu.edu/Honor.html>

## Detailed Course Description

- Study security problems on the Internet
  - What are the security issues?
    - Vulnerabilities, tradeoffs and their solutions
- Study principles, techniques and applying them in building secure networks
  - Basic cryptography
  - Authentication, authorization
  - Digital signature

## Course Description Continued...

- Examine existing Internet security techniques and protocols
  - IPSEC/VPN
  - Firewall
  - SSL/TLS
- Beyond traditional applications
  - Wireless systems
  - Adhoc networks and their applications
- Limitations of existing protocols

## Course Objectives

- Understand of basic issues, concepts, principles, and techniques in Internet security.
  - Basic definitions
  - Cryptography
  - Authentication, authorization
  - Network security
  - Intrusion detection and response
- Be able to determine appropriate mechanisms to enforce a chosen security requirement.
  - Example
    - Only you can change your address
    - Only I can give your grade in this class

## Course Outline

- Network-based attacks
  - Passive, active attacks
  - Eavesdropping, unauthorized access, modification, deletion, forgery of confidential information
  - Denial-of-service attack
- Basic concepts security
  - Confidentiality, integrity, identity, anonymity, availability
  - Security policies, security mechanisms, assurance

## Course Outline (Cont'd)

- Cryptography
  - Basic number theory
  - Secret key cryptosystems
  - Public key cryptosystems
  - Hash function
  - Key management

## Course Outline (Cont'd)

- Identification and Authentication
  - Basic concepts of identification and authentication
  - Password authentication
  - Kerberos
  - Security handshake pitfalls
  - Attack on authentication and identification
    - Identity theft
    - Impersonation

## Course Outline (Cont'd)

- IPSEC/VPN
  - Security association
  - Authentication header (AH)
  - Encapsulating Security Payload (ESP)
  - Transport, tunnel modes
  - Virtual Private Network (VPN)
- IPSEC Key Management
  - ISAKMP/Oakley
- PKI

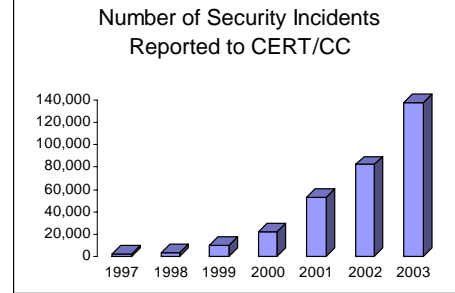
## Course Outline (Cont'd)

- Firewall
  - Packet filtering
  - Stateful inspection
  - Transparent proxy
  - Enclave
  - Firewall limitations
  - Firewalls and Virtual Private Network
- SSL/TSL

## Course Outline (Cont'd)

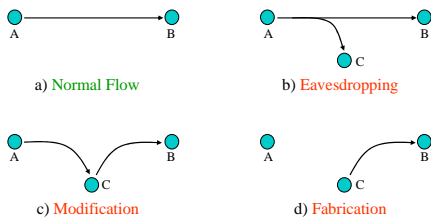
- Beyond basics
  - Anonymity and privacy
  - Virus, worms, Trojan horse
- Applications
  - Wireless systems
  - Adhoc networks
  - Telecommunication systems
  - Control system security

## Security Problems on the Internet



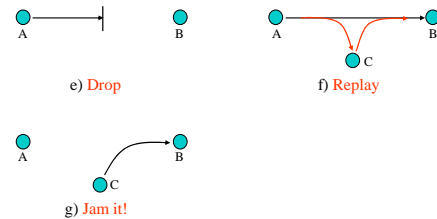
## Behavior in networks - content

- Normal and their *unauthorized* deviations



## Behavior on networks – continued!

- Start From The Basics



## Other modifications

- Delay
- Change the order of packets

## Traditional definitions from security

- **Confidentiality**
  - Prevent information from being exposed to unintended party
- **Integrity**
  - Assure that the information has not been tempered
- **Identity**
  - Assure that the party of concern is authentic - it is what it claims to be
- **Availability**
  - Assure that unused service or resource is available to legitimate users
- **Anonymity**
  - Assure that the identity of some party is remain anonymous
- **Non-Repudiation**
  - Assure that authenticated party has indeed done something and it can not deny it

## Real-world examples

- **Confidentiality**
  - An employee should not come to know the salary of his manager
- **Integrity**
  - An employee should not be able to modify the employee's own salary
- **Identity**
  - An employee should be able to uniquely identify and authenticate himself/herself
- **Availability**
  - Paychecks should be printed on time as stipulated by law
- **Anonymity**
  - The manager should not know who had a critical review for him
- **Non-repudiation**
  - Once the employee has cashed out the paycheck, he/she can't deny it

## Real-World Network Based Attacks

- Distributed Denial of Service (DDOS) attacks
- Worm and Virus attacks (e.g., worm Sasser)
- Monitoring and capturing of network traffic
  - User IDs, passwords, and other information are often stolen on Internet
- Exploiting software vulnerabilities (MS-Windows)
- Unauthorized access to resources
  - Disclosure, modification, and destruction of resources
- Using a compromised system as a **stepping stone** for an attack
- Masquerade as an authorized user or end system
- Data driven attacks
  - Importing infected code
- E-Mail Forgery and Spam

## Contributing Factors

- **Lack of awareness** of threats and risks of from the network
  - Security measures are often not considered until an Enterprise has been penetrated by malicious users
- **Wide-Open Network Policies**
  - Many Internet sites allow wide-open Internet access
- **Vast majority of network traffic is unencrypted**
  - Network traffic can be monitored and captured

## Contributing Factors (Cont'd)

- **Lack of Security in TCP/IP Protocol Suite**
  - Most TCP/IP protocols were not built with security in mind
  - Work is actively progressing within the Internet Engineering Task Force (IETF)
- **Complexity of network security management**
- **Exploiting Software Bugs (e.g., Protocol Implementation)**
  - Example: Sendmail bugs
- **Attacker-skills always improves!**

## Existing Internet Security Mechanisms

- **Prevention**
  - Firewall
  - Authentication, authorization
  - IPSEC/VPN
  - Access control
  - Encryption
- **Detection**
  - Auditing
  - Misuse detection
  - Anomaly detection
- **Survivability**
- **Response to intrusions?**

## Existing Internet Security Mechanisms

- Security mechanisms implement functions that help to *prevent, detect, tolerate, respond* to security attacks
- Prevention is ideal, but...
  - Detection seeks to prevent by threat of punitive action
  - Detection requires that the audit trail be protected from alteration
- If can't completely prevent attack from happening, detection is the only option
- There could be attacks we can't detect, then live with it - survivable system
- Once detect the attack, then what? Active response!!!
- Cryptography underlies (almost) all security mechanisms

## Security by Obscurity

- Security by obscurity says that if we hide the inner workings of a system it will be secure
- **It is a bad idea**
- Less and less applicable in the emerging world of vendor-independent open standards
- Less and less applicable in a world of widespread computer knowledge and expertise

## Security by Legislation

- Security by legislation says that if we instruct our users on how to behave we can secure our systems
- **It is a bad idea**
- For example
  - Users should not share passwords
  - Users should not write down passwords
  - Users should not type in their password when someone is looking over their shoulder
- User awareness and cooperation is important, but cannot be the principal focus for achieving security

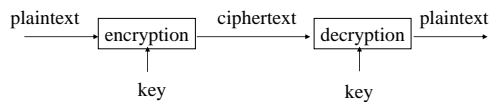
## Unique Aspects of Security

- The whole system is as strong as its weakest point
- The root cause of security problem is not the computer, but the human behavior
- Ever changing - moving target
  - countermeasures by adversary
- Conflicting requirements
  - Identity/authentication
  - Anonymity

## Cryptography

- Cryptography
  - Comes from Greek word
  - Original meaning: The art of secret writing
  - Becoming a science that relies on mathematics (number theory, algebra)
  - Process data into intelligible form, reversible, without data loss
  - Usually one-to-one (not compression)

## Encryption/Decryption



- **Plaintext:** a message in its original form
- **Ciphertext:** a message in the transformed, unrecognized form
- **Encryption:** the process that transforms a plaintext into a ciphertext
- **Decryption:** the process that transforms a ciphertext to the corresponding plaintext
- **Key:** the value used to control encryption/decryption.

## Cryptanalysis

- Ciphertext only:
  - Analyze only with the ciphertext
  - Example: Exhaustive search until “recognizable plaintext”
  - Smarter ways available
- Known plaintext:
  - Secret may be revealed (by spy, time), thus <ciphertext, plaintext> pair is obtained
  - Great for mono-alphabetic ciphers

## Cryptanalysis (Cont'd)

- Chosen plaintext:
  - Choose text, get encrypted
  - Useful if limited set of messages
- Chosen ciphertext:
  - Choose ciphertext
  - Get feedback from decryption, etc.

## Security of An Encryption Algorithm

- Unconditionally secure
  - It is impossible to decrypt the ciphertext
  - One-time pad (the key is as long as the plaintext)
- Computationally secure
  - The cost of breaking the cipher exceeds the value of the encrypted information
  - The time required to break the cipher exceeds the useful lifetime of the information

## Secret Keys v.s. Secret Algorithms

- Security by obscurity
  - We can achieve better security if we keep the algorithms secret
  - Hard to keep secret if used widely
  - Reverse engineering, social engineering
- Publish the algorithms
  - Security of the algorithms depends on the secrecy of the keys
  - Less unknown vulnerability if all the smart (good) people in the world examine the algorithms

## Secret Keys v.s. Secret Algorithms (cont'd)

- Commercial world
  - Published
  - Wide review, trust
- Military
  - Keep algorithms secret
  - Avoid giving good ideas to bad people
  - Military has access to the public domain knowledge anyway.
- **Security of a secure protocol should be based on the secrecy of the keys, rather than based on obscurity**

## Some Trivial Codes

- Caesar cipher: substitution cipher:
  - Replace each letter with the one 3 letters later
  - $A \rightarrow D, B \rightarrow E$
- Captain Midnight Secret Decoder rings:
  - shift variable by  $n$ :  $IBM \rightarrow HAL$
  - only 26 possibilities

## Some Trivial Codes (Cont'd)

- Mono-alphabetic cipher:
  - generalization, arbitrary mapping of one letter to another
  - $26!$ , approximately  $4 \times 10^{26}$
  - statistical analysis of letter frequencies

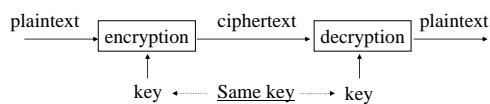
## Some Trivial Codes (cont'd)

- Poly-alphabetic Ciphers
  - A set of related mono-alphabetic substitution rules is used
  - A key determines which particular rule is chosen for a given transformation

## Types of Cryptography

- Number of keys
  - Hash functions: no key
  - Secret key cryptography: one key
  - Public key cryptography: two keys - public, private
- The way in which the plaintext is processed
  - Block cipher: divides input elements into blocks
  - Stream cipher: process one element (e.g., bit) a time

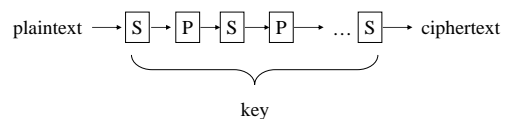
## Secret Key Cryptography



- Same key is used for encryption and decryption
- Also known as
  - Symmetric cryptography
  - Conventional cryptography

## Secret Key Cryptography (cont'd)

- Basic technique
  - Product cipher:
  - Multiple applications of interleaved substitutions and permutations



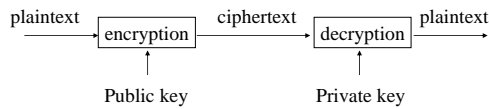
## Secret Key Cryptography (cont'd)

- Ciphertext approximately the same length as plaintext
- Examples
  - Stream Cipher: RC4
  - Block Cipher: DES, IDEA, AES

## Applications of Secret Key Cryptography

- Transmitting over an insecure channel
  - Challenge: How to share the key?
- Secure Storage on insecure media
- Authentication
  - Challenge-response
  - To prove the other party knows the secret key
  - Must be secure against chosen plaintext attack
- Integrity check
  - Message integrity code (MIC)

## Public Key Cryptography



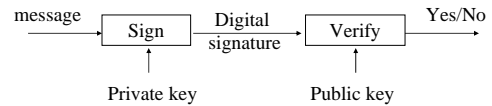
- Invented/published in 1975
- A public/private key pair is used
  - Public key can be publicly known
  - Private key is kept secret by the owner of the key
- Much slower than secret key cryptography
- Also known as
  - Asymmetric cryptography

ISE at George Mason University

ISA 666 D. Wijesekera

49

## Public Key Cryptography (Cont'd)



- Another mode: digital signature
  - Only the party with the private key can create a digital signature.
  - The digital signature is verifiable by anyone who knows the public key.
  - The signer cannot deny that he/she has done so.

ISE at George Mason University

ISA 666 D. Wijesekera

50

## Applications of Public Key Cryptography

- Data transmission:
  - Alice encrypts  $m_a$  using Bob's public key  $e_B$ , Bob decrypts  $m_a$  using his private key  $d_B$ .
- Storage:
  - Can create a safety copy: using public key of trusted person.
- Authentication:
  - No need to store secrets, only need public keys.
  - Secret key cryptography: need to share secret key for every person to communicate with.

ISE at George Mason University

ISA 666 D. Wijesekera

51

## Applications of Public Key Cryptography (Cont'd)

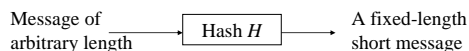
- Digital signatures
  - Sign hash  $H(m)$  with the private key
    - Authorship
    - Integrity
    - Non-repudiation: can't do with secret key cryptography
- Key exchange
  - Establish a common session key between two parties

ISE at George Mason University

ISA 666 D. Wijesekera

52

## Hash Algorithms



- Also known as
  - Message digests
  - One-way transformations
  - One-way functions
  - Hash functions
- Length of  $H(m)$  much shorter than length of  $m$
- Usually fixed lengths: 128 or 160 bits

ISE at George Mason University

ISA 666 D. Wijesekera

53

## Hash Algorithms (Cont'd)

- Desirable properties of hash functions
  - **Performance:** Easy to compute  $H(m)$
  - **One-way property:** Given  $H(m)$  but not  $m$ , it's difficult to find  $m$
  - **Weak collision free:** Given  $H(m)$ , it's difficult to find  $m'$  such that  $H(m') = H(m)$ .
  - **Strong collision free:** Computationally infeasible to find  $m_1, m_2$  such that  $H(m_1) = H(m_2)$

ISE at George Mason University

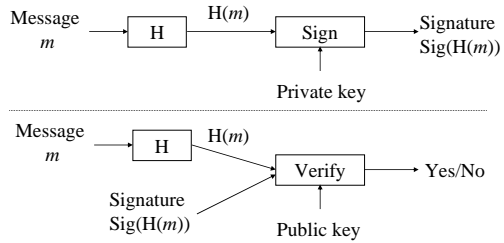
ISA 666 D. Wijesekera

54

## Applications of Hash Functions

- Primary application

- Generate/verify digital signature



## Applications of Hash Functions (Cont'd)

- Password hashing

- Doesn't need to know password to verify it
- Store  $H(\text{password} + \text{salt})$  and salt, and compare it with the user-entered password
- Salt makes dictionary attack more difficult

- Message integrity

- Agree on a secret key  $k$
- Compute  $H(m|k)$  and send with  $m$
- Doesn't require encryption algorithm, so the technology is exportable