CEN-5079
Secure Application Programming

Class 1

Bogdan Carbunar
Outline

- Administrative Issues
- Textbooks
- Security Overview
Administrative Issues

- **Staff**
  - Bogdan Carbunar, assistant professor

- **Communications**
  - E-mail: carbunar@gmail.com

- **Office Hours**
  - Thursday after class, ECS 383

- **What about you?**
Class Grading (subject to changes)

- 1 midterm worth 35%
  - Date of exam: TBA
- 1 final worth: 35%
  - Date of exam: TBA
- Homeworks: 20%
- Class participation: 10%
Class Grading: Details (cont’d)

- Class participation: 10%
  - Class presence does not mean participation
  - But it is encouraged
  - Participation means *asking* and *answering* questions
Class Grading for Online Students

- **1 midterm** worth: 40%
  - Date of exam: TBA
- **1 final** worth: 40%
  - Date of exam: TBA
- **Homeworks**: 20%
Outline

- Administrative Issues
- Textbooks
- Security Overview
Textbooks

- *Security In Computing – 4th edition*
  Pfleeger and Pfleeger

- *Cryptography and Network Security*
  William Stallings

- *Applied Cryptography – 2nd edition*
  Bruce Schneier; Available online; *Don’t need to buy it!*


- Papers assigned for reading
  - See class webpage
Outline

- Administrative Issues
- Textbooks
- Security Overview
Information & Computer Security

- Protect **information** and information systems from **unauthorized** access
- Computer Security: security applied to **computers**
  - Objective: protect information and property
  - Against theft, corruption, or natural disaster
  - Information and property remain accessible to intended users

[Source: wikipedia]
System Security

- Goals: Protect
  - Confidentiality
  - Integrity
  - Availability

System Security
Confidentiality

- Information about system or its users cannot be learned by an attacker

- Data Confidentiality:
  - Private or confidential information is not revealed to unauthorized individuals

- Privacy:
  - Users control what information about them can be
    - Collected
    - Stored
    - By whom
The system continues to operate properly, only reaching states that would occur if there were no attacker

Data Integrity
- Information and programs are changed only in specified and authorized manner

System Integrity
- System performs intended function free from unauthorized system manipulation
Availability

- Actions by an attacker do not prevent users from having access to use of the system
  - Enable access to data and resources
  - Timely response
  - *Fair* resource allocation
Examples

- **Confidentiality**
  - Student grades
  - Available only to student, parents, employer

- **Integrity**
  - Patient information e.g., allergies
  - Can lead to loss of human life

- **Availability**
  - Authentication service
  - Unavailability can lead to financial loss
More Required Concepts

- **Authenticity**
  - Being able to be verified and trusted
  - Confidence in the validity of a message (originator)

- **Accountability**
  - Actions of an entity can be traced to it
  - Tracing a security breach to a responsible party
Security is about

- Honest user (e.g., Alice, Bob, ...)
- Attacker
- How the Attacker
  - Disrupts honest user’s use of the system (Integrity, Availability)
  - Learns information intended for Alice only (Confidentiality)
Aspects of Security

- **Threat**
  - Potential for violation of security
  - Possible danger exploiting a vulnerability

- **Security attack**
  - Action that compromises the security of a system
  - Confidentiality, integrity, availability
  - Deliberate attempt to evade security mechanisms
Aspects of Security (cont’d)

- Security mechanism
  - Process designed to detect, prevent or recover from attack

- Security service
  - Enhances security of data and its transfer
  - Counter security attack
Vulnerabilities

- **Hardware: physical attack**
  - Accidental
  - Voluntary machine slaughter / theft

- **Software**
  - Buffer overflows, incomplete mediation, time of check to time of use (TOCTOU)

- **Data**
  - Confidentiality, Integrity, Availability

- **Networks**
  - Hardware + software + data +
  - *Communication media*
What is Security?

- **System correctness**
  - If user supplies expected input, system generates desired output
  - Good input $\Rightarrow$ Good output
  - More features: better

- **Security**
  - If attacker supplies unexpected input, system does not fail in certain ways
  - Bad input $\Rightarrow$ Bad output
  - More features: can be worse
Security Attacks

- **Passive Attacks**
  - Learn and use information from a system without affecting system resources

- **Active Attacks**
  - *Attempt to affect and alter system resources*
Passive Attacks

1. Alice
2. Eavesdrop
3. Bob

Perform traffic analysis on encrypted messages
Active Attacks

- Impersonation
- Replay
- Modify messages
- Denial of Service (DoS)
Impersonation

1. Hi, A, unique id

2. Eavesdrop

3. Hi, “I’m A”, unique id

May enable M to obtain A’s privileges
Replay

Alice

“You owe me $10”

Eavesdrop

Bob

Malory

Bob owes Alice $20 or Bob owes Malory $10

You owe me $10
Message Modification

1. Give Carol access
2. Eavesdrop
3. Give Malory access

Note that A’s message is delayed or removed
Why Security Vulnerabilities?

Some contributing factors

- Few courses in computer security 😊
- Programming text books do not emphasize security
- Few security audits
- *C is an unsafe language*
- Programmers have many other things to worry about
- Consumers do not care about security
- Security is expensive and takes time
Crash Course in Crypto

- Building blocks that we will define and use in the next lectures
- Will describe in more detail later
  - Encryption
  - Signature
  - Hash
Basic Terminology

- Plaintext
  - Original message
- Ciphertext
  - Coded message
- Cipher or Encryption Algorithm
  - Algorithm for transforming plaintext to ciphertext
- Key
  - Info used in cipher known only to sender/receiver
Basic Terminology (cont’d)

- Encrypt (encipher)
  - Converting plaintext to ciphertext

- Decrypt (decipher)
  - Recovering plaintext from ciphertext
Basic Terminology (cont’d)

- Cryptography
  - Study of encryption principles/methods

- Cryptanalysis (codebreaking)
  - Study of principles/methods of deciphering ciphertext without knowing key

- Cryptology
  - Field of both cryptography and cryptanalysis
Cryptosystem/Encryption System

Encryption Key

Encryption Algorithm

Ciphertext

Decryption Key

Decryption Algorithm

Plaintext
Symmetric Cryptosystems

Encryption Key

Plaintext

Encryption Algorithm

Ciphertext

Decryption Key

Decryption Algorithm

Plaintext
Requirements

1. Strong encryption algorithm
2. Secret key known only to sender / receiver
   - Mathematically:
     \[ Ciphertext = E(K, Plaintext) = E_K(Plaintext) \]
     \[ Plaintext = D(K, Ciphertext) = D_K(Ciphertext) \]
3. Assume encryption algorithm is known!
4. Assume a secure channel to distribute key
In Real Life

C = E(K, M)

1. Alice
   - K - secret
   - C = E(K, M)
   - Cannot Decrypt C!

2. Intercept
   - C = E(K, M)

3. Bob
   - K - secret
   - C' = E(K, “Hi, A, B, ‘postpone attack’”)
   - Cannot Produce C’!

M = Hi, A, B, “attack tomorrow”
Cryptanalysis

- Objective:
  - Recover message given ciphertext
  - Recover key – more significant – why?

- General approaches:
  - Cryptanalytic attack
  - Brute-force attack

- If either succeed all key use compromised
Cryptanalytic Attack

- Ciphertext only
  - Only know algorithm & ciphertext
  - Assume you know or can identify plaintext

- Known plaintext
  - Know/suspect plaintext & ciphertext
  - *How?*
  - Perhaps know some protocol
Cryptanalytic Attack (cont’d)

- **Chosen plaintext**
  - Select plaintext and obtain ciphertext
  - Example: Encrypt 0
  - Encrypt 1
  - Encrypt 01, 10, etc

- **Chosen ciphertext**
  - Select ciphertext and obtain plaintext

- **Chosen text**
  - Select plaintext or ciphertext to en/decrypt
Cryptanalysis – Brute Force

- Always possible to simply try every key
- Work is a function of key size
- Given ciphertext
  1. Try every key until decryption is intelligible
  2. Assume either know / recognise plaintext
- Use cloud computing for parallelism
- Use volunteer computing
  - SETI@Home, Folding@Home, Large prime search, RSA challenge
# Brute Force Speed

<table>
<thead>
<tr>
<th>Key Size (bits)</th>
<th>Number of Alternative Keys</th>
<th>Time required at 1 decryption/µs</th>
<th>Time required at 10^6 decryptions/µs</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>(2^{32} = 4.3 \times 10^9)</td>
<td>2^{31} (\mu s) = 35.8 minutes</td>
<td>2.15 milliseconds</td>
</tr>
<tr>
<td>56</td>
<td>(2^{56} = 7.2 \times 10^{16})</td>
<td>2^{55} (\mu s) = 1142 years</td>
<td>10.01 hours</td>
</tr>
<tr>
<td>128</td>
<td>(2^{128} = 3.4 \times 10^{38})</td>
<td>2^{127} (\mu s) = 5.4 \times 10^{24} years</td>
<td>5.4 \times 10^{18} years</td>
</tr>
<tr>
<td>168</td>
<td>(2^{168} = 3.7 \times 10^{50})</td>
<td>2^{167} (\mu s) = 5.9 \times 10^{36} years</td>
<td>5.9 \times 10^{30} years</td>
</tr>
<tr>
<td>26 characters (permutation)</td>
<td>(26! = 4 \times 10^{26})</td>
<td>(2 \times 10^{26} \mu s) = 6.4 \times 10^{12} years</td>
<td>6.4 \times 10^{6} years</td>
</tr>
</tbody>
</table>

- **Key size is essential!**
- **Moore’s law:**
  - As computers get faster, old ciphertexts become vulnerable
Public Key Cryptosystems (PKC)

- Most significant advance in the 3000 year history of cryptography!
  - Uses two keys – a public and a private key
  - Asymmetric: parties are not equal

- Public invention
  - Whitfield Diffie & Martin Hellman at Stanford University in 1976
  - Known earlier in classified community
Why Public Key?

- Addresses two key issues:
  - **Key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **Digital signatures** – how to verify a message comes intact from the claimed sender
PKC in a Nutshell

Encryption Key

Plaintext

Encryption Algorithm

Decryption Key

Decryption Algorithm

Plaintext
PKC in Real Life!

1. $C = \text{Encrypt}(\text{pubKey}_B, M)$

2. Intercept $C$

Bob

pubKey$_B$ - public

privKey$_B$ - private

$M = \text{Decrypt}(\text{privKey}_B, C)$

Malory

Cannot Infer privKey$_B$ from pubKey$_B$ !

Cannot Obtain M !

Alice

Has message $M$
More Definitions

- **Unconditional security**
  - No matter how much computer power or time is available, the cipher cannot be broken
  - The ciphertext provides insufficient information to uniquely determine the corresponding plaintext

- **Computational security**
  - Given limited computing resources (e.g., time needed for calculations is greater than age of universe), the cipher cannot be broken
Crash Course in Crypto

- Building blocks that we will define and use in the next lectures
- Will describe in more detail in class 5, 6, 7
- Encryption
- Signature
- Hash
Digital Signatures

- Verify author, date & time of message
- Authenticate message contents
- Verifiable by third parties to resolve disputes
Digital Signature Model

Plaintext $\rightarrow$ Private Key $\rightarrow$ Signature Algorithm $\rightarrow$ Signature $\rightarrow$ Verification Algorithm $\rightarrow$ Valid!

Plaintext $\leftarrow$ Public Key $\leftarrow$ Verification Algorithm $\leftarrow$ Invalid!
In Real Life

Alice

Bob

Malory

pubKey_B - public
privKey_B - private

M = I owe Alice $1000
S = Sign(M, privKey_B)

Verify(M, S, pubKey_B) = true!

M' = I owe Malory $1000 for same S

Bob's Signature

M' = I owe Malory $1000 for new S'

S = Sign(M, privKey_B)

Intercept

2
In Real Life

Alice

Bob

pubKey\(_B\) - public

privKey\(_B\) - private

\(M = \text{I owe Alice }$1000\)

\(S = \text{Sign}(M, \text{privKey}_B)\)

\(\text{Verify}(M, S, \text{pubKey}_B) = \text{true}!\)

Bob Cannot Deny Signature \(S\)!
Digital Signature Requirements

- Depend on the message signed
- Use information unique to sender
  - Prevent both forgery and denial
- Easy to generate
- Easy to verify
- Computationally infeasible to forge
  - New message for existing digital signature
  - Fraudulent digital signature for given message
Crash Course in Crypto

- Building blocks that we will define and use in the next lectures
- Will describe in more detail in class 5, 6, 7
- Encryption
- Signature
- Hash
Hash Functions

- Condenses message $M$ to fixed size
  - $h = H(M)$

- Assume hash function is public
- Used to detect changes to message

Looks Random!
Hash Properties

- **Pre-image resistance:**
  - Given value $h$, hard to find message $M$ such that $h = H(M)$

- **Second pre-image resistance:**
  - Given message $M_1$, hard to find $M_2$ such that $H(M_1) = H(M_2)$

- **Collision resistance:**
  - Hard to find any $M_1$ and $M_2$ such that $H(M_1) = H(M_2)$
Crash Course in Crypto

- Building blocks that we will define and use in the next lectures
- Will describe in more detail in class 5, 6, 7
- Encryption
- Signature
- Hash
Outline

- Administrative Issues
- Rules of the Class
- Class Overview
- Information Assurance Overview