Midterm Review

ECE 422/CS 461

Early Informal Feedback

Help us make discussions better

Midterm Date/Format

- Location & Time
  - October 16 - Monday
  - ECEB 1002
  - 7 - 9pm
- Conflict
  - October 13 - Friday
  - 4 - 6pm
  - **MUST** have ***already*** emailed Ryan

- ~50% Lecture content
  - Multiple Choice
  - Short answer
- ~50% MPs 1 & 2
  - Short answer
  - Long answer
- **Note**: *Can be more than one answer to MC questions*

Midterm Content

- **Lecture Content**
  - Authentication
  - Message integrity
  - Pseudorandomness
  - Symmetrical/Asymmetrical Cryptography
  - TLS
  - Key Management
  - Buffer Overflows and Beyond
  - Malware
  - Access Control
  - Isolation
  - and more ...
- **Cryptography MP problems**
  - Weak Hashing Algorithm
  - Factoring P's/Q's
  - Length Extension
  - Padding Oracle
  - and more ...
- **Systems MP problems**
  - Buffer overflow
  - Pointer Manipulation
  - Integer overflow
  - ROP
  - Format String Attack
  - and more ...

Note: These slides are not a comprehensive review of all materials

Security Mindset

Laws/Disclosure models

- CIA triad
  - Confidentially
  - Integrity
  - Availability
- CFAA
  - Illegal access
- DMCA
  - Circumvent protection
- ECPA
  - Restrictions on wiretaps of electronic messages (transmitted/stored)
- FERPA/HIPAA
  - Access of educational/medical records
- Non-disclosure
  - Keep vulnerability secret
- Full Disclosure
  - Distribute knowledge to everyone
- Limited Disclosure
  - Privately disclose to vendor so a patch can be developed

Cryptography Unit

Don’t rely on a secret function

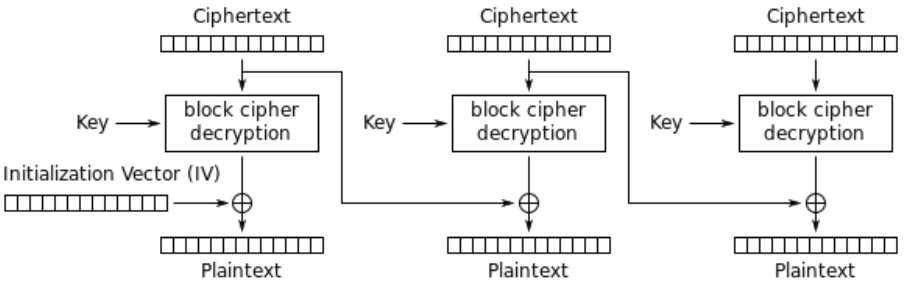
Authentication

- Binds **identity** to a subject
- Passwords
  - Attacks
  - Uniformly distributed
  - Longer → larger brute force search space
  - Store hashes
- Token-based
  - Something user ***has***
  - Static memory cards
    - Credit Card
  - Smart card
    - **OTP**
  - Challenge Response
- Biometric Authentication
  - Something user ***is*** or ***does***
  - Biological features
    - Facial recognition
  - Approximate matching
  - Replay Attack/Can be forced to give away (fingerprints)
- Multifactor Authentication (**2FA**)
  - Combination of previous measures

Hashes

- Hash Function Properties
  - **1st Pre-image**
    - Given h(m), find m
  - **2nd Pre-image**
    - Given m1, find h(m1) = h(m2)
  - **Collision**
    - h(m1) = h(m2)
- Avalanche Effect
  - Flipping one bit of input causes all output bits to change w/ 50% probability
- Confusion
  - Output changes on several parts based on input
- Diffusion
  - Changing a character changes many characters in output (vice versa)

- Attacks
  - Pigeonhole principle
    - Input space > output space
  - Birthday Attack
    - Table of entries
  - Cycle Finding
    - “Tortoise and hare”



Cipher Block Chaining (CBC) mode decryption

Integrity

- **Problem:** Send a message over an untrusted channel without being changed
- **Provably Secure:** Random Function
- **Practical:** Pseudorandom function
- **Real world use:** HMACs

Hash-based Message Authentication Code

= Hash(k ⊕ opad || Hash(k ⊕ ipad || m))

k = secret key m = message

opad = 0x5c5c... ipad = 0x3636...

Confidentiality

- **Problem:** Sending a message in the presence of an eavesdropper
- **Provably Secure:** OTP
  - *Never* reuse any part of the pad
- **Practical:** Pseudorandom Generator, Block Cipher
- **Real world:** Stream ciphers, AES

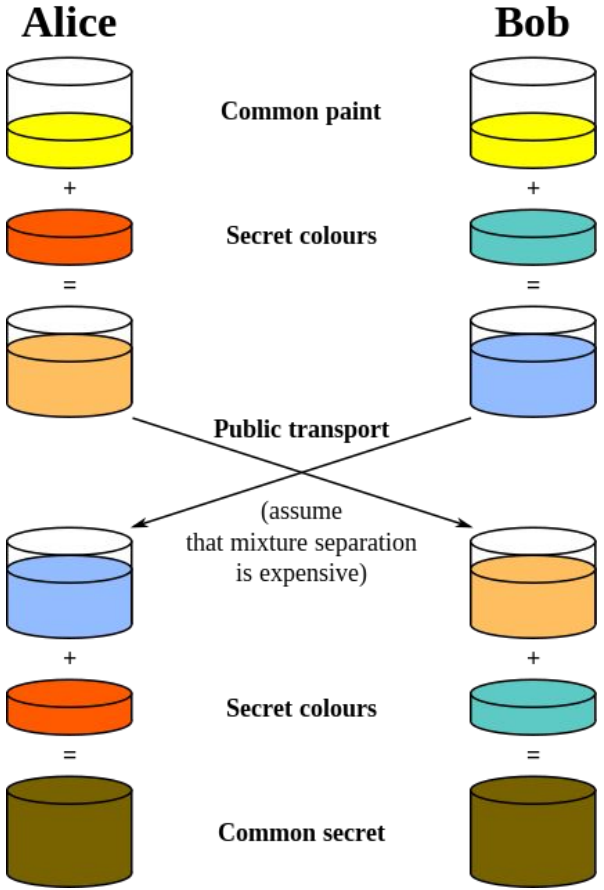
PRF vs. PRG

PRF = arbitrary length input ➔ fixed length

PRG = fixed k (seed) ➔ arbitrary length stream

Key Exchange

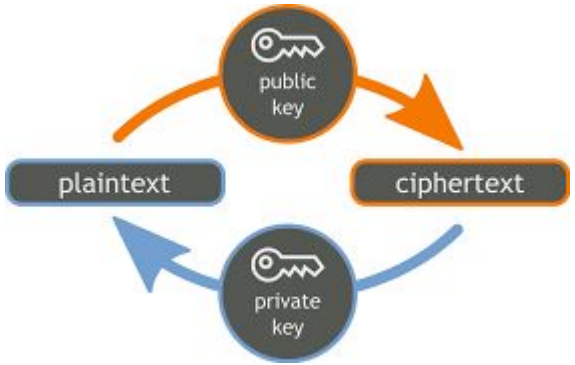
- **Diffie-Hellman Protocol**
  - Agree on public parameter
  - Alice generates random secret a, send g^a
  - Bob generates random secret b, send g^b
  - Computes (g^b)^a == (g^a)^b, shared secret
- Safe against **passive** eavesdropping attacks
- Vulnerable to **Man-in-the-middle attacks**
  - Out of band communications
  - Physical contact
  - DH w/ user authentication



Public Key Encryption

- **Key Generation:** Generate key pair (public, private)
- **Encrypt:** Anyone can encrypt with public key - C = Enc(public, M)
- **Decrypt:** Needs private key to decrypt - M = Enc(priv, C)
- **Security:** Infeasible to guess M or private key even knowing ciphertexts and public key

- Can be used for digital signatures
- Sign with private key, anyone can verify with public key
- **“Unforgeable”** - computationally infeasible to guess S or the private key



RSA

Key Generation

- Pick large random primes **p**, **q**
- Compute **N** = **p** * **q**
- Pick **e** to be relatively prime to Φ(**N**)
- Find **d** s.t. **ed** mod (**p**-1)(**q**-1) == 1

Public key is (**e, N**)
Private key is (**d, N**)

Encryption

$$c = m^e (mod\ n)$$

Decryption

$$m = c^d (mod\ n)$$

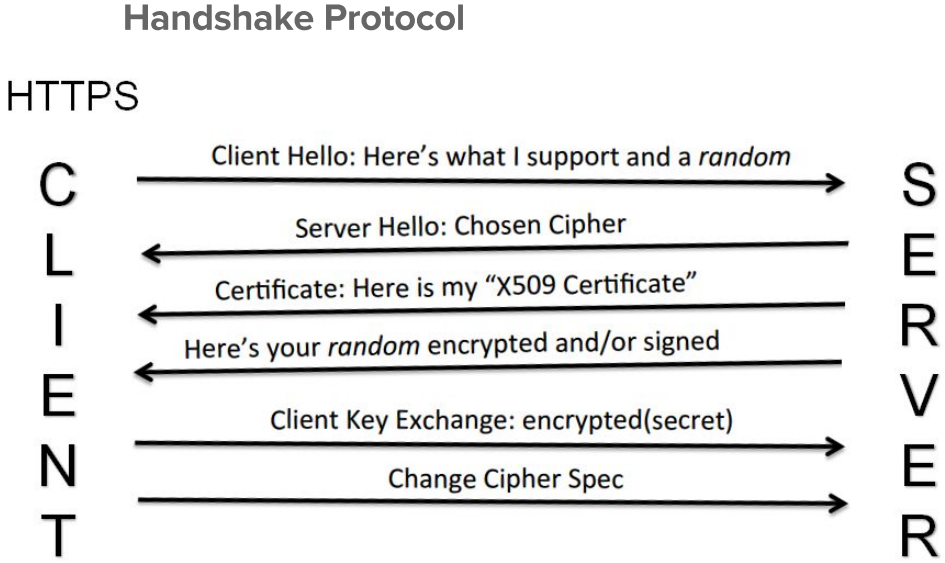
- **Confidentiality**
  - **Encrypt** with public key
  - **Decrypt** with private key
- **Integrity**
  - Sign (**encrypt**) with private key
  - Verify (**decrypt**) with public key
- Implementation details can lead to compromise
  - Common **p**,**q**'s

Key Management

- Hardest part of crypto is **key management**
- Each key should only have one purpose
- Vulnerability increases:
  - The more you use it
  - The more places you store it
  - The longer you use it
- **Forward Secrecy**: Learning old key shouldn't help adversary learn new keys
- **Secure Channel**
  - Confidentiality and integrity
  - Encrypt then MAC
  - Use separate keys

TLS

- Almost all encrypted protocols use SSL for transport encryption
- Lives in between the HTTP (application) and TCP (transport) layer
- Certificates are issued by Certificate Authorities and are used by the browser to verify identity and trust
  - “Chain of trust”
- Cipher Suite
  - Key Exchange Algorithm (ECDHE_RSA)
  - Authentication Algorithm (ECDSA)
  - Bulk Encryption (AES_256_CBC)
  - MAC (SHA384)
  - Example:
    - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

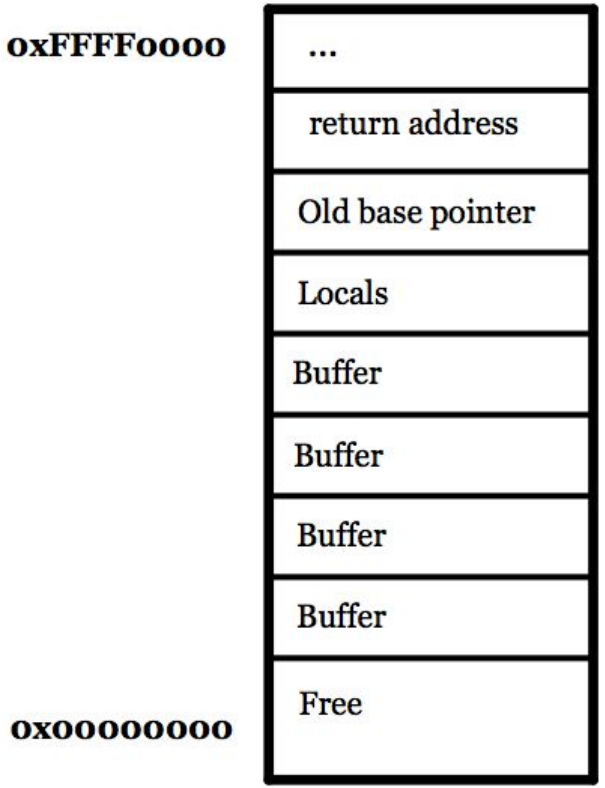


Systems Unit

Smashing the Stack for Fun and Profit

Buffer Overflow

- Attack consists of overflowing the buffer to write arbitrary values to memory and variables
  - Little Endian
- Once the **EIP** is overwritten, whole control of the program is gained
- Main goal is usually to run a **root** shell
- **Canaries**
  - Random value between return address and local variables, check if same
- **DEP**
  - Instructions on the stack are not allowed to be executed
- **ASLR**
  - Randomize addresses of each memory region



Beyond Buffer Overflows

- Return Oriented Programming
- Integer Overflow
- Heap Overflow
- Format String Exploits

Malware & Attacks

- Program inserted into a system with the intent of compromising the CIA of the victim's data
- **Insider attack**
  - Security breach caused by someone part of the org
- **Backdoor**
  - Hidden feature or command that allows users to perform commands not normally allowed
- **Trojan Horses**
  - Software designed to desirable function but is designed to perform malicious functions

Virus

- Code that propagates across systems by arranging to execute itself creating new instances
  - Infects by altering stored code
  - Typically w/ help of **user assistance**

Worms

- Code that self-propagates across by arranging itself to immediately execute
  - Generally infects by altering running code
  - **No user intervention required**

Payloads

- Information theft
- Stealthing
  - Backdoor
  - Rootkits
- Adware
- Ransomware
- Droppers
- Keylogging / Password Stealing

Botnets

- Collection of compromised machines under control of an attacker
- “Phones home” to a C&C server

APT

- Advanced, custom malware targeted at specific systems
- Usually incorporate zero-days and are persistent
  - Example: Stuxnet

Access Control

- Collection of methods/components that support CIA
- Only allow authorized subjects to access permitted objects
- Mandatory Access Control
  - Don’t allow users to define permissions
- Discretionary Access control
  - Users given ability to determine the permissions governing access to their files
- Role based access control
  - Users are assigned roles
  - Roles have permissions
- Attribute-based Access Control
  - Users/Objects given attributes
  - Decisions made based on attributes of both
- Security Policy
  - Defining constraints and rules so that an asset is secure
- Unix permissions
  - Each file owned by user and group
    - Rwx
- Data Leakage by lost devices
  - Lost laptops
  - USB sticks
- Device to Data - Encryption
- Password policies

Isolation

- Confinement
  - Ensure misbehaving app cannot harm the rest of the system
  - Hardware, VMs
- Reference Monitor
  - Mediates requests
  - Must always be invoked
  - Tamperproof

- Virtual Machines
  - Run apps in their own OS on top of Host OS
  - Sample Checks
  - Covert Channels
    - Unintended communication channel between isolated components
    - Can leak classified data

Malware Prevention

- Intrusion Prevention Systems (**IPS**)
  - Prevent Attack before reaches system
- Intrusion Detection Systems (**IDS**)
  - Detect attack after it has happened
  - **HIDS**/**NIDS** - Host/Network based **IDS**
- **Anomaly** based
  - Collect data on legit users
  - Flag behaviour that is weird
- **Signature**/**Heuristic** based
  - Examine known attack patterns
  - Develop signatures that match to those patterns
- **Consistency**
  - % two binaries are classified the same by different AV
- **Completeness**
  - % of malware samples detected

- **Encrypted Virus**
  - Decryption engine + encrypted payload
  - **Detection**: Look for decryption engine
- **Polymorphic Virus**
  - Encrypted virus w/ random variations of the decryption engine
  - **Detection**: Use CPU emulator
- **Metamorphic Virus**
  - Different virus bodies
  - Code permutation, instruction replacement
  - **Detection**: Challenging to detect

Machine Problems

Did ya miss us?

Cryptography

- Basic Ciphers
  - Caesar
  - Substitution
  - etc.
- Weak Hashing Algorithm
- Length Extensions
- MD5 Collisions
- Padding Oracle
- Common P,Q’s
- Colliding Certs

Systems

- Stack layout and registers
- C Calling Conventions
- Shellcode
- Buffer overflow
  - Variables
  - Pointers
  - EIP
- Integer overflow
- Linked List
- System
- ROP
- Callback Shell
- Format String Attack