EECS 565 Cheat Sheet Exam 1

Week 1:

- What is **Security**?
- The **CIA** notion: Confidentiality, Integrity, Availability
- **Confidentiality**
  - Student grade information is an asset whose confidentiality is considered to be very high
  - Student enrollment information: may have moderate confidentiality rating; less damage if enclosed
- **Integrity**
  - A hospital patient's allergy information: high integrity data. a doctor should be able to trust that the info is correct and current
  - An online newsgroup registration data: moderate level of integrity
- **Availability**
  - A system for authentication: high availability requirement. If customers cannot access resources, the loss of services could result in financial loss
  - A public website for a university: a moderate available requirement; not critical but causes embarrassment
- **Threat vs. Vulnerability**
  - **Vulnerability**: security weakness that might be exploited to cause undesired consequences.
  - **Threat**: a set of circumstances that potentially cause loss or harm.
  - **Attack**: the exploitation of vulnerabilities by threats.
  - Threat is blocked by a control of a vulnerability
  - Zero-day vulnerability - not patched
- **MOM Attack**
  - Method: skills, knowledge, tools to pull off the attack
  - Opportunity: time and access
  - Motive: expected gains
  - Eliminate one of them
- Principle of Easiest Pen
- Not whr strongest defenses are
- Methods of Defense
- **Prevent**–close vulnerability
- **Deter**-make attacks more difficult
- **Deflect** - make another target attractive
- **Mitigate** - make attack's impact less severe
- **Detect**-know when attack occurs
- **Recover** - mitigate attack's effects
- Security Life Cycle: **1.**Planning **2.**Implementation **3.** Monitor&Manage **4.** Intrusion Detection **5.**Security Assessment **6.**Threat/Risk Analysis **7.**Security Policy Creation
- Principle of Effectiveness-used & properly to be effective
- Principle of Weakest Link-no stronger than weakest link

Why RSA Works?

- **Euler's totient theorem**: $m^{\varphi(n)} \equiv 1 \bmod n$
  - $\varphi(n)$ is the number of integers that are less than or equal to n and relatively prime to n.
    - if n is prime, $\varphi(n) = n - 1$
    - if $n = p \times q$, $\varphi(n) = (p-1)(q-1)$
- Now, let's take a look at RSA decryption.
  - $ed \equiv 1 \bmod \varphi(n)$, so $ed = k \times \varphi(n) + 1$
  - $c^d \bmod n = (m^e)^d \bmod n = m^{ed} \bmod n$

$$= m^{k\varphi(n)+1} \bmod n = m \times m^{k\varphi(n)} \bmod n$$
$$= [m \times (m^{\varphi(n)})^k] \bmod n = (m \times 1^k) \bmod n = m$$

Why is RSA secure?

- **Factoring problem**: Given a large positive integer n, find primes $p_1, \ldots, p_k$ such that $n = p_1^{e1} \ldots p_k^{ek}$.
  - This is considered a hard problem that we don't have a solution for centuries.
- **RSA problem**: Given c, $n = pq$, and e, find m such that $m^e = c \bmod n$.
  - This is to find the e^{th} root of c modulo n. It can be solved if n can be factorized.
  - There is no known efficient algorithm for this, without knowing p and q.
- **Finding RSA private key**: Given (e, n), find d without computing $\varphi(n)$.
  - Considered at least as difficult as factoring n.

Takeaway: we should NOT use the same n among multiple users.

Week 2:

- **Cryptography**: conceals against unauthorized access. (Encipher, Digital Sig, Auth. Exchange)
- **Cryptosystem**: system for encryption and decryption. Algorithm, All P-text & C-text, keys
- **Ceaser** cipher: char replaced 3 slots to right A →D (Monoalphabet Sub)
- **Shift cipher**: $E_K = (m+K)\%26$ $D_K = (c-k)\%26$ K = 1-25 (A→P M→ A)
- **Mono cipher:** sub 1 character with another (26! possibilities)
- **Frequency Analysis:** correlates to statistical patterns in language
- **Cryptanalysis:** study of methods for obtaining meaning of encrypted info. "hacking" finds the weakness
- **Cryptology**=Crypto(graphy+analysis)
- **Polyalphabetic** → multi alphabet
- **Homophonic** → multiple possible output for an input
- **Polygram** → encipher groups of letters at once.
- **Vigenere:** table of different shifts for each row. **K =** {5,19,7,11,21} char1 = row 5, char2 = row 19
- **K=** superbowl **P=**rockchalkjayhawk
- **Cipher Text =** jirotiohvbunlrxy
- **Vernam Cipher**=one-time pad OTP Key is the same length of plaintext **Ciphertext** = plaintext + key % 26
- **OTP** = bit-level XOR "modulo 2"
- (k XOR k = 0) (p XOR k XOR k = p)
- **Plus:** In theory OTP is impossible to crack as long as key is truly random
- **Problems:** Key = length of p-text. Insecure if key is reused. Doesnt guarantee integrity only confidentiality. (Attacker can change cipher text)
- **Sub Cipher:** 1 set of bits for another
- **Transposition Cipher:** rearranged order of cipher to break repeats. ex: p=ROCK CHAL K c= OKRC HLCA K key = [1,2,3,4] → [2,4,1,3] d = 4
- Can use common pairs to figure d.
- **Substitution adds confusion**. Makes relationship b/w p-text and c-text as complex as possible.
- **Transposition adds diffusion.** dissipate the statistical structure of p-text. (1 change in p-text → multiple changes in c-text)
- **Claude Shannon** ("father of information theory") "perfect secrecy if as many possible keys as p-text and each key is equally likely
- P(guessing p-text | know cipher) is equal to P(guessing p-text)
- **S-P Network**/Sub and Permutation
- **Good crpytosystem: 1.** enumerate all pos keys. **2.** find key from reasonable amount of c-text and p-text by enumerating possible keys. **3.** Produce p-text from c-text w/o key. **4.** Distinguish c-text from random values.
- **Kerckhoffs'** - security should only depend on secrecy of private key.
- **Stream cipher:** one symbol at a time. i^{th} symbol → i^{th} part of keystream. **Adv.** (Fast, Less Code, Low Error Prop) **Dis.** (Low diff & vulnerable to insert + modif)
- **Block cipher:** encrypt each block (DES,3DES,AES) **Adv.** (High Diff, +imune 2 insertion) **Dis.** (Slow High Error Prop)
- **Keyed Permutation:** N-bit input = 2^N! possible permutation. 2^K possible keys.
- **Symmetric Block Cipher:** A seq. of rounds. Substitutions and Permutations controlled by key in each round.

Week 3:

Block Ciphers:

- **DES: Data Encryption Standard**
  - Symmetric Encryption (KES Attack and Multiple Encryption): Block ciphers and stream ciphers
  - Minor version of Feistel structure
  - Symmetric (Receiver uses same key to encrypt and decrypt)
  - Uses basic steps: substitutions and permutations
  - Effecient hardwar implementation
  - Product cipher: realizes 2^k possible transformations for k-bit key and m-bit block
    - Divides m into 2 halves: m=2w
    - n rounds: n subkeys
    - For each round: $L_i=R_{i-1}$ and $R_i=L_{i-1}\oplus$**F**(R_{i-1},K_i) for F is a round function
  - XOR is reversible
  - Round function F can be anything
  - **DES Basics: 64 bit block size**
  - **56 bit key** (Effective length because every 8th it is used for parity)
  - 2^{56} possible keys
  - 16 rounds (16times/block)
  - Suffers from key exhaustive search attack
  - S-Boxes (Substitution): Shrink R_i from 48 to 32 bits
  - 6bit input, 4 bit utput
  - **Key Exhaustive Search (KES)**
  - For a public-known encryption algo E, E: $\{0,1\}^k x\{0,1\}^l$->$\{0,1\}^l$
  - Attacker tries all possible keys
  - Probability = $1/(2^k)$
  - Average num of attempts (how secure): 2^{k-1}
- **Definitions of security (Cryptanalysis)**
  - Unconditional secure: it is unbreakable no matter how hard attacker tries (OTP)
  - Computational secure: Cost of breaking cipher exceeds data
- **3DES:Triple DES**
  - Encrypt plaintext 3 times (DES+inverse DES+DES)
  - Uses 2 or 3 keys (But depreciating)
  - Doubles key length (112bits)
  - Very slow (3x)
  - Double DES encrypts twice with two different keys
- **Meet-in-the-middle attack**
  - K1 and K2 are unknown but the middle value is the same in the encryption/decryption phases
  - Running two parallel exhaustive searches cracks cipher
  - If cracker has Plaintext/Cipher pairs, guesses all possible K1
  - Encrypt P1 with all possible K1 and record middle value
  - Guess all possible K2
  - Decrypt C1 with all K2 and record middle values
  - Look for collision in middle values
  - Needs two encryption/decryption operations
  - Security strength of 2DES is 2^{56}
- **Cryptanalysis** is process to find the weakness in cryptographic algo
- Ciphertext only - frequency
- Known-plaintext - KES, MITM
- Chosen plaintext (CPA): can obtain cipher for any plaintext
- Chosen ciphertext (CCA): Can decrypt and cipher except target

Week 4:

Public Key Cryptography:

In symmetric systems: n(n-1)/2 keys are needed for n users.

In public key cryptography (PKC): produce two mathematically related keys, share public key & hide private.

1. Primary use of **PKC** is key management (key distribution). S = E(pk_{BOB},K) → K = D(sk_{BOB},S) Encrypt K w/ Bob Public. Decrypt w/ private to get secret Key.

2. Can also be used for **key agreement**, when 2+ user negotiate secret key.

3. PKC used to build the PKI (digsignatur)

4. Provides integrity and authentication

Encrypt M into S with public key, decrypt S into M with private.

RSA is most popular public key method.

Public key: (e,n) Private key: (d,n)

For plaintext message m and ciphertext c, where 0<m<n

Encryption: $c = m^e \bmod n$

Decryption: $m = c^d \bmod n$

Signing: $s = m^d \bmod n$

Verification: $m = s^e \bmod n$

Don't use textbook RSA, small plaintext & easy for attackers to try all possible messages. Output is deterministic

Week 5:

Diffie-Hellman Key Agreement (1976): Enables negotiation of a secret over an insecure media.

P is a large prime, g is a generator: $Z^*_p (2 \le g \le p-2)$

$1 \le a, b \le p-2$

One person ends $g^a \bmod p$, the other sends $g^b \bmod p. \rightarrow k = g^{ab} \bmod p.$

Why is D-H **secure?**:

Discrete logarithm problem, computational D-H problem, Decisional D-H problem.

D-H is secure against **passive** attackers (interception. But it's not secure against **active** attackers (modification).

Vulnerable to **man-in-the-middle attacks.**

How to fix? Use published **DH** numbers and authenticated **DH** exchange.

Elliptic-Curve Cryptography (ECC):

More efficient than DSA- more used

$y^2 = x^3 + ax + b$

Any non-vertical line passes through at most three points on curve. Given any two points, P and Q, we can find the third point R.

ECC achieves the same security strength as **RSA** with smaller keys.

Hash functions: Messages of variant length -> hash of fixed size.

Hash function is a **lossy compression function.**

One-way, collision resistance, weak collision resistance.

Collision Resistance: It should be hard to find $x \ne x'$ such that $h(x) = h(x')$

Birthday paradox means **brute-force collision search is only** $O(2^{m/2})$**.** For example, 160-bit hash, a collision can happen after selected 2^{80} random values instead of 2^{159}.

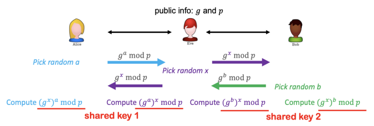
One way-ness **does not** imply collision resistance.

Collision resistance **does not** imply one-wayness.

Weak collision resistance **does not** imply collision resistance.

- DES Basics: 64 bit block size
- **56 bit key** (Effective length because every 8th it is used for parity)
- Each half: circular left shift
  - Shift 1 bit in round 1, 2, 9, 16,
  - Shift 2 bits in other rounds
- 2^{56} possible keys
- 16 rounds (16times/block)
- Suffers from key exhaustive search attack

Man-In-The-Middle Attacks



Digital Signatures

1. Key generation: (pk, sk)
2. Sign: $\sigma = sign_{sk}(m)$
3. Verify: verify$_{pk}(m, \sigma)$, if valid, accept sig.

Digital Signature Algorithm (DSA)
Public info: $p, q, g \in Z_p^*$: p, q-prime:g-gen
Private key: x, Public key: y=g^xmodp
Birthday Paradox: k ppl k diff birthdays
C(365,k)=365!/(365-k)!*365^k
2 ppl/same bday: Pr(n, k) = 1 – Q 365, k
Common Hash: SHA-256,512,224,384
Message Authentication Code(MAC)
Secure: need key to generate/verify MAC.
(MAC key ≠ encryption key)
HMAC: hmac(k,M)=h(k_2||h(k_1||M))
Message M % into blocks, pad last block
k-secret key-pad k: b-bit K+: pad 0 to left
ipad=36=00110110|opad=5C=01011100

EXAM REVIEW
Introduction to Computer Security
■ Security Concepts
– Security objectives: confidentiality, integrity, availability
■ other objectives besides CIA
■ given a scenario, identify: What needs to be protected? What do we protect for?
■ downloading software with SHA-2 hash
– Threats, vulnerabilities, attacks, and controls
■ threats vs. Vulnerabilities
■ threats can be interception, interruption, modification, fabrication
■ vulnerabilities: hardware,software,data
■ defense controls: prevent, deter, deflect, detect, recover
■ MOM: method, opportunity, motive
■ Security principles
– Principle of Adequate Protection
– Principle of Easiest Penetration
– Principle of Weakest link
– Principle of Effectiveness
Cryptography
■ Terminology and Concepts
– S: sender (Alice); R: recipient (Bob); O: outsider or intruder
– Chuck; Eve: eavesdropper; Mallory: malicious attacker
– Cryptographic algorithms
– Key, plaintext, and ciphertext
– Cryptosystems
– Mathematical representation of cryptosystems
■ Terminology and Concepts
– Cryptology: Cryptography + Cryptanalysis
■ Kerckhoffs's principle
– Unconditional vs. computational secure
– Shannon Secrecy
■ probability models (concepts)
■ confusion vs. diffusion
■ Secret-key cryptography
■ Public-key cryptography
■ Cryptographic hash functions

Modes of operation

- Electronic Codebook (ECB)
- Cipher Block Chaining (CBC)
- Output Feedback (OFB)
- Cipher Feedback (CFB)
- Counter Mode (CTR)

ECB: Divide poaintext into blocks and encrypt each block with the key, then concatenate the output of each block

- Identical blocks of plaintext will make identical blocks of ciphertext (**Information leakage**)
- No integrity checks of blacks and its order, so blocks can be re-ordered or inserted (**Ciphertet manipulation)**
- Error in plaintext results in only one cipher block error (**Error propagation**)
- **CBC:**Plaintext of block i is XOR'ed with ciphertext of block (i-1) before encryption
- Identical blocks of plaintext are encrypted differently
- No information leakage
- Ciphertext manipulation difficult
- Parallel processing: No in encryption yes in decryption
- Cipher text error propagation yes (only in 2 blocks)
- **CTR:** Use a counter that is equal to the plaintext block size
- Identical blocks of plaintext will be encrypted differently
- No information leakage, difficult cipheretext manipulation, parallel processing, no error propagation
- **CFB:** Ciphertext of previous block is feedback to the current block (shift register)
- Identical blocks are encrypted differently
- Block manipulation is difficult
- No parallel processing in encryption but yes in decryption
- Error DOES propagate
- **OFB:** Shift output of previous block feedback to current block
- No error propagation
- **CTR, CFB, and OFB** can convert **block** cipher to a **stream** cipher

---

EXAM REVIEW PART 2
Classic Ciphers
■ Cipher algorithms
– Shift cipher, Caesar cipher:
– Substitution ciphers, monoalphabetic ciphers: just concepts
– Polyalphabetic ciphers: Vigenère Cipher
– One-time pad
■ Need to know:
– encryption & decryption schemes: practiced in MP1 and homework
– Key weaknesses in security: small key space, static statistical patterns
Cryptanalysis
– Brute force: key exhaustive search attacks
■ How many attempts are need to crack a cipher, on average or in worst cases?
■ What determines the efficiency of the brute force attack?
– Frequency analysis
■ Why it works in cracking substitution and transposition ciphers?
■ You practice the technique in homework (cracking using FA is not required in exam)
– Special attacks
■ Meet-in-the-middle/Man-in-the-middle
– Concepts: ciphertext-only, known-plaintext, CPA, CCA

AES: Advanced Encryption Cipher
DES: broken, 3DES: slow
Private-key, symmetric block cipher
128bit data block, 128, 192, and 256 bit keys
Strong, faster than 3DES
Secure for next 50-100 years
Not a **Feistel** cipher: each round operates on all bits instead of halves (2 rounds is full diffusion)
10, 12, or 14 rounds (128, 192, 256 bit)
3DES: 48 rounds
Rounds take 4 operations:
1. SubBytes: non-linear byte substitution
2. Shiftrows: circular byte shift in each row. 3. Mixcolumns: add diffusion, addroundkey
AES State Array
Keeps a state array of four 4-byte columns
Perform a byte-for-byte substitution
Padding is necessary if message is not multiple of 16 bytes
Key Expansion
Input: 16 bytes
Each round: 4 bytes (10 rounds and one initial XOR: 44 bytes)
AES Decryption
Run cipher in forward direction but use inverse operations
AES Security
Efficient, secure (Strength for 128bit key is 2^{127})
No known successful attacks against full AES
Linear cryptanalysis – generally reduced due to designs to frustrate linear analysis
– Correlate input with output
■ Differential cryptanalysis – generally reduced due to many rounds
– Correlate differences in input with differences in output
■ Should consider new side-channel attacks

EXAM REVIEW CONT. PART 3
Block Ciphers
■ DES and AES
– concepts: block, block size, key length
– concepts: how confusion and diffusion are achieved
■ Feistel, P-box, S-box, byte substitution, shift-row, mix-column, add round key
– comparison: DES, double-DES, 3DES, AES ■key/security length, efficiency
■ Modes of ops: ECB, CBC, CTR, CFB, OFB
– concepts: info leakage, ciphertext manipulation, parallel processing, error propagation
– schemes (not required in exam)
Public-key Crypto
■ RSA
– Schemes: key generation, enc, dec
■ concepts: p, g, n, phi(n), and their properties; selection of e and d, input space (plaintext) and output space (ciphertext)
■ concept: textbook RSA, padding
■ concept: why RSA is secure
■ Apps:– Encryption: key distribution (why)– Signing: digital signatures, PKI
Public-key Crypto
■ Diffie-Hellman key agreement
– concept: why we need D-H for key agreement
– the protocol: scheme, no calculation required
– man-in-the-middle attack and defenses
Public-key Crypto
■ Hash– collision/collision resistance
– calc: brute force to find a hash collision

Homework:

1. **Sub-Cipher:**ciphr="vealruwgwwk"key= "dawn"plain="seeyouattwo"
2. **Poly-Sub:** each letter subbed by another using multi alphabet. Not as vulnerable to freq. analysis
3. **OTP:** Secure as long as the key is used once.
4. **Freq. Analysis:** used to decipher messages by looking at patterns and frequencies of both individual letters and common words.
5. **DES:** Block (64 bit) Key (56 bit) You cannot use diff keys to get same plain to cipher. Algorithm is complex, very unlikely to have same
6. **Meet in Middle:** Attacker uses matching plain and cipher to guess every possible key. Possible in 2-DES (2^{56}) Not poss in 3DES (2^{168})
7. **Key Exhaus Attack:** 2^{191}/40 mil number of guesses (half key space)
8. **Error during transmission:** An error in the first block will cause the corresponding and next plaintext to be faulty. B. Just the first recovered plaintext block will have one bit error.
9. **RSA:** p = 3 q = 11 e = 7 p*q = 33 (p-1)(q-1)=20 d = (d*7)%20=1 = 3 Encryption = ([plaintext]^E % 33) Decryption = ([ciphertext]^D % 33) RSA is vulnerable to cipher attacks where key can be guessed by reencrypting cipher to produce same cipher, i.e. you guessed key
10. **Diffie-Hellman:** Vulnerable to MITM attack. Can intercept exchange of p and g%p and change to p' and g'%p'. The attacker can further encryp or decryp. A digital sig. can be used to verify communication is only b/w them.
11. **SHA-256:** Has very strong collision resistance. Irreversible, no expected plaintext after hash is generated. 2^{128} guesses 4 desired collision prob > 50%
12. **HMAC:** Message authen based on hash. Combination w/ secret key very hard to crack. Ex. Authenticating API req.

- Which of the following is NOT a block cipher: **A) Vignere**
- Which factor is most crucial in calculating comp time for Poly Sub Cipher? **D) Key space size**
- In **key exhaustive attack**, need to search half key space to crack key $\mathbf{2^{55}}$
- **SHA-512** can be used in HMAC to gen the message auth code of 512 bit