CS161 MT2 Cheat Sheet

Cryptography Main Goals:

- **Confidentiality:** make sure data is private
- **Integrity:** make sure message sent isn't altered (regardless of confidentiality)
- **Authentication:** determine who sent the message (generally implies/requires integrity)

Kerckhoff's Principle:

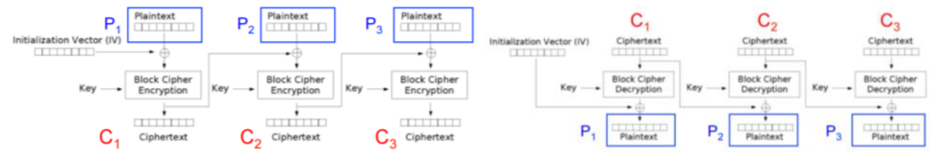
1. Cryptosystems should remain secure even when attacker knows all internal details (don't reply on security-by-obscurity)
2. Private keys are the only thing that must stay secret
3. It should be easy to change keys

One Time Pad:

- **Confidentiality: YES**, **Integrity: NO, Authentication: NO**
- One-Time Pad is provably secure, give pad is random, but impractical. Idea is to use different key for each message M.
- E(M, K) = $M \oplus K$, D(C, K) = $C \oplus K = M \oplus K \oplus K = M \oplus 0 = M$
- Don't ever reuse a K. If so, C = E(M, K), C' = E(M', K), Eve can compute $C \oplus C'$ which is $(M \oplus M') \oplus (K \oplus K) = M \oplus M'$ which is extra info

Block Cipher:

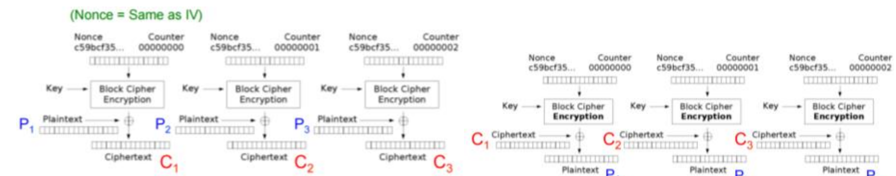
- **Confidentiality: YES**, **Integrity: NO, Authentication: NO**
- Fixed-size, stateless, requires "modes" to securely process longer messages
- Properties:
    1. CORRECTNESS: $E_K(M)$ is Bijective -> invertible
    2. EFFICIENCY: computable in micro seconds
    3. SECURITY: for unknown K, output is random
- Encryption Standards:
    1. DES (Data Encryption Standard) 64 bit block size, 56 bit key
    2. AES (Advanced Encryption Standard) 129 block size, 129, 192 or 256 bit key, not proven secure, but no known flaws
- 2^{10} is approximately 10^3 for brute force 2^{128} is $\sim 10^{39}$, which is hard
- **ECB (Electronic Code Book)** – $C_i = E(P_i, K)$ no confidentiality if len(message) > len(key)
- **CBC (Cipher Block Chaining)** – C = E(Plaintext, K), not parallelizable, widely used, if no reuse of nonce, provably secure, if underlying block cipher is secure, parallelizable alternative is CTR mode



Cipher Block Chaining (CBC) mode encryption

Cipher Block Chaining (CBC) mode decryption

- **CTR (Counter Mode)** – Parallelizable alternative with counter



Counter (CTR) mode encryption

Counter (CTR) mode decryption

Stream Ciphers:

- **Confidentiality: YES**, **Integrity: NO, Authentication: NO**
- Keeps state from processing past message elements, can continually process new elements -> one-time pad on the cheap
- PRNG(seed) returns same random numbers for specific seed
- Encryption: E(M, K) = PRNG(K, IV) $\oplus$ M,
- Decryption: D(C, K) = PRNG(K, IV) $\oplus$ C

Public Key Cryptography:

- **Confidentiality: YES**, **Integrity: NO, Authentication: NO**
- Gen random large primes p, q, compute $n = p*q$, computer $\phi(n) = (p-1)(q-1)$, if Eve sees n, she can't deduce $\phi(n)$, choose an e $2<e<\phi(n)$, where e and $\phi(n)$ are relatively prime (e = 3 is common)
- Public key: $K_E = \{n, e\}$, compute $d = e^{-1} \bmod \phi(n)$, d is multiplicative inverse of e
- Private key $K_D = \{d\}$, hard to find d without knowing $\phi(n)$
- Encryption: $E(M, K_E) = E_{\{n, e\}}(M) = M^e \bmod n$, Decryption: $D(C, K_D) = D_{\{d\}}(C) = C^d \bmod n = (M^e)^d \bmod n = M^{e \cdot d} \bmod n = (M^{e \cdot d-1}) \cdot M \bmod n$
- Vulnerable to dictionary attacks (ex. know it's "buy" or "sell")

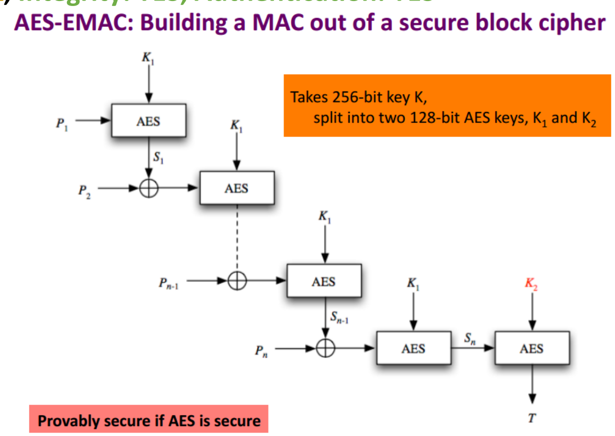
Hash Functions:

- **Confidentiality: NO/NA**, **Integrity: YES**, **Authentication: NO**

- If we hash message with a salt, then it is much less likely to leak information
- Variable input size -> fixed output size, one way, but not one to one
- Properties: **preimage resistant**: intractable to reverse, **second preimage resistant:** intractable to find x' s.t H(x) = H(x'), **Collision resistant**: intractable to find any x, y s.t H(x) = H(y), implies second preimage resistant (SHA-256 is currently not broken)

<u>MACs (Message Authentication Codes) –Symmetric-Key Cryptography:</u>

- **Confidentiality: NO/NA**, **Integrity: YES, Authentication: YES**
- Requirements: 1. Mallory can't send a MAC as Alice (private key must be used – Only Alice has private key), 2. Different messages should not map to the same MAC output (hash like properties)
- HMAC(M, K) = $H[(K^* \oplus \text{Pado}) \| H((K^* \oplus \text{Padi}) \| M)]$
    1. HMAC most widely used, safe even if hash is flawed



<u>Digital Signatures –Public Key Cryptography:</u>

- **Confidentiality: NO/NA**, **Integrity: YES, Authentication: YES**
- Use RSA private key to sign, and public key to verify
- Non-repudiation property: Alice can't deny signing message

<u>Diffie-Hellman Key Exchange:</u>

- Trade symmetric private keys reliably without meeting
    1. Large primes p and g (1 < g < p-1) are sent (Eve can see them)
    2. Alice and Bob generate private a, b between 1 and p - 1
    3. Alice gets g^b mod p and Bob gets g^a mod p, both can compute g^{ab} mod p, and that will be the secure symmetric private key

<u>Trusted Authorities & Trusted Anchors & Digital Certificates:</u>

- A cert is a signed claim about someone's key
- We can trust the cert if we trust the authorities and that the private key wasn't stolen
- Certificate Authorities (CAs) are trusted parties in a PKI
- Revocation: CA screws up; fixes: expiration dates, revocation list and online certificate status protocol (OCSP)

<u>OSI Layers:</u>

- Layer 1 Physical (ex. Ethernet, WiFi, Fiber Optics), sniffing possible
- Layer 2 Link (ex. Ethernet, 802 WiFi), sniffing possible (tap a link)
- Layer 3 Network (ex. IP, localhost)
- Layer 4 Transport (ex. UDP, TCP, SYN)
- Layer 7 Application (ex. SMTP, FTP, POP3, HTTP, DNS, Skype, DHCP)
- **IP Header**: 32 bit IP addresses, payload protocol (4), payload size

<u>Access Points (APs) Connection:</u>

- K = F(HMAC-SHA1, "$secret!", "ATT192", KeyCounter, 4096)
- WPA2 personal -> anyone one network can see traffic
- DHCP Threats: attacker can win race to substitute fake DNS server/gateway by spoofing a response as a "router" DHCP offer

<u>TCP (Transmission Control Protocol) + UDP:</u>

- Reliable delivery (packets are numbered, response is required)
- **TCP Header**: src/dnt port, sequence number, ack number, flag
- http port is usually 80, https is usually 443, RST is abrupt end, no ack
- 3 way handshake required to establish connection:
    - A -> B: SrcA, SrcP, DstA, DstP, SYN, seq = x
    - B -> A: SrcA, SrcP, DstA, DstP, SYN+ACK, seq = y, ack = x + 1
    - A -> B: ScrA, SrcP, DstA, DstP, ACK, seq = x, ack = y + 1
- We are toast if an attacker can see our TCP traffic (port & sequence #)
- Blind Spoofing is possible if attacker can guess our port and sequence #, especially if sequence number is picked on a clock

<u>DNS (Domain Name Servers) + Resolver:</u>

- Question, Answer, Authority, Additional sections in dig command
- Don't accept entries in additional section, unless in Bailiwick
- Identification field is 16 bits, if an attacker can guess, then send fake DNS reply. Karminsky spoofing uses additional field, many lookups
- Fix for blind spoofing? Use random srcP and ID field, for 32 bits