Cheatography

CS440: Intro to cyber security Part I Cheat Sheet

by csthrowaway via cheatography.com/201525/cs/43099/

Week 1: Basics

3 Types of security threats: At home (Client), On the move, On the guest house (Server)	CPU Protection rings (The more outer the rings, the less access to sensitive information)	TCP/IP Internet suite: ATNLP	Defining security: Confidentiality, Integrity, Availability
Confidentiality: Only intended participants can gain access to information.	Integrity: Information not modified by authorised parties.	Availability: You can gain access to information at any time you want.	Kerckoffs Principle: Security of a system lies in its keys only. Everything else should be public knowledge.
Tradeoffs for achieving security: High functionality + Low cost -> Low security	High security -> High cost + Low functionality	Best to have a good enough balance	High security -> Low performance + Low compatibility

Week 2: Symmetric Encryption

Plaintext: Original message/data	Ciphertext: Encoded message/data after encryption	Key: Information used in encryption & decryption	Encryption: Turn plaintext to ciphertext
Decryption: Turn ciphertext back to plaintext	Symmetric Encryption: Shared key is used in encrypting and decrypting data.	Motivation for encryption: Protect confidentiality between two parties. (Only authorised parties can gain access to data)	How does encryption achieve this? Plaintext is turned into ciphertext using substitution and permutation. Adversary cannot find patterns and cannot interpret the meaning of ciphertext.
Classical ciphers: Caesar cipher, Move plaintext by X number (Enc), Move ciphertext back by x number (Dec)	One-time pad ciphers: Ciphertext = Plaintext XOR key. Key must be a random bitstring of same length as plaintext, every enc uses a newly chosen key. Plaintext = Ciphertext XOR key (same key), Limitation: Need to store an unlimited # Keys and key generated must be truly random.	Block ciphers: Message is broken down into blocks. Each block is a fixed number of bits of the message	Symmetric Encryption Enc process: Plaintext -> (Encryption algorithm + secret key) -> Ciphertext
Symmetric Encryption Dec process: Ciphertext -> (Decryption algorithm + secret key) -> Plaintext	Encryption & Decryption algorithm is public but secret key is private.	Security relies on the secret key only.	Chicken & egg problem: To exchange secret data, you must have a shared secret.

Week 2: Symmetric Encryption (cont)

AES algorithm: AES is a symmetric key algorithm, uses fixed size blocks of data (128 bits), If the blocks of data (message) not exactly divisible by 128 bits need to add padding, No specific key sizes (larger key size means more rounds of transformation)	Modes of Encryption: ECB & CBC	ECB Enc Process: Plaintext split into blocks of equal size -> Each block is encrypted using the same secret key -> Encrypted blocks are joined together to form final ciphertext.	ECB Dec Process: Reverse of Enc process. Each block can be decrypted independently using shared secret key.
Characteristics of ECB: Both Enc & Dec can be performed in parallel since each block is independent, ECB is deterministic as same plaintext block will always enc to same ciphertext block with same secret key.	Limitation of ECB: Larger size messages are vulnerable to pattern analysis since they can contain repetitive patterns that could be seen in ciphertext.	Use cases of ECB: Suitable for small amounts of data / Messages that do not contain repetitive patterns.	CBC Enc Process: Message is broken down into blocks -> Block 1 XOR IV -> Ciphertext is Enc with K -> Cipher block 1 -> Block 2 XOR Cipher block 1 -> Ciphertext is Enc with K -> Cipher block 2 -> Block 3 XOR Cipher block 2 -> ... till the last block. Ciphertext will be the result of all the cipherblocks chained together.
CBC Dec Process: Decrypt each block followed by XOR operation with previous cipherblock/ IV for first block.	Characteristics: Chaining (Enc of each block depends on all previous blocks, each erroneous block will lead to wrong Dec in the one block after. (Block 2 is wrong, then Block 3 will be wrongly Dec)	Enc occurs sequentially but Dec can be done in parallel	IV introduce randomness to Enc process. Length of IV is same as block size (128 bits), IV is the first block, IV is freshly chosen for every process Enc, IV is not kept a secret since what attacker needs is the secret key.

Week 2: Symmetric Encryption (cont)

Use cases: Larger data/messages, Data where security is a high priority.	Comparison between ECB & CBC: ECB has identical ciphertext blocks while CBC has different ciphertext blocks	For ECB, each block is Enc/Dec independently while for CBC proper Enc/Dec requires correct previous ciphertext blocks, For ECB no error propagation while for CBC ciphertext block error affects the decryption of itself and next block.	Cryptanalysis: Methods for gaining access to encrypted contents/information.
Ciphertext only attack: Only know ciphertext, need to deduce plaintext.	Known plaintext attack: Knows both plaintext and its corresponding ciphertext, goal is to find secret key to decrypt other ciphertexts. (Note you just know the mapping of plaintext & ciphertext, but you don't choose the pair)	Chosen Plaintext attacks: Obtain the ciphertexts corresponding to plaintexts chosen by you. (Note I chose the plaintext)	Chosen ciphertext attack: Obtain the plaintexts corresponding to ciphertexts chosen by you.
Frequency analysis: Analyse the frequency of letters in ciphertext, compare them with expected frequency in the English Lang, Start guessing the potential mapping of ciphertext to plaintext.	Brute force: # combinations for a 256 bit key is (2 to power of 256). Attacker will need to try all possible keys which is computationally infeasible.	Moore's law: Computing power doubles every 1.5 year (18 months)	

Week 3: Public Key Encryption

Why we need Public Key Enc? For symmetric Enc, to exchange secret data, you must have a shared secret. (Chicken-egg problem)	# of different secret keys in total = n(n-1)/2, where n is the number of people	# of secret key stored by each user: n-1, where n is the number of people	Public Key Enc Process (Alice wants to send message to Bob): Alice Plaintext (message) -> Encryption algorithm (RSA) + Bob public key -> Ciphertext -> Decryption algorithm (RSA) + Bob private key -> Bob receive back Alice plaintext (message)

Week 3: Public Key Encryption (cont)

Bob public key is accessible to everyone	Bob private key is only known to Bob	It is computationally infeasible to compute Bob private key from public key.	RSA encryption: Performed only on message sizes smaller than the RSA modulus (n) , M <=n
Public Key = (n, e), Private Key = d	Enc Formula: C = M^(e) % n	Dec Formula: M = C^(d) % n	Security does not depend on e, e is a random value so there can be many co primes.
If n is small, easy to factor n to obtain p and q -> lambda = LCM (p-1, q-1) -> 1<e<lambda, gcd(e, lamda) = 1 -> d = e^(-1) % lambda	But when n is large enough (>2048 bits), it is infeasible to factor n to obtain p and q and from there d.	RSA deterministic problem: Attacker has the public key and can encrypt chosen plaintexts (Chosen plaintext attack) -> Attacker can test if they are equal to stolen/intercepted ciphertext. If a match is found, corresponding plaintext to the stolen cipher text is discovered.	Solution to deterministic problem: Before encryption, RSA choose a random padding R -> Encrypt (P||R) -> Different ciphertexts for even same messages.
Enc formula: C = (R||P)^(e) % n	Dec formula: (R||P) = C^(d) % n	If padding is 34 bytes and n is 256 bytes, P = 222 bytes. What if P > 222 bytes? When M > n, then it cannot be Enc using RSA as Enc & Dec would yield different results. Solution to this is to use Hybrid Enc	Limitations of RSA: Toruble Enc Large files, Textbook RSA is subjected to chosen plaintext attack as it is deterministic.
Comparison between Public Key Enc & Symmetric Key Enc: For public key encryption, public key can be sent over public channel while for symmetric key Enc, secret key must be sent over a secured channel.	Public key Enc is scalable for multi party communications but for symmetric Enc it is not scalable for multi party communications.	Public key Enc has long keys (2048 bits) while symmetric Enc has relatively shorter keys.	Public Key Enc has a slow Enc speed while symmetric Enc has a fast Enc speed.

Hybrid Enc (Alice to Bob): Alice selects a AES key K and then Enc message P to get C1 -> Alice use Bob's Public Key to Enc AES key K to get C2 -> Alice share C1 and C2 over public channel to Bob -> Bob use his private key to Dec C2 to get AES key K -> Bob use AES key K to Dec C1 to get message P.