Sec. obj.: Confidential (Private: Students grades vs unauthorized usage), Integrity (Trustable: Doc in hospital), Available (Uptime: Authenticator, Website)

> Protec Valuable [Hardw, Softw, Data] Attack use one val to attack others. [From hard to data].

[Authenticity (Validate origin), Authorization (Perms/Ctrl Privilege), Non-Repudiation (Accountability/Acknowledge receival), Privacy (Rights of individual on what's collected)] Threats (Circumstance cause harm/loss), Vulnerabilities (Sec. weakness, exploitable), Attacks (Exploit of vulner. by threats), and Controls (Prot. measure, blok threat by ctrl vulner)

Threats vs. Vulnerabilities (Threat: benign or malicious; mistke or fail or attck; human or computer initiated). (Who: Script kiddies, Amateurs(avg usr), crackers(chall), Criminals(profit)) Threats - interception, interruption, modification, fabrication. (Atck Surface: Full set of vulner., Atck tree: Technique for exploit)

Vulner in hardware (easiest control, plug device/spill/backhoe (cut line)), software (break, del, theft, life-cycle (born, discover, patch), 0-day), and data (info of people,wiretap,bug) Defense controls: prevent (close vulner.), deter (diffic), deflect (other target), detect (know when), recover (mitigate effect).

(Insider problem: Vulner. to inside atck, designed for extern, policy enforcement needed, Trojan horses?)

MOM: method (skills, knowledge, tools), opportunity (time and access), motive (expected gains) : Control? – Eliminate one of them. (Adminis: Instruct react, Tech: Hardw/Softw) Principles (Life Cycle: Planning - Policy implement - Monitor - Manage - Intrusion detect - Sec. Assess - Risk Anal - Sec Policy Creation - LOOP)

- of Adequate Protection (Prot till lose value: Protec to consistent with val) - of Easiest Penetration (Intruder- easiest/any means, site/method not obvious, not where strongest defence!) - of Weakest link (Security is no stronger than the weakest link.) - of Effectiveness (Controls must be used and used properly to be effective; Efficient; Easy to use; Appropriate)

S: sender (Alice); R: recipient (Bob); O: outsider or intruder. Eve: Evedropper; Mallory: Malicious attacker ; Crpyto algo (way plaintext is transformed into the cipher)

-Cryptosystems (system for encryption and decryption, algorithm aka cipher- encryption/decryption, all possible plaintexts, all possible ciphers, all possible keys)

-Cryptology: Cryptography (conceals data vs unauth access; encipherment, digital signature, authentication exchange)+ Cryptanalysis (Study methods to get meaning of encrypted w/o accessing information.) Kerckhoffs’s principle (system must not be secret, must able to fall to enemy w/o inconvenience)

-Unconditional secure (cipher not enough to get plaintext; unbreakable) vs. computational secure (cost breaking > value/time needed > lifetime of data)

-Shannon Secrecy (Probability of guessing the plain knowing cipher = Probability guessing without knowing/Probability of any message giving a cipher is the same)

probability models (concepts) (perfect secrecy iff as many possible keys as possible plaintexts, every key equally likely)

confusion vs. diffusion (Substitution adds confusion: Make relationship b/w plaintext and cipher(or cipher & key) complex possible change plain affect cipher unpredictable)

Transposition adds diffusion: Dissipate stats struct of plain to long-range stats of the cipher, changing plain affects cipher big, have plain bits affect each cipher bit)

Secret-key cryptography (single key needed to encrypt and decrypt messages; shared secret between the communicating parties)

Public-key cryptography (PKC) : symmetric key systems protect encryption key. keys needed (n\*n-1/2). Produce related key pair; Share public key; Keep secret key.

Cryptographic hash functions (Hash function is a lossy compression function; c-hash: secret key is used in hash function; Messages of variant length, Hash fixed size)

Algorithms -Shift cipher, Caesar cipher: (Caesar cipher is a special case of shift ciphers, Caesar means shift cipher & key=x, shift key is the mapping by shift letter in order alphabets)

-Substitution ciphers (shift is special case of substitution; substitute one letter with another), monoalphabetic ciphers (substitute one letter w/ another, each letter used only once.)

-Polyalphabetic ciphers (use multiple alphabets; Homophonic: multiple possible output for 1 input; Polygram: encipher groups of letters at once)

-Vigenère Cipher (Constructs a table; Each row in table is different shift; sender, receiver need know and agree on seq. of rows expressed as key, disguise pattern)

-One-time pad (stream of random data as the key, bit in keystream used to encrypt bit of plain, uniform output independent of plain bit, XOR is mod-2 opration, same key encrypt and decrypt, a secret key encrypt scheme, theory cant crack, Key is random, v. quick to compute, Encry & decry take same operation, Key must be long)

-Key weaknesses in security: small key space (too short, vulnerable to the brute-force attacks), static statistical patterns (Freq analys. correlates statistical patterns)

-Brute force: key exhaustive search attacks (Attckr intercept q plain-cipher pairs, encrypt w/ same secret key, tries all possible keys of k-bit length to find one Key)

Attempts needed to crack, on average (2^k-1) or in worst cases(all keys in key space = 2^k, P(cracking) = 1/(2^k))

What determines the efficiency of the brute force attack? (# of possibilities of the key as try every possible key, or # of possible subst letters, possible keys known)

-Frequency analysis (Freq analys. correlates statistical patterns, Before + More possible guesses, require more computation)

Why it works in cracking substitution and transposition ciphers (substitut ciphers susceptible to freq analysis, as some letters are more common than others)

Meet-in-the-middle (key unknown, but middle value is same encryption/decryption, run 2 parallel exhaustive searches to crack, do 2 encrypt/decrypt operations)

Man-in-the-middle (attacker insert, alter, and delete messages, think they are talking directly, performing two separate key exchanges)

Amnt of info known to atcker: Cipher-only (frequency analysis), known-plaintext (Knows some plaintext-cipher pairs. Ex. KES, meet-in-the-middle attack),

CPA (Chosen-plaintext attack-Obtain cipher for any plain of choice), CCA (Chosen-cipher attack- Decrypt any cipher except target)

DES (minor variation of Feistel struct, Symmetric: same key to decrypt, Use: Substit-confusion & Permutations-diffusion, standard arithmetic and logical operators)

-concepts: block (64-bit blocks, 16 rounds, 16 times/block), block size (64-bit), key length (56-bit keys, 8 parity), Init permut (IP) (8×8, left half Li/right Ri- 32 bits)

round function (Mangler func: Permut w/ expansion (Ri to 48 bits, plain affect cipher), S-Box (Shrink Ri to 32 bits, subst: 6-bit input, 4 output), P-Box (rearranged by fixed permutation)) AES (Rijndael - Standardized as AES, private-key symmetric block, Each round four operations, views bytes as elements)

-concepts: block, block size (128 bits = 16 bytes), key length (10 rounds: 128-bit key, 12 rnds(192-bit), 14 (256))how confusion (substitution) and diffusion (permutation) are achieved. Feistel (Product cipher: realizes 2^k possible transformations k-bit key and m-bit blocks), byte substitution (S-box: covert byte in input state array to other byte in output s.a, non-linear mapping), shift-row (Cyclically left shift bytes in row by offset), mix-column (Each column processed separately: 1 byte in an input col to create four bytes in the out col, XOR all), add round key (XOR the state with 128-bit round key)

-comparison: DES(security strength only 2^55, key 2^156), double-DES (2DES is only 2^56, double security strength, encryption, not worth), 3DES (doubles key length to 112 bits, speed v. slow, 3x DES, more secure, DES + inverse DES + DES, in ANSI X9.17), AES (128-bit key (AES-128), its strength is 2^127, Most Efficient)

Modes of operations: ECB (Electronic Codebook: Divide plain into blocks, encrypt each block with the key, concat for output), CBC (Cipher Block Chaining: Plaintext of block i is XOR'ed with the cipher of block (i-1)before it is encrypted) , CTR(Counter Mode: counter that is equal to the plaintext block size, Identical blocks encrypted differently), CFB (Cipher Feedback Mode: Cipher of previous block feedback to current block (in shift register)), OFB (Output Feedback Mode: Shift output of previous block feedback to current block)

-concepts: info leakage (Identical blocks of plaintext will produce identical blocks of cipher), cipher manipulation (No integrity checks of blocks/order, so blocks can be re-ordered or inserted), parallel processing (blocks be processed in parallel), error propagation (if error in the plain block, error in more than one cipher block?)

RSA (Rivest-Shamir-Adelman, relies on difficulty of factoring large composite numbers in finite set, recommended key length is 2048 bits (1024 bits in practice), encrypt 214 (86) bytes)

-Schemes: key generation, encryption, decryption (Look at the other side)

concepts: p (primes), g (hash fn, SHA-256 for g and h), n (p\*q),

phi(n) (p-1\*q-1, number of integers that are less than or equal to n and relatively prime to n);

selection of e (usually 3 or 65537 (2^16+1)) and d (denote the public and private keys resp, large integers for keys ), input space (plain), out sp (cipher) (check up)

concept: textbook RSA (Bad textbook RSA, small plaintext, easy to try all plain, use larger keys (2048 bits), and messages), padding (padding called PKCS#1 OAEP)

concept: why RSA is secure (Factoring problem: large +ve int n find primes is hard, RSA problem: no known efficient algorithm for Given 𝑐, 𝑛=𝑝𝑞, and 𝑒, find 𝑚, Finding RSA private key: at least as difficult as factoring 𝑛, should NOT use the same 𝑛)

-Encryption: key distribution (why) (PKC: transfer secret key securely, secure to exch. keys w/o prior exchange of secret keys, provides integrit and authentic via digital sign) -Signing: (Using private key, a user can sign a message, users use the public key (public parameters) can verify the signature.)

digital signatures (digitally signs his message using the private key, compute signature (hve priv key: Bob), verify a signature (have public key: everyone)),

PKI (Public Key Infrastructure, bind the public key of a user in the digital certificate that is issued to this user, certificate can be verified by its digital signature)

Diffie-Hellman key agreement (share secret over insecure media. Eg X.509, SSL, IPsec)

-concept: why we need D-H for key agreement (usrs exchange puzzles; solved easily with additional info, discrete logarithm problem in multiplicative group)

-the protocol: scheme (Public info: g and p; Private info: a and b, randomly selected (Alice & Bob), g^ab mod p = Shared key)

-man-in-the-middle attack and defenses (grab data from middle and fake key swap)

-DSA: concept (signer generates secure hash of message, then uses their private key as a signature from hash. The sign and message can be shared for verification purposes.)

-ECC: concept (Elliptic-Curve Cryptography: elliptic curve-based encryption methods, E.g., the Koblitz curve, curve has all points whose coordinates (𝑥, 𝑦) satisfy the equation for any a and b, non-vertical line passes through at most three points, Chord method: given any two points, P and Q, we can find the third point R, based on the line, and define a special operation on the curve, ECC achieves the same security strength as RSA with smaller keys, Smaller keys ⇒Faster computation, mathematics behind EC crypto is sophisticated, EC cryptosystem is used in digital signatures, Diffie-Hellman, e.g., ECDH.)

Hash -concepts: hash properties (One-way (no invert), Collision resistance, Weak collision resistance, should look random) and application (Password hashing one-wayness, Software integrity (Weak collision resistance), Commitment (e.g., auction, use onewayness))

-concept: collision and weak collision resistance (Given a randomly selected 𝑥, hard to find 𝑥’ such that ℎ(𝑥)=ℎ(𝑥’), if found, is collision), collision resistance (attacker only needs to find any collision, hard to find any 𝑥 ≠𝑥’such that ℎ(𝑥)=ℎ(𝑥’))

-calculation: brute force to find a hash collision (Birthday Paradox: brute-force collision search is only 𝑂(2^𝑚/2), 160-bit hash, 2^80 is collision chances > 0.5)

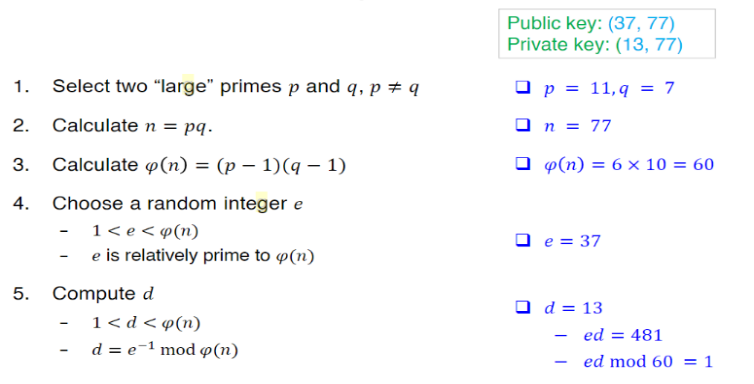
-Concept: why need Message Authentication Code MAC (Get integrity: Transmit over insecure channel, use to verify message content is not altered, want a cryptographic checksum, idea is given message, only someone who knows key can compute the correct MAC, need secure (only the key can generate or verify the MAC), collision resistance (hard to find two messages that have the same MAC) and small (don’t want to increase message size), (MAC key ≠ encryption key)

-Concept: how to implement MAC (secret-key vs, public-key vs. cryptographic hash)

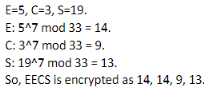
(Block ciphers: e.g., we can use the last 16 or 32 bits of ciphertext as MAC, NIST recommends CMAC, Hash-based MAC (HMAC), Crypt hash in MAC – HMAC; standard approach for keyed hash MAC (RFC2104), In SSL/TLS, mandatory for IPsec, Hash is used as a “black-box” here; use any hash function; Proven secure Want both confidentiality and integrity: we combine encryption scheme and a MAC, should do encryption first: Encryption-then-MAC as MAC is deterministic)

secure cipher + ciphertext integrity →Authenticated Encryption (AE)

RSA KEY GEN



AES:

 Encryption & decryption schemes: RSA -

p = 3 and q=11, and e=7.

public key is <e, n> = <7, 33>.

Convert “EECS” into numbers as :

