Lecture notes for 2.1-2.4

LOTS of definitions today

Def: Cryptography
- The art [science] of secret writing
- Codemaking

Def: Cryptanalysis
- The art [science] of breaking codes
- Codebreaking

Def: Cryptology
- Cryptography UNION cryptanalysis

Def: Encryption [enciphering]
- Conversion of plaintext [cleartext] to ciphertext
- C = E(P)

Def: Decryption [deciphering]
- Conversion of ciphertext to plaintext
- P = D(C)

Def: (Reversible) Cryptographic scheme [cipher]
- Encryption algorithm UNION decryption algorithm
- Constraint: D(E(P)) == P

Cryptographic schemes have 0, 1, or 2 keys

Def: Cryptographic hash function
- H = E(P)
- Preimage resistant
  - Given H, “hard” to find P
- Second preimage resistant
  - Hard to find P and P' satisfying E(P) = E(P')
- Properties:
  - Non-reversible (preimage resistant)
  - 0 keys: anyone can compute a hash of a plaintext
  - 1 key: only people possesing key can compute a hash
  - Principle: non-linear bit transformations (substitutions, transpositions)
- Common algorithms:
  - MD5, SHA1, SHA256

Def: Symmetric [secret] key cryptography
- P = D(E(P, K), K)
- Properties:
  - Reversible
  - Same encryption & decryption key
  - Key must be a shared secret, called “secret key”
  - Principle: non-linear bit transformations (substitutions, transpositions)
- Common algorithms:
  - DES, 3DES, AES

Def: Assymetric [public] key cryptography
- P = D(E(P, e), d)
- Properties:
  - Reversible
  - Different encryption & decryption keys
  - One of the keys can be published, called “public key”
  - One key not disclosed, called “private key”
  - Principle: math, NP-hard problems
- Common algorithms:
  - RSA

Foundation of confidentiality & integrity
- Confidentiality: Encrypt private data
- Integrity: Use a cryptographic hash to detect alterations of data
  - “Summary” of the input data, attacker cannot recompute summary if changing data
    - This is the principle used to verify passwords at system login

Comparisons
- Symmetric key crypto
  - Fast
  - Must manage many keys
  - Must protect many keys
- Assymetric key crypto
  - Slow
  - Must manage no keys
  - Must protect only own private key

Principles:
- Attackers know algorithms (Why?)
- Entire security comes from secrecy of the key

Key distribution problem
- Symmetric key crypto
  - How do both parties share the key ahead of time? Need confidential channel, but doesn't exist until keys already established. Chicken-and-egg.
  - Generally done via out-of-band channel (phone call, keyboard)
  - Diffie-Hellman key exchange
- Assymetric key crypto
  - How do I trust your public key? Attacker may be spoofing identity
  - Certificates (you've gotten cert warnings in your web browser)
    - Assertion by someone you trust that an unknown identity is legit
    - Root trust set via out-of-band channel (OS update mechanism)

Cryptanalysis techniques
- Brute force (try all possible keys)
- Discover mathematical properties of ciphertext
- For cryptograms: use properties of natural language (covered in textbook)
- Guess at what plaintext might have been (headers...)

Def: Ciphertext-only attack
- Cryptanalyst recovers plaintext (& possibly key) from ciphertext

Def: Known plaintext attack
- Cryptanalyst recovers key from plaintext / ciphertext pair

Def: Chosen plaintext attack
- As before, but attacker gets to choose plaintext
- Some plaintexts may trigger weaknesses in the system & reveal info
- Occurs when the cipher is operating as an encryption oracle

Def: Perfect secrecy
- The property that every plaintext (of the appropriate length) is mathematically equally likely for a given ciphertext
- Requirement: bits of key >= bits of plaintext
- Example: one-time pad

Modes of operation apply to many enciphering & deciphering algorithms

Def: Stream mode
- A cryptosystem that encrypts and decrypts data symbol-by-symbol
- Useful for handling low latency data (cannot wait for entire block)

Def: Block mode
- A cryptosystem that encrypts and decrypts data block-by-block
- Useful for bulk encryption/decryption

Rebuttal to text: Shannon principle 4
- “Errors should not propogate”
- Lack of propagation may be weakness in system
- Want data to continually mix, affect data further away
- Stream & block ciphers mix current data with previous data

Basic math
- Modular arithmetic
- Exponentiation

Diffie-Hellman key exchange
- Protocol to allow two parties to establish secret key over insecure medium
- Invented (in secret) by GCHQ in early 1970s
- 1. Alice choose a, g, p (p prime, g < p)
- 2. Alice compute A = g^a mod p
- 3. Alice sends g, p, A to Bob
- 4. Bob choose b
- 5. Bob compute B = g^b mod p
- 6. Bob send B to Alice
- 7. Alice compute K = B^a mod p
- 8. Bob compute K = A^b mod p
- K is the shared secret key
- Works because g^ab mod p = g^ba mod p && K, a, b are never sent in the clear && g is a generator of Z_p since gcd(g, p)==1

Trivial cryptosystems
- Alphabetic shift
  - Caeser cipher
- Vegeniere