Cryptography

Relation to this Course

- Underlies many fundamental services
  - Confidentiality
  - Authentication
  - Data integrity
- Is perhaps *the* basic foundation

A Brief History

- Steganography: "covered writing"
  - Demaratus (5th century B.C.)
  - German microdots (WWII)
  - Crucial flaw: Discovery yields knowledge
    - Confidentiality through obscurity
- Cryptography: "secret writing"
  - TASOIINRNPSTO and TVCTUJUVUJPO

A Brief History

- Two basic types of cryptography
  - Transposition (TASOIINRNPSTO)
    - Message broken up into units
    - Units permuted in a seemingly random but reversible manner
    - Ex: scytale
    - Difficult to make it easily reversible only by intended receiver
    - Exhibits same first-order statistics

A Brief History

- Two basic types of cryptography (cont)
  - Substitution (TVCTUJUVUJPO)
    - Message broken up into units
    - Units mapped into ciphertext
    - Ex: Caesar cipher
    - First-order statistics are isomorphic in simplest cases
    - Predominant form of encryption

How Much Security?

- Monoalphabetic substitution cipher
  - Permutation on message units—letters
  - 26! different permutations
  - Each permutation considered a *key*
  - Key space contains $26! = 4 \times 10^{26}$ keys
    - Equal to number of atoms in a gallon of water
    - Equivalent to a 88-bit key (more than DES!)

How Much Security?

- So why not use substitution ciphers?
  - Hard to remember 26-letter keys
    - But we can restrict ourselves to shorter keys
    - Ex: JULISCAERBDFGHKM, etc
  - Remember: first-order statistics are isomorphic
    - Vulnerable to simple cryptanalysis
    - Hard-to-read fonts for crypto?!

Substitution Ciphers

- Two basic types
  - Symmetric-key or conventional
    - Single key used for both encryption and decryption
    - Keys are typically short, because key space is densely filled
    - Ex: DES, 3DES, RC4, Blowfish, IDEA, etc

Substitution Ciphers

- Two basic types (cont)
  - Public-key or asymmetric
    - Two keys: one for encryption, one for decryption
    - Keys are typically long, because key space is sparsely filled
    - Ex: RSA, El Gamal, DSA, etc

Conventional Cryptography

- Block ciphers encrypt message in units called blocks
  - DES: 8-byte key (56 key bits), 8-byte block
  - Larger blocks make simple cryptanalysis useless (at least for short messages)
    - Not enough samples for valid statistics
    - "Octogram Statistics Needed"

Key and Block Size

- Do larger keys make sense for an 8-byte block?
  - 3DES: Key is 112 or 168 bits, but block is still 8 bytes long (64 bits)
  - Key space is larger than block space
  - But how large is permutation space?

Anatomy of a Block Cipher

- DES: Data Encryption Standard
  - Developed as Lucifer (one of a few) at IBM in 1970s
  - Break message into 8-byte (64-bit) blocks
    - Each block broken into 32-bit halves
    - Initial permutation
    - 16 rounds of scrambling
    - Final (reverse) permutation

The Scrambling Function

- In each round i, we have L_i and R_i
  - $L_{i+1} = R_i$ ← typical of Feistel networks
  - $R_{i+1} = L_i + f(R_i)$
- f-function
  - Key is compressed and permuted to 48 bits
  - R_i is expanded and permuted to 48 bits
  - 48 bits XOR'd, passed through S-boxes, then permuted again

Key Compression

- Reduction to 56 bits (no parity bits)
- Broken into halves
  - Each half is rotated by 1 or 2 bits
  - 48 bits out of 56 selected
- Why do this?
  - Use a different set of bits for each round
  - Not exactly symmetric

Data Expansion

- Data broken into 4-bit groups
- Each group expanded to 6 bits
- Why do this?
  - Match subkey length
  - Data diffusion occurs faster

Substitution Boxes (S-Boxes)

- 48 bit result broken into 6-bit units
- Each unit passed through an S-box
  - 6-bit input, 4-bit output
  - Each S-box is a 4x16 array of 4-bit numbers
  - b_1 and b_6 specify row, b_2 through b_5 specify column
- End result passed through P-box

Modes of DES Operation

- Electronic Codebook (ECB)
  - Each block encrypted in isolation
  - Vulnerable to block replay
- Cipher Block Chaining (CBC)
  - Each plaintext block XOR'd with previous ciphertext before encryption
  - Easily incorporated into decryption
  - What if prefix is always the same? IV!

Modes of DES Operation

- Cipher Feedback (CFB)
  - For encrypting character-at-a-time (or less)
  - Chains as in CBC
  - Also needs an IV
    - Must be unique—why?
- Output Feedback (OFB)
  - Like CFB, but some bits of output fed back into input stream

Variants and Applications

- 3DES: Encrypt using DES 3x
  - Two and three-key types
  - Inner and outer-CBC modes
    - Inner is more efficient, but less secure
- Crypt: Unix hash function for passwords
  - Uses variable expansion permutations
- DES with key-dependent S-boxes
  - Can't be done blindly

Attacks on DES

- No known systematic attack (for 16 rounds)
  - Is DES "closed" (that is, a group)?
    - If it were, double encryption would be useless
    - Is it useful at all?
  - Is DES "pure"?
    - If it were, triple encryption would be useless
- Brute force attacks only
  - Try all 2^{56} keys!

Lucifer Goes Standard

- Generally regarded in 1970s as one of the strongest cryptosystems
- Heading toward standardization as DES
  - NSA managed to get key size reduced to 56 bits (from 112), yielding 10^{17} keys
  - Also apparently changed S-boxes
  - Why (or why not) do this?

Certification of DES

- Had to be recertified every ~5 years
  - 1983: Recertified routinely
  - 1987: Recertified after NSA tried to promote secret replacement algorithms
    - Withdrawal would mean lack of protection
    - Lots of systems then using DES
  - 1993: Recertified after continued lack of alternative

Enter AES

- 1998: NIST finally refuses to recertify DES
  - 1997: Call for candidates for Advanced Encryption Standard (AES)
  - Fifteen candidates whittled down to five
  - Criteria: Security, but also efficiency
    - Compare Rijndael with Serpent
  - 2000: Rijndael selected as AES

Structure of Rijndael

- Unlike DES, operates on whole bytes for efficiency of software implementations
- Key sizes: 128/192/256 bits
- Variable rounds: 9/11/13 rounds
- Rounds are not Feistel networks

Structure of Rijndael

- Round structure
  - Run block through S-box
  - Permute result into 4x4/4x6/4x8 array of bytes
  - Multiply each byte by 1, 2, or 3 in GF(2^8)
  - Mix subkey into result

Security of Rijndael

- Key size is enough
- Immune to linear or differential analysis
- But Rijndael is a very structured cipher
  - S-box consists of byte reciprocals in GF(2^8)
  - Permutations are regular
- Attack on Rijndael's algebraic structure
  - Breaking can be modeled as equations

Impact of Attacks on Rijndael

- Currently of theoretical interest only
  - Reduces complexity of attack to about 2^{100}
  - Also applicable to Serpent
- Still, uncomfortably close to feasibility
  - DES is already insecure against brute force
  - Schneier (somewhat arbitrarily) sets limit at 2^{80}
- Certainly usable pending further results

Public Key Cryptography

- aka asymmetric cryptography
- Based on some NP-complete problem
  - Unique factorization
  - Discrete logarithms
    - For any b, n, y: Find x such that $b^x \bmod n = y$
- Modular arithmetic produces folding

A Short Note on Primes

- Why are public keys (and private keys) so large?
- What is the probability that some large number p is prime?
  - About 1 in 1/ln(p)
  - When $p \sim 2^{512}$, equals about 1 in 355
    - About 1 in 355^2 numbers $\sim 2^{1024}$ is product of two primes (and therefore valid RSA modulo)

RSA

- Rivest, Shamir, Adleman
- Generate two primes: p, q
  - Let n = pq
  - Choose e, a small number, relatively prime to (p-1)(q-1)
  - Choose d such that $ed = 1 \bmod (p-1)(q-1)$
- Then, $c = m^e \bmod n$ and $m = c^d \bmod n$

An Example

- Let p = 5, q = 11, e = 3
  - Then n = 55
  - d = 27, since (3)(27) mod 40 = 1
- If m = 7, then c = 7^3 mod 55 = 343 mod 55 = 13
- Then m should = 13^{27} mod 55

An Example

- Computing 13^{27} mod 55
  - 13^1 mod 55 = 13, 13^2 mod 55 = 4, 13^4 mod 55 = 16, 13^8 mod 55 = 36, 13^{16} mod 55 = 31
  - 13^{27} mod 55 = (13)(4)(36)(31) mod 55 = (1872 mod 55)(31) mod 55 = 62 mod 55 = 7 (check)

Other Public Cryptosystems

- ElGamal (signature, encryption)
  - Choose a prime p, and two random numbers g, x < p
  - Public key is g, p, and y = g^x mod p
  - Private key is x; to obtain from public key requires extracting discrete log
  - Mostly used for signatures

Other Public Cryptosystems

- Elliptic curve cryptosystems
  - $y^2 = x^3 + ax^2 + bx + c$
  - Continuous elliptic curves used in FLT proof
  - Discrete elliptic curves used to implement existing public-key systems
    - Allow for shorter keys and greater efficiency

Digital Signatures

- Provides data integrity
  - Can be done with symmetric systems
    - Verification requires shared key
    - Doesn't provide non-repudiation
- Need proof of provenance
  - Hash the data, encrypt with *private* key
  - Verification uses public key to decrypt hash
  - Provides non-repudiation

Digital Signatures

- RSA can be used
- DSA: Digital Signature Algorithm
  - Variant of ElGamal signature
  - Adopted as part of DSS by NIST in 1994
  - Slower than RSA (but likely unimportant)
  - NSA had a hand in its design (?!)
  - Key size ranges from 512 to 1024 bits
  - Royalty-free

Key Exchange

- Diffie-Hellman key exchange
  - Choose large prime n, and generator g
    - For any b in (1, n-1), there exists an a such that $g^a = b$
  - Alice, Bob select secret values x, y, resp
  - Alice sends $X = g^x \bmod n$
  - Bob sends $Y = g^y \bmod n$
  - Both compute $g^{xy} \bmod n$, a shared secret
    - Can be used as keying material

Hash Functions

- Given m, compute H(m)
- Should be…
  - Efficient: H() easy to compute
  - One-way: Given H(m), hard to find m' such that H(m') = H(m)
  - Collision-resistant: Hard to find m and m' such that H(m') = H(m)

Use of Hashes in Signatures

- Reduce input to fixed data size
  - MD5 produces 128 bits
  - SHA1 produces 160 bits
- Encrypt the output using private key
- Why do we need collision-resistance?

Signing Using Only Hashes

- Generate random n_1, n_2, n_3, …
- Distribute $H(n_1)$, $H(n_2)$, $H(n_3)$, …
- To authenticate message m_i, release n_i
- Problems
  - Seem to need 2^{128} or 2^{160} hashes to sign
  - Need to bootstrap signature
- Resolvable?

Quick Announcements

- Interim place for notes
  - http://www.isi.edu/~brian/csci530/
- Prof Neuman will address D-clearances
- Paper assignment will be introduced in next two weeks