CS 161 – Introduction to Cryptography; Asymmetric Cryptography

13 September 2006

Symmetric crypto

- Advantages
  - Fast
  - Reasonably well-understood
  - Standardized
  - Can be implemented in hardware easily
  - Exhaustive search attack hard (with large key size)
- Disadvantages
  - Key distribution
  - Single target
  - Still needs to be implemented in protocols

Attacks on cryptography

- Direct attack
  - example: exhaustive search
- Known plaintext
- Chosen plaintext

- Usual assumptions: chosen plaintext attack; attacker knows E, D but not key

Notation

- Ciphertext = Encryption (Plaintext, encryption-Key)
  - sometimes we use “cleartext” instead of “plaintext”
- Key $\in$ Keyspace
- Keysize = $\log_2(\ |\text{Keyspace}|\ )$

- c=E(m,k) (or $c=E_k(m)$ or $c=\{m\}_k$)

- Also Plaintext = Decryption(Ciphertext, decryption-Key)
- encyption-Key = decryption-Key (symmetric)
- encyption-Key ≠ decryption-Key (asymmetric)
- $m=D(c,k)=E^{-1}(c,k)$ (or $c=D_k(m)$)

Asymmetric cryptography

- encryption-Key ≠ decryption-Key
- We cannot simply run operations backwards
- Some things are hard to reverse
  - Often “hard” means “not in P”
  - Cryptanalysis is always easy in NP
  - Does P = NP?
- Multiplication
  - Easy to multiply two large primes
  - Hard to factor
  - Factoring up to 663 bits (200 digits) now demonstrated
    - Intensive computing; record set in May 2005
  - More efficient factoring methods unknown
  - Proving factoring is hard probably would require solving P vs. NP

Using hard problems to make crypto

- Gauss (building on work by Fermat) proved
  - If p and q are primes and
  - If m is not a multiple of p or q
  - Then $m^{(p-1)(q-1)} = 1 \bmod pq$

- Example, p=3, q=5, pq = 15, (p-1)(q-1) = 8
  - $1^8 = 1 = 1 \bmod 15$
  - $2^8 = 256 = 1 \bmod 15$
  - $4^8 = 65536 = 1 \bmod 15$
  - $7^8 = 5764801 = 1 \bmod 15$
  - $8^8 = 16777216 = 1 \bmod 15$
  - $11^8 = 214358881 = 1 \bmod 15$
  - $13^8 = 815730721 = 1 \bmod 15$
  - $14^8 = 1475789056 = 1 \bmod 15$

RSA

- Rivest, Shamir, Adleman (1978 – published 1979)
- Idea:
  - Given e, find d, such that $ed = K(p-1)(q-1)+1$ for some K
  - Encryption: $c = E(m) = m^e \bmod pq$
  - Decryption: $D(c) = c^d \bmod pq$
  - So $D(E(m)) = m^{ed} \bmod pq = m^{K(p-1)(q-1)+1} \bmod pq = m$
- Issues:
  - Given e, how can we find d?
    - Answer: use EGCD (extended greatest common divisor)
      - Euclidean algorithm
    - Given x, y, EGCD finds $Ax + By = GCD(x, y)$
    - Let $x=e$, $y=(p-1)(q-1)$, then $Ae = (-B)(p-1)(q-1) + 1$
  - How can compute exponentiation modulo pq fast?
    - Repeated squaring mod pq – use binary form of number

Factoring & RSA

- Factoring is easy → RSA is easy
- We have not proved that RSA is as hard as factoring.
- Soon, we will see a cryptosystem that is equivalent to factoring (Rabin).

RSA allows for “public keys”

- Encryption key public, decryption key private
  - Easy way to send secret messages
  - If we can guess plaintext, we can break (so we add random bits)
  - Decryption only by intended recipient
  - Perfect for distributing symmetric keys
- Encryption key private, decryption key public
  - Only I can send messages, anyone can verify (and read)
  - A type of “digital signature”
  - We will develop this idea in detail

Asymmetric crypto

- Advantages
  - Doesn’t require advance set up
  - Strongest forms are as hard as factoring
  - Perfect for solving key distribution problem
  - Good for building protocols
- Disadvantages
  - Slow, slow, slow (& takes space too)
  - Secrecy & source authentication takes two encryptions
  - Need to find a way to prove “public keys” are honest
    - Future lecture: public key hierarchy