CS 161 – Zero knowledge

20 September 2006

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 = $\log_2$(|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 $c=D_k(m)$)

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.

- We need better cryptosystems
  - Secret sharing – allows party to store message secretly
  - Rabin signatures – equivalent to factoring

Chinese Remainder Theorem

- We can represent numbers mod pq
- Alternatively as a pair mod p and mod q

- 1 = <1 mod 3, 1 mod 5>
- 7 = <1 mod 3, 2 mod 5>
- 12 = <0 mod 3, 2 mod 5>

Square roots

- This means that a square root mod pq has four roots.
- Suppose that $r^2 = m \bmod pq$
- And r = <s mod p, t mod q>
- Then for square roots are:
- <s mod p, t mod q>
- <-s mod p, t mod q>
- <s mod p, -t mod q>
- <-s mod p, -t mod q>
- If we can find the square roots, then we can factor pq
- <s mod p, t mod q> + <-s mod p, t mod q> = <0 mod p, 2t mod q> = multiple of p

Rabin Signature algorithm

- If we can factor pq, it is easy to take square roots
- This means square roots are a great signature
- Easy to verify (just take a square)
- If someone has a square root taking algorithm then he can factor easily.

- Square roots ↔ factoring

Leaky protocols

- Many protocols leak information
- For example, consider the following authentication protocol:

> A $\rightarrow$ B: Prove you are Bob, sign message M
> B $\rightarrow$ A: Sign(M, B)

- Now Alice has some information she didn't have before
- She has Sign(M, B)
- Perfect for what kind of attack?

Zero-knowledge protocol

- Idea: interactive proof
- At the end of the proof, A is convinced B knows a proof of fact F
- But A has no information about that proof

How to prove identity using zero-knowledge

- B publishes b^2 mod pq
- B → A: r^2 mod pq (random r)
- A flips coin
- A → B: coin flip
- If heads
  - B → A: r mod pq
  - A verifies $(r \bmod pq)^2 = r^2 \bmod pq$
- If tails
  - B → A: rb mod pq
  - A verifies $(rb \bmod pq)^2 = (r^2)(b^2) \bmod pq$

Comments

1. This is an easy-to-perform protocol
2. After each round, convinced with 50% probability

   If B knows both rb & r (mod pq), he knows rb/r (mod pq)

   Fake-B will be caught 50% of the time
3. A learns nothing – if she does, she could just generate pairs <r, r^2>on her own. (Or, <rb, rb^2>.)