Secret-Sharing & Zero-knowledge Proof

Dawn Song

dawnsong@cs.berkeley.edu

Review

- **DH key exchange protocol**
- **Password authentication protocol**
- **Random number generation**

Lessons Learned

- **Seeds must be unpredictable**
- **Algorithm for generating pseudorandom bits must be secure**

Generating Pseudorandom Numbers

- **True random number generator (TRNG)**
  - **Generates bits that are distributed uniformly at random, so that all outputs are equally likely, with no patterns, correlations, etc.**
- **Cryptographically secure pseudorandom number generator (CS-PRNG)**
  - **Taking a short true-random seed, and generates long sequence of bits that is computationally indistinguishable from true random bits**

4

CS-PRNG

- **CS-PRNG: cryptographically secure pseudorandom number generator**
  - **G: maps a seed to an output G(S)**
    - **E.g., G: $\{0,1\}^{128}$ -> $\{0,1\}^{1000000}$**
  - **Let K denote a random variable distributed uniformly at random in domain of G**
  - **Let U denote a random variable distributed uniformly at random in range of G**
  - **G is secure if output G(K) is computationally indistinguishable from U**
- **Sample construction**
  - **Use the seed as a key k, and compute AES-CBC(k, 0^n)**

5

TRNG (I)

- **TRNG should be random and unpredictable**
- **Good or bad choices?**
  - **IP addresses**
  - **Contents of network packets**
  - **Process IDs**
  - **High-speed clock**
  - **Soundcard**
  - **Keyboard input**
  - **Disk timings**

6

TRNG (II)

- How to convert non-uniform sources of randomness into TRNG?
  - Use a cryptographic hash function, such as SHA1
  - Suppose x is a value from an imperfect source, or a concatenation of values from multiple sources, and it is impossible for an attacker to predict the exact value x except with probability $1/2^n$
  - Then hash(x) truncated to n bits should provide a n-bit value that is uniformly distributed, if hash() is secure

Secret Sharing

- A trusted authority TA has a secret K
- Wants to split K into n shares S1, …, Sn, distributing to n users U1,…,Un respectively, s.t.
  - A reconstruction algorithm can be used to efficiently reconstruct K from any t of the n shares
  - Any t-1 of the n shares reveal no information about K
- Such a scheme is called an (n,t) threshold secret sharing scheme

(n,n) Secret Sharing Scheme

- Suppose the secret K is an integer btw 0 and M-1
- (n,n) threshold scheme:
  - Pick $S_1,\ldots,S_{n-1}$ uniformly at random btw 0 and M-1
  - Set $S_n = K - (S_1 + \ldots + S_{n-1}) \bmod M$
- How to reconstruct K?
- What happens if n-1 users get together?

(n,t) Threshold Scheme

- Polynomials modulo prime p
  - Polynomials whose coefficients are elements mod p
  - E.g., $f(x) = x^2 + 2x + 4 \bmod 5$
  - Degree-n polynomial f (mod p) is uniquely determined by any n+1 distinct pairs (x_i, y_i) s.t. $f(x_i) = y_i$
    - Lagrange interpolation
- To (n,t) threshold share secret K:
  - Pick a random polynomial f (mod p) of degree t-1 s.t. f(0) = K
  - Share $s_i = f(i)$ for i = 1 to n
  - How to recover K?
  - How many shares do you need to recover K?
  - What happens if you have fewer shares than t?

Administravia

- Hw1 hand-in procedure

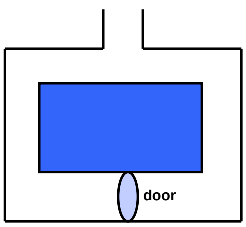
Zero-knowledge Proof

- Alice->Bob: I know the solution to Que 3 in hw 1, but I can't tell you what the solution is
- Bob->Alice: tell me, o.w. I don't believe you
- Alice->Bob: Ok, I'll prove to you that I know the solution in Zero-knowledge

Zero-knowledge protocol

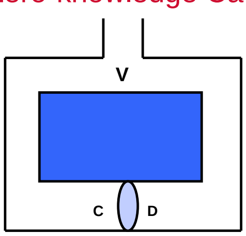
- Idea: (interactive) proof btw prover A & verifier B
- At the end of the proof, B is convinced A knows a secret satisfying a fact F
- But B has no information about that secret

The Zero-knowledge Cave (I)



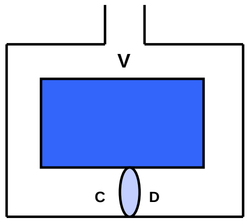
- Alice wants to prove to Bob that she knows how the magic word to open door
  - Without telling Bob the magic word

The Zero-knowledge Cave (II)



1. Alice walks to either C or D;
2. Bob stands at V, calling either Left or Right;
3. Alice complies, using her magic word to open door if needed;
4. Alice & Bob repeats steps 1-3 for n times

The Zero-knowledge Cave (III)



- What if Alice didn't know the magic word?
- What does Bob learn at the end of the proof?

How to prove knowledge of square root

- Finding square root mod N=pq is as hard as factoring
- A knows b s.t. $b^2 = y \bmod pq$, & wishes to prove to B that she knows such b.
- A $\rightarrow$ B: $s =: r^2 \bmod pq$ (A picks random r)
- B flips coin
- B $\rightarrow$ A: coin flip
- If heads
  - A $\rightarrow$ B: $t =: r \bmod pq$
  - B verifies $t^2 \equiv s \bmod pq$
- If tails
  - A $\rightarrow$ B: $t =: rb \bmod pq$
  - A verifies $t^2 \equiv sy \bmod pq$
- What if A didn't know the square root?
- What did B learn after the proof?