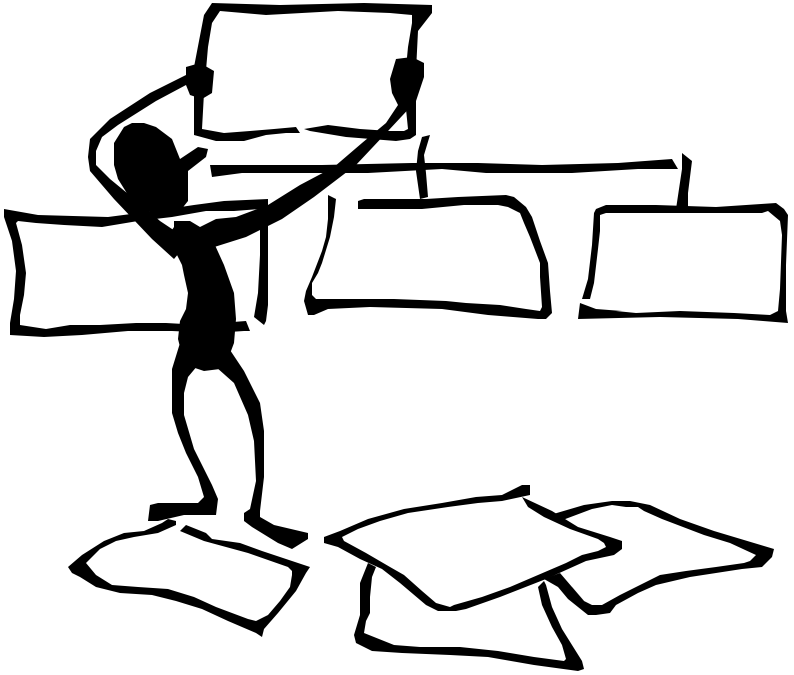
Introduction to Cryptography
CS 355

Lecture 32

Zero Knowledge Proof Protocols

Lecture Outline

- Properties of ZK proof of knowledge
- Schnorr protocol
- Noninteractive ZK



Properties Zero-Knowledge Proofs

- Properties of ZK Proofs:
  - completeness
    - honest prover who knows the secret convinces the verifier with overwhelming probability
  - soundness (is a proof of knowledge)
    - no one who doesn't know the secret can convince the verifier with nonnegligible probability
  - zero knowledge
    - the proof does not leak any additional information
- How to define soundness and ZK?

Defining the Soundness Property

- The protocol should be a “proof of knowledge”
- A knowledge extractor exists
  - that given a prover who can successfully convince the verifier, can extracts the secret
- Why the Fiat-Shamir Protocol is a proof of knowledge?
  - if the prover can respond to more than one challenge in any round, then the secret is revealed.

Defining ZK property

- Intuition: the proof is ZK if what the verifier sees during the protocol (i.e., the transcript) can be simulated without knowing the secret.
- Honest verifier ZK
  - if the verifier follows the protocol, then the transcript can be simulated
- ZK
  - for any algorithm acting as the verifier, the transcript can be simulated

Fiat-Shamir is honest-verifier ZK

- The transcript of a protocol run consists of t tuples (x, c, y) such that x is a random QR in Z_n^* and $y^2 \equiv xv^c \bmod n$
- Proof that Fiat-Shamir is honest verifier ZK
  - Construct a simulator as follows
    - Repeat the following: pick random $c \in \{0,1\}$,
    - if c=0, pick random r and outputs (r^2, 0, r)
    - if c=1, pick random y, and outputs ($y^2 v^{-1}$, 1, y)
  - The transcript generated by the simulator is from the same prob. distribution
- Fiat-Shamir is also ZK

Schnorr Id protocol (ZK Proof of Discrete Log)

- System parameter: p, q, g
  - q | (p-1) and g is an order q element in Z_p^*
- Public identity: v
- Private authenticator: s $v = g^s \bmod p$
- Protocol

1. A: picks random r in [1..q], sends $x = g^r \bmod p$,
2. B: sends random challenge e in $[1..2^t]$
3. A: sends $y=r+se \bmod q$
4. B: accepts if $x = (g^y v^{-e} \bmod p)$

Security of Schnorr Id protocol

- probability of forgery: $1/2^t$
- soundness (proof of knowledge):
  - if A can successfully answer two challenges e1 and e2, i.e., A can output y1 and y2 such that $x=g^{y1}v^{-e1}=g^{y2}v^{-e2}$ then $g^{y1-y2}=v^{c1-c2}$ and thus the secret
    $s=(y1-y2)(c1-c2)^{-1} \bmod q$
- ZK property
  - is honest verifier ZK.
  - is ZK when t is small

Converting Interactive ZK to Non-interactive ZK

- The only interactive role played by the verifier is to generate random challenges
  - challenges not predictable by the prover
- The same thing can be done using one-way hash functions

Interactive ZK Implies Signatures

- Given a message M, run all rounds in parallel,
  - generate the commitments all at the same time, let X denote all commitments
  - replace the random challenge of the verifier by the one-way hash c=h(M||X)
  - append the response

Schnorr Signature

Key generation (uses $h:\{0,1\}^{*}$ ® Z_q)

- Select two primes p and q such that
  $q \mid p-1$
- Select $1 \le a \le q-1$
- Compute $y = g^a \bmod p$

Public key: (p,q, g,y)

Private key: a

Schnorr Signature

Signing message M

- Select random secret k, $1 \leq k \leq q-1$
- Compute

  $r = g^k \bmod p$,

  $e = h(M \| r)$

  $s = ae + k \bmod q$

Signature is: (r, s)

To verify that (r,s) is the signature of M

- Compute

  $e = h(M \| r)$

- Verify that

  $r = g^s y^{-e} \bmod p$

Coming Attractions …

- Key agreement protocols