Zero Knowledge Proofs (ctd.)

Lecture 13

RECALL

ZK Proof for NP Languages

- Consider an NP language L specified by a poly-time computable predicate R: i.e., $x \in L$ iff $\exists w$ s.t. $R(x,w)=1$. A ZK proof protocol $P \leftrightarrow V$ for L has the following properties
  - Completeness: if $\exists w\ R(x,w)=1$, then $Pr[P(x,w) \leftrightarrow V(x) = 1] = 1$
  - Soundness: if $\nexists w\ R(x,w)=1$, then $Pr[P^*(x) \leftrightarrow V(x) = 1] = negl$ (for any P^*)
    - A stronger notion: Proof of Knowledge
  - Zero-Knowledge: if $\exists w\ R(x,w)=1$, then view of the verifier in $P(x,w) \leftrightarrow V(x)$ can be (indistinguishably) simulated from x
    - This is called Honest Verifier ZK (HVZK)
    - Stronger property: For any PPT V^*, there is a simulator S s.t., $View_{V^*}(P(x,w) \leftrightarrow V^*(x)) \approx S(x)$

ZK argument: soundness required only against PPT P^*

V learns nothing beyond the fact that x has the property

RECALL

HVZK Proof of Knowledge

- Proof of Knowledge: If an adversary can give valid proofs (with significant probability), then there is an efficient way to extract a witness from that adversary
- A ZK Proof of knowledge of discrete log of $Y=g^y$
  - $P \rightarrow V$: $R := g^r$
    $V \rightarrow P$: x
    $P \rightarrow V$: $s := xy + r$ (modulo order of the group)
    V checks: $g^s = Y^x R$
  - Proof of Knowledge:
    - Firstly, $g^s = Y^x R \Rightarrow s = xy+r$, where $R = g^r$
      - If after sending R, P could respond to two different challenges x_1 and x_2 as $s_1 = x_1y + r$ and $s_2 = x_2y + r$, then can solve for y (in a prime-order group)
  - HVZK: simulation picks s, x first and sets $R = g^s/Y^x$

HVZK and Special Soundness

- HVZK: Simulation for honest (passively corrupt) verifier
  - e.g. in PoK of discrete log, simulator picks (x,s) first and computes R (without knowing r). Relies on verifier to pick x independent of R.
- Special soundness: If given (R,x,s) and (R,x′,s′) s.t. x≠x′ and both accepted by verifier, then can derive a valid witness
  - e.g. solve y from s=xy+r and s′=x′y+r (given x,s,x′,s′)
  - Implies soundness: for each R s.t. prover has significant probability of being able to convince, can extract y from the prover with comparable probability (using "rewinding", in a stand-alone setting)

Honest-Verifier ZK Proofs

- ZK PoK to prove equality of discrete logs for ((g,Y),(h,Z)), i.e., $Y = g^y$ and $Z = h^y$ [Chaum-Pederson]
  - Can be used to prove equality of two El Gamal encryptions (A,B) & (A′,B′) w.r.t public-key (g,Y): set (h,Z) := (A/A′,B/B′)
  - P→V: $(R,W) := (g^r, h^r)$
    V→P: x
    P→V: $s := xy + r$ (modulo order of the group)
    V checks: $g^s = Y^x R$ and $h^s = Z^x W$

    Two parallel executions of the previous proof, with same x and s (forcing same r, y)
- Special Soundness:
  - $g^s = Y^x R$ and $h^s = Z^x W \Rightarrow s = xy+r = xy'+r'$ where $R=g^r$, $Y=g^y$ and $W=h^{r'}$, $Z=h^{y'}$
  - If two accepting transcripts (R,W,x_1,s_1) and (R,W,x_2,s_2) $(x_1 \neq x_2)$, then $s_1 = x_1y + r = x_1y' + r'$ and $s_2 = x_2y + r = x_2y' + r'$. Then can find $y = y' = (s_1-s_2)/(x_1-x_2)$ (in a prime-order group).
- HVZK: simulation picks x, s first and sets $R=g^s/Y^x$, $W=h^s/Z^x$

Fiat-Shamir Heuristic

- Limitation of HVZK proofs: Do not guarantee ZK when verifier is actively corrupt
- In principle, can be fixed by implementing the verifier using "secure 2-party computation" (possibly implicitly)
- If verifier is a public-coin program (as in Chaum-Pederson) — i.e., simply picks random values and sends them — then, 2PC needed only to generate random coins
- Alternatively, **Fiat-Shamir Heuristic**: random coins from verifier defined as H(trans), where H is a random oracle and trans is the transcript of the proof so far (including the statement)
  - Also, importantly, removes need for interaction in the proof!

Fiat-Shamir Heuristic

- Fiat-Shamir Heuristic applied to the ZK Proof of knowledge of discrete log of $Y=g^y$
  - $P \rightarrow V$: $R := g^r$
    $V \rightarrow P$: x
    $P \rightarrow V$: $s := xy + r$
    V checks: $g^s = Y^x R$
  - $P \rightarrow V$: $R := g^r$
    $x = H(g,Y,R)$
    $s := xy + r$
    V checks: $g^s = Y^{H(g,Y,R)} R$
- Essentially, the prover is giving the proof "to the random oracle" and then reporting the transcript to the verifier
- To get an acceptable transcript, the prover must be able to convince the random oracle at least once (verifier checks that x matches what the oracle would have asked)
- But if the proof system has negligible soundness error, it cannot do that in polynomial number of attempts, unless the statement is correct

Example Application: VRF

- Verifiable Random Function
  - Is a PRF, but the (secret) key is sampled along with a public verification key PK
  - With SK, can not only compute $w = F_{SK}(q)$, but also generate a (non-interactive) proof that w is computed correctly; the proof can be verified using PK.
  - Even knowing PK, and after seeing proofs, for a new q, $F_{SK}(q)$ (without proof) should be pseudorandom, and also it should be infeasible to break the soundness of the proof system
- Several applications: To implement a lottery (e.g., in Algorand), to assign pseudonyms that can be revealed later (e.g., in NSEC5), ...
- We will see a simple VRF based on the computational Diffie-Hellman assumption, and in the random oracle model, using Fiat-Shamir heuristics

A PRF from RO

- $F_{SK}(q) = H(SK||q)$ is a PRF if H is a random oracle (and SK long enough)
  - Why? Infeasible to guess SK correctly. Without querying H on prefix SK, F_{SK} is identical to a truly random function.
- But no PK for this F and no way to prove correct evaluation
- Instead, let $(SK,PK) = (y, Y=g^y)$ and $F_y(q) = H(h^y)$, where $h=H'(q)$
  - H' maps the input q into a random element in the group
  - Still a PRF: infeasible to find h^y from (g,g^y,h), assuming CDH
  - Need to prove that $F_{SK}(q) = w$ (where SK corresponds to PK)
    - Proof should not reveal SK, or make it feasible to break pseudorandomness (for new inputs)

A VRF from RO

- $(SK,PK) = (y, Y=g^y)$ and $F_y(q) = H(h^y)$, where $h=H'(q)$
- If H is an R.O., then CDH ensures F is a PRF even given PK
- Proof that $F_y(Q) = w$:
  - Output Z s.t. H(Z) = w and give a ZK proof of equality of discrete logs for (g,Y) and (h,Z)
    - i.e., proving that $\exists y \;\; Y=g^y$ and $Z=h^y$
  - Non-interactive proof using the Fiat-Shamir heuristic applied to Chaum-Pederson protocol
- Does adding the proof hurt pseudorandomness/soundness?
  - Proof reveals nothing more than what (g,Y,h,Z) reveals
  - Which reveals nothing more than what (g,Y) reveals: (h,Z) can be simulated as (g^r,Y^r) since H' random oracle

Summary

- Fairly efficient ZK proofs systems exist for all NP properties
- Even more efficient HVZK proof systems for specialised problems like equality of discrete logs
- Fiat-Shamir heuristics can convert such protocols into **non-interactive** proofs secure against **actively corrupt verifiers** too (but in the Random Oracle model)
- An example application: VRF