Zero Knowledge Proofs (ctd.)

Lecture 14

Schnorr Signatures

Digital Signatures from Proof Systems

- Digital signatures can be seen as a proof of possession of a secret (signing) key, where the proof is tied with a message in a non-malleable fashion
  - Unforgeability: Seeing a proof tied to one message shouldn't leak the key, or enable one to give a proof of possessing it tied to another message
- It turns out that "proof systems" can indeed be turned into signature schemes
  - In the random oracle model, these form the basis of some of the most standard signature systems (DSA/ECDSA, EdDSA)
- Last time
  - Interactive proof systems
    - Eventually, to be useful as a digital signature, we will need a non-interactive proof.
  - Zero-Knowledge proof systems
    - When used for signatures, ZK ensures the signing key not leaked

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 $P \longleftrightarrow V$ for L has the following properties
  - Completeness: if $\exists w\ R(x,w)=1$, then $\Pr[P(x,w) \longleftrightarrow V(x) = 1] = 1$
  - Soundness: if $\nexists w\ R(x,w)=1$, then $\Pr[P^*(x) \longleftrightarrow V(x) = 1] = \text{negl}$ (for any P^*)

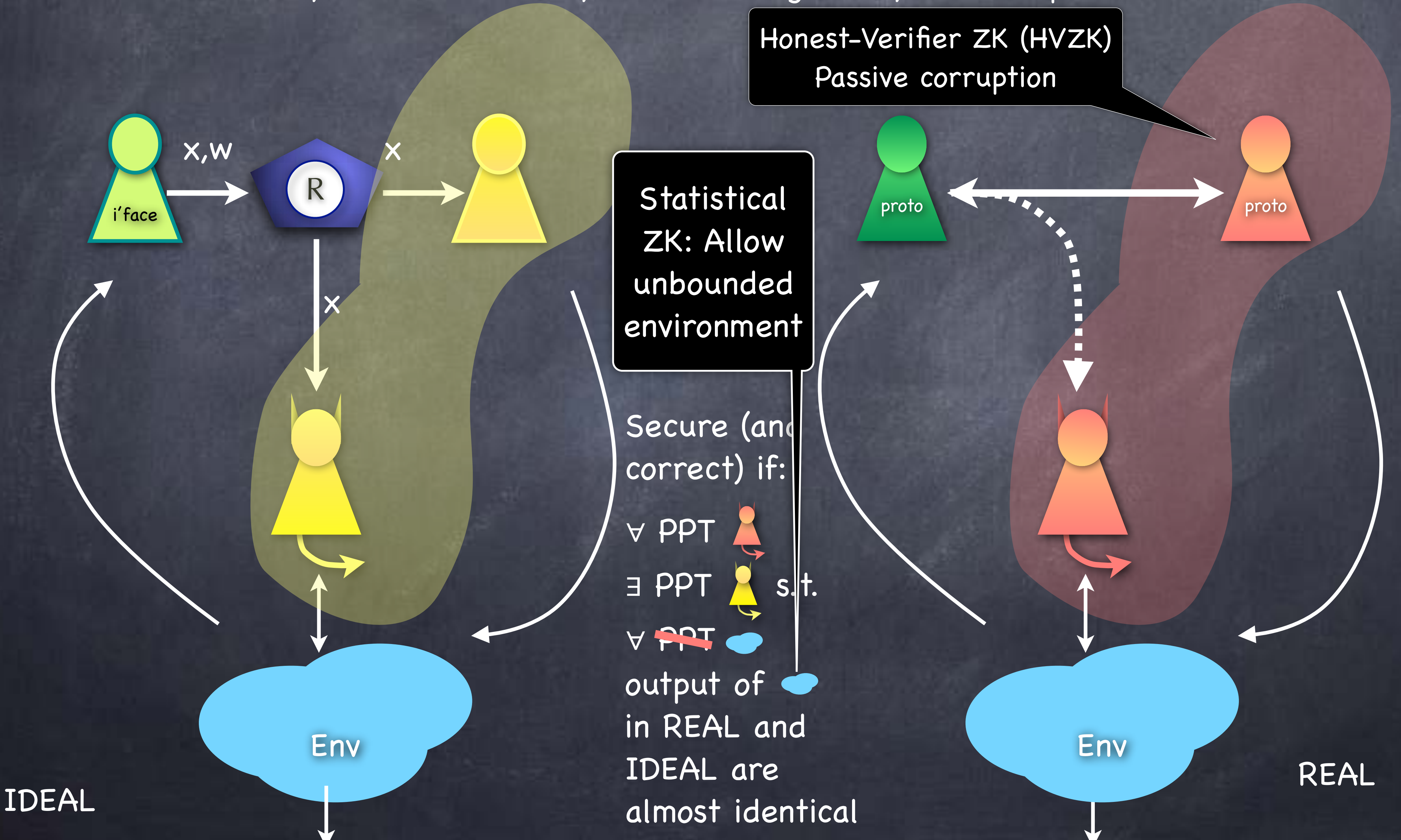
    ZK argument: soundness required only against PPT P^*

    - A stronger notion: Proof of Knowledge

    V learns nothing beyond the fact that x has the property
  - Zero-Knowledge: if $\exists w\ R(x,w)=1$, then view of the verifier in $P(x,w) \longleftrightarrow V(x)$ 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., $\text{View}_{V^*}(P(x,w) \longleftrightarrow V^*(x)) \approx S(x)$

ZK Property

Classical definition uses simulation only when **receiver is corrupt**;
Also uses only standalone security: Environment gets only a transcript at the end



Proof of Knowledge

- In a Proof of Knowledge, an adversary that gives valid proofs (with significant probability), "can give" a witness (i.e., can be extracted from it)
- A ZK Argument of Knowledge of discrete log of $Y=g^y$ (in a prime-order group G; say $|G|=N$)

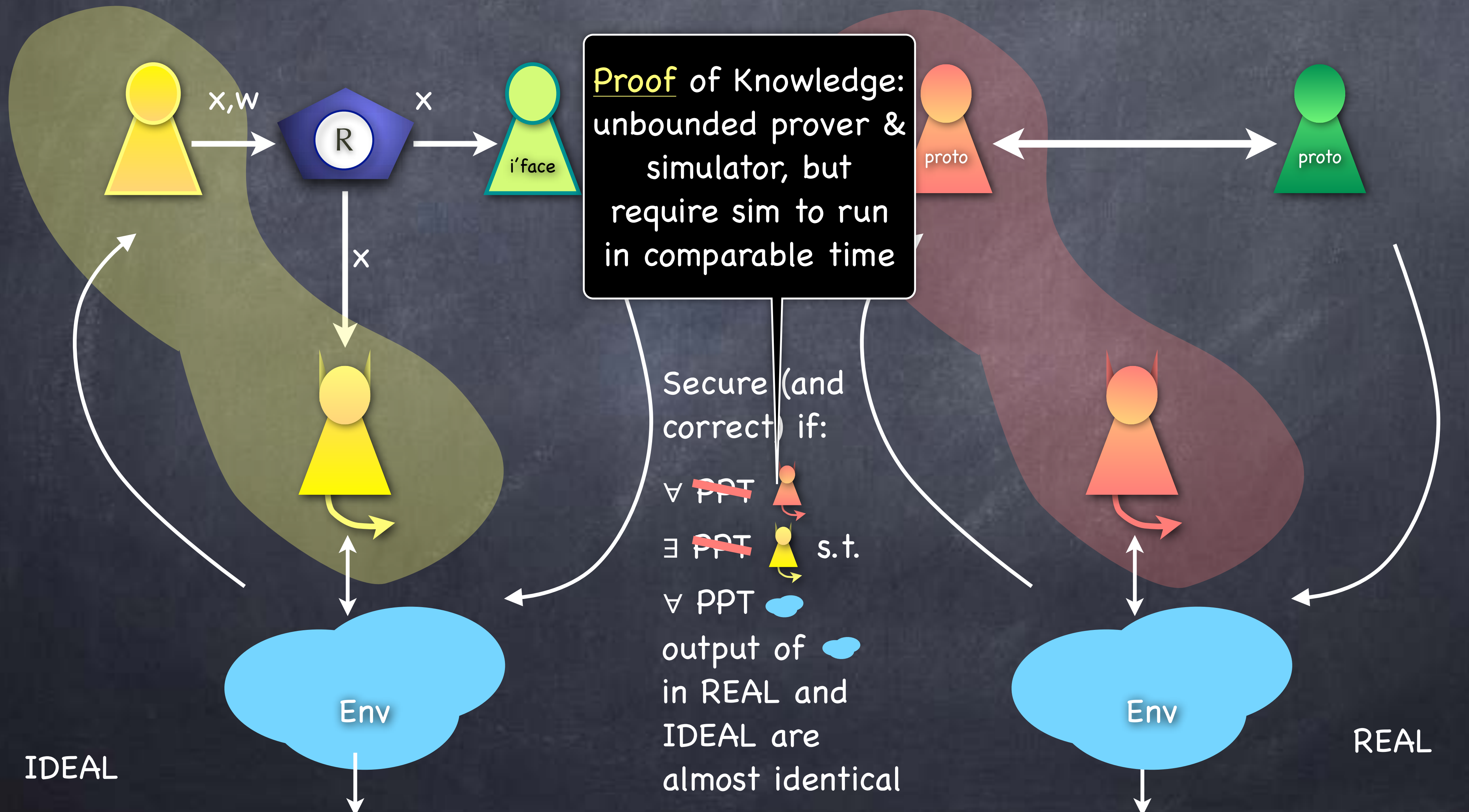
  The term "Proof" is used to indicate that the corrupt prover could be computationally unbounded.

  Not the case here.

  - $P \rightarrow V$: $R := g^r$ for a random r modulo N
    $V \rightarrow P$: x random modulo N
    $P \rightarrow V$: $s := xy + r$ modulo N
    V checks: $g^s = Y^x R$
  - Knowledge-Soundness:
    - 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 $s_1 = x_1 y + r$ and $s_2 = x_2 y + r$, then can solve for y (in a prime-order group)
  - ZK: simulation picks s, x first and sets $R = g^s/Y^x$

Knowledge Soundness

- Require simulation also when prover is corrupt
  - Then simulator is a witness extractor
  - With all entities PPT, corresponds to Argument of Soundness



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 knowledge-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)) (in a prime-order group). 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')
  - $P \rightarrow V$: $(R,W) := (g^r, h^r)$
    $V \rightarrow P$: x
    $P \rightarrow V$: $s := xy + r$ (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 $y = y'= (s_1-s_2)/(x_1-x_2)$.
- 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
- If verifier is a public-coin program (as in Chaum-Pederson) — i.e., simply picks random values and sends them — then, need only to generate trustworthy random coins
- **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
  - Note: In the standard setting, ZK proofs need to be interactive; else a corrupt prover can give simulated proofs!

Fiat-Shamir Heuristic

- Example: 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 gives the proof "to the random oracle" and then reports the transcript to the verifier (who also checks x)
- To get an acceptable transcript, the prover must be able to convince the random oracle at least once
- But if the proof system has negligible soundness error, can't do it in polynomial number of attempts, unless the statement is correct
- Further, special soundness still yields knowledge soundness (via an argument called "Forking Lemma")

Fiat-Shamir Heuristic

- Zero-Knowledge property still holds (assuming an honest prover is unlikely to use the same partial transcript in independent proofs)
- Intuitively, if the partial transcript is fresh, its hash is indeed a uniformly random string, just like an honest verifier would have sent
- Formally, a simulator which programmes the hash function
  - First generate a simulated transcript, say (R,x,s) and then program the random oracle so that H(stmt||R) = x
    - Note: stmt||R assumed to be fresh. But the original proof system will anyway need this to avoid the verifier being able to run a knowledge extractor.

Schnorr Signature

- From a ZK Argument of knowledge of discrete log of $Y=g^y$ (in a prime-order group)
  - P→V: $R := g^r$
    V→P: x
    P→V: $s := xy + r$

  V checks: $g^s = Y^x R$

Fiat-Shamir heuristic

(hash m too)

Schnorr signature

$(SK,VK) = (y,(g,Y))$ where $Y=g^y$

Signature = (R,s) where

Pick $R := g^r$

Let $x = H(m||VK||R)$

Let $s := xy + r$

Verification: $g^s = Y^{H(m||VK||R)} R$

- Hashed "transcript" includes the message as well now
- By ZK of the proof system, can simulate a signing oracle (without knowing signing key)
  - First simulate a transcript (R,x,s) (Recall: pick x, s first, then set $R=g^s/Y^x$). Then program $H(m||VK||R) = x$
- By special soundness (and forking lemma) a non-negligible advantage, using polynomial queries to the RO, can be converted into similar advantage for solving DL

Schnorr Signature

- EdDSA is based on Schnorr Signature
  - Uses a particular group based on "Edwards curves"
  - Instead of a random nonce r, sets it to be a hash of message and (part of) private key
    - The nonce should be unpredictable (not queried to the random oracle previously by the adversary), for the ZK simulation
- There is a (somewhat) similar signature scheme called El Gamal Signature
  - Standards DSA and ECDSA are based on it

Schnorr signature

$(SK,VK) = (y,(g,Y))$ where $Y=g^y$

Signature $= (R,s)$ where

Pick $R := g^r$

Let $x = H(m||VK||R)$

Let $s := xy + r$

Verification: $g^s = Y^{H(m||VK||R)} R$

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)
- Security of EdDSA (Schnorr signature) is directly based on this. DSA/ECDSA are similar schemes

These, as well as RSA signatures, all rely on the Random Oracle Model