Non-Interactive ZK and Fiat-Shamir

April 16, 2018

Logistics

- Problem set 1: graded by Thursday (drop deadline Friday)
  - * average time 15-25 hours (355 = 255+100)
  - * target time: 10-15 hours
  - * Favorite problem: 2/3
  - * Least fav: Problem 4
  - ⇒ Please submit feedback

- Solutions will be on Piazza
- HW2 out now
  - ↳ should be somewhat easier
- Ask if unclear / underspecified / wrong! We try to be correct but we fail often! :)

Plan

- Recap: Proofs of Knowledge
- Non-interactive ZK
- Fiat-Shamir heuristic
  - * Idea
- Applications
  - * Schnorr sig and DSA
  - * RFS protocol

Recap: Proofs of Knowledge

A language $\mathcal{L}$ is in NP if $\exists$ poly time VRF relation R s.t.

$$x \in \mathcal{L} \iff \exists\ w \in \{0,1\}^{\text{poly}(|x|)} \text{ s.t. } R(x,w) = 1$$

x = "instance" or "statement"
w = "witness"

Eg: ϕ is SAT formula
$\vec{a}$ is assignment to vars
$R(\phi, \vec{a}) = \begin{cases} 1 & \text{if } \phi(\vec{a}) = \text{satisfied} \\ 0 & \text{o.w.} \end{cases}$

Intuition behind proof versus PoK

Proof	PoK
"____ is true"	"I 'know' why ____ is true"
N is product of two primes	I 'know' p, q s.t. $N = pq$
$\phi \in$ SAT	I know satisfying assignment for ϕ
$x \in \underbrace{\mathcal{L}(R)}_{\text{NP lang}}$	I know w s.t. $R(x,w) = 1$

A ZK proof for NP language $\mathcal{L}(R)$ is a PoK if has

1) Completeness
2) knowledge
3) ZK

↳ Idea: If (possibly malicious) P^* convinces honest V that $x \in \mathcal{L}(R)$ w.p. $\geq \frac{1}{2}$, then exist ppt "extractor" E s.t.

$$\Pr\left[w \leftarrow E^{P^*} ; R(x,w) = 1\right] \geq \frac{1}{2} - \varepsilon$$

⇒ By interrogating P^*, can extract a witness from it
↳ to convince V, P^* must be able to produce w

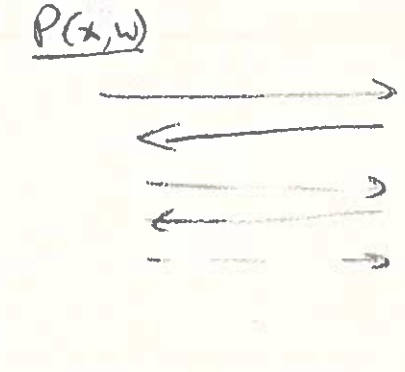
Extractor can "rewind" P^*
↳ Why? P^* is just a program... can run and re-run it
↳ In real interaction, cannot rewind P^* ⇒ Can't break ZK
↳ Fact that time goes forward is what makes ZK possible

Q: If you had a time machine, could you break ZK?
↳ Aaronson and Watrous paper on CTCs, VM reset attacks...

Non-Interactive ZK

↳ See recent paper from Sam & David

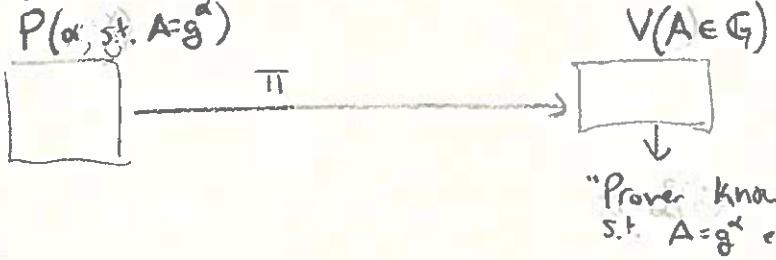
So far, we have looked at interactive protocols for NP relations $R(x,w)$



⇒ We saw that these IPs can provide ZK — V learns only that $x \in L$

In practice, it would be nice to have non-interactive ZK proofs of knowledge

$\mathbb{G} = \langle g \rangle$ is group in which dlog hard

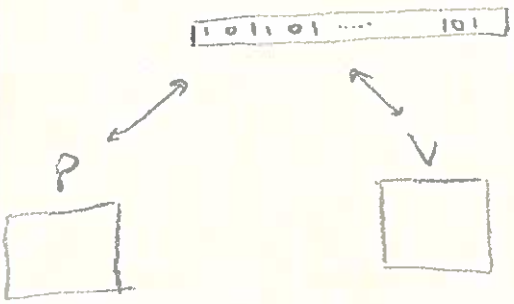


Turns out, such "one-shot" ZK protocols only exist for "easy" (BPP) languages, in the standard model. (Why?)

When faced w/ impossibility, what do we do?

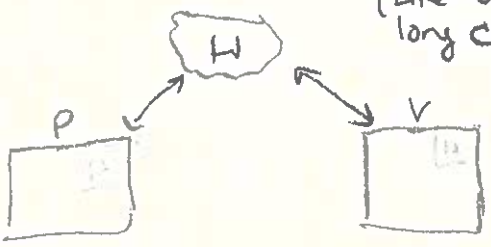
1) Introduce crazy assumptions
2) Change the model ←

CRS model



Random-Oracle Model

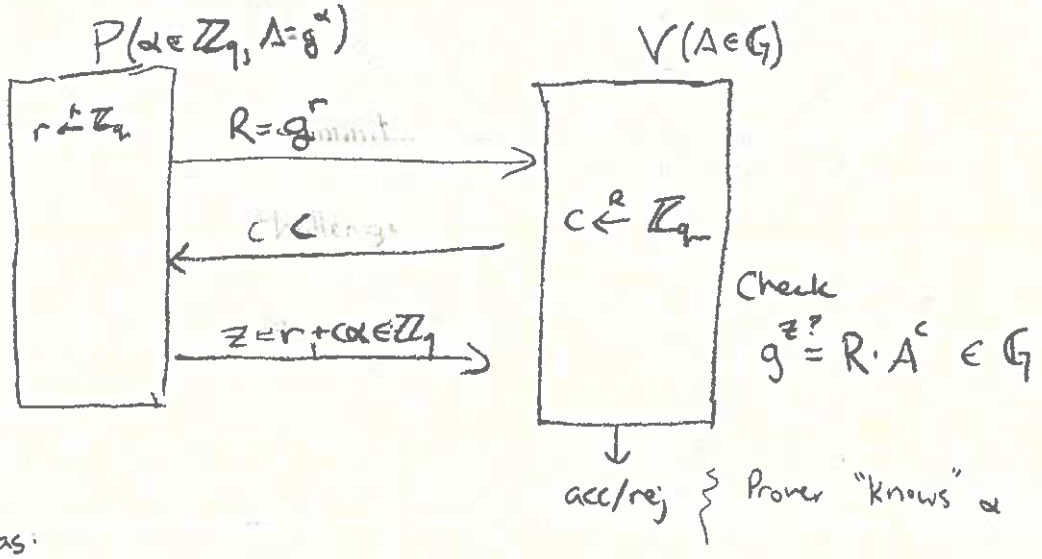
(Like an exponentially long CRS)



A Good Idea: Fiat-Shamir Heuristic

Convert Sigma Protocol for language $\mathcal{L}(R)$ $\Rightarrow$ NIZKPoK for $\mathcal{L}(R)$ in R.O.M.

Recall Schnorr's protocol for proving knowledge of dlog $A = g^\alpha \in \mathbb{G}$, $|\mathbb{G}| = q$



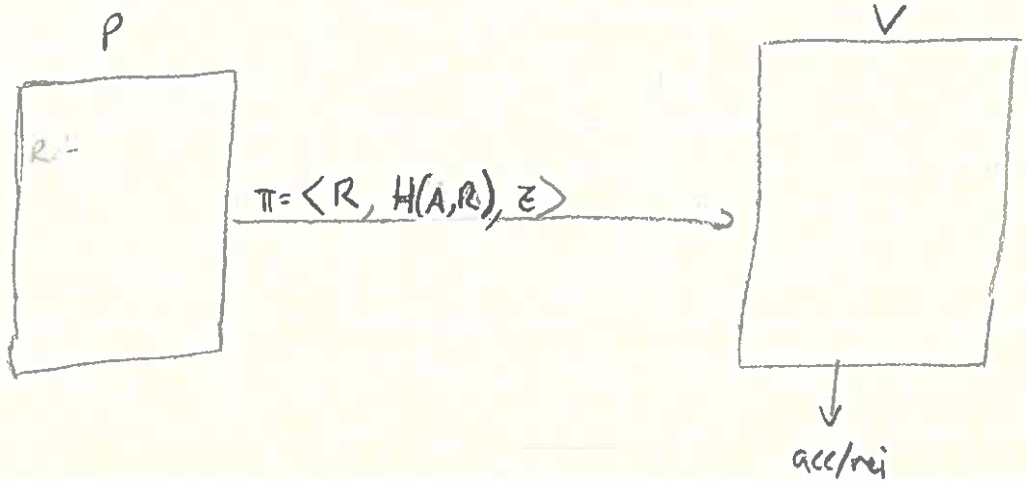
Has:

1) Completeness
2) Knowledge
3) HVZK

Notice that V (1) sends only random value to P } "Public coin"
(2) Has no secret state.

Idea of Fiat-Shamir: * Replace V w/ a Hash fn $H: \{0,1\}^* \to \mathbb{Z}_q$
* Security in R.O.M.

$(c \leftarrow H(A, R) \in \mathbb{Z}_q)$



Fiat-Shamir & Schnorr

We need to show (informally)

1. Completeness ✓
2. Knowledge → Use "rewinding" (Boneh-Shoup 19.1-19.2)
3. HVZK ✓

Idea:
* Run P^*, get (R, c, z)
* Rewind P^* and run again
* When P^* queries RO, give different answer
  ↳ (R, c', z')
* Can solve for dlog α using linear algebra

⇒ Some technical subtleties

Schnorr Signature Scheme ($\mathbb{G}$ is group of prime order q, dlog hard)

$\text{Gen}() \to \begin{cases} \alpha \xleftarrow{R} \mathbb{Z}_q \\ sk = \alpha, \; pk = g^\alpha \in \mathbb{G} \end{cases}$

$\text{Sign}(sk = \alpha, m) \begin{cases} r \xleftarrow{R} \mathbb{Z}_q; \; R = g^r \in \mathbb{G} \\ c \leftarrow H(pk, R, m) \\ z \leftarrow r + c\alpha \in \mathbb{Z}_q \\ \text{output } (R, c, z) \end{cases}$

Can compress & send only (c, z)

$\text{Verify}(pk = A, (R, c, z), m) \begin{cases} c \stackrel{?}{=} H(A, R, m) \\ R \stackrel{?}{=} g^z A^{-c} \in \mathbb{G} \end{cases}$

Proof of security is not immediate.... see book

Implementation note: Knowledge error is $1/|\mathcal{C}|$ & challenge $c \in \{1, \ldots, \ell\}$
↳ Use 128-bit challenge in practice, while $q \approx 256$ bits

Schnorr

Idea: * Take an interactive ZK protocol for knowledge of discrete l.g
  * Compile into NIZK using ROM
  * Stick msg into ROM to make it a sig scheme.

In theory, can use any NP language as basis for a sig scheme in this way

Graph isomorphism, 3-Col, factoring, ...
↳ Why not NP-complete problem? e.g. 3SAT?

- Schnorr Sigs have beautiful theoretical basis
- In real life (TLS, Bitcoin, etc), we all use DSA/ECDSA → same idea, but worse

Why? Patents! (expired 2008)

Practical Note: Implementers beware!

The prover/signer in Schnorr signatures is randomized.
↳ Indeed, Prover in all ZK protocols is randomized, unless language L in question $\in BPP$. (Why?)

Prover must use fresh randomness for each signature!

What happens if not?

$$(g^r,\ c = H(A,R,m),\ z = r + c\alpha)$$
$$(g^{r'},\ c' = H(A,R,m'),\ z' = r + c'\alpha)$$

$$z - z' = (c - c')\alpha \in \mathbb{Z}_q$$
$$\frac{z - z'}{c - c'} = \alpha \in \mathbb{Z}_q$$
↳ Get signer's secret key!

⇒ This has led to ≥ 20 practical attacks!
Playstation 3 hack, Bitcoin theft, ...

↳ Many attacks even if random choice r is somewhat biased

Deterministic Schnorr

To avoid attacks, can sign using PRF $F(k, m) \to \mathbb{Z}_q$

secret key = (α, k)

Sign(sk, m): $r \leftarrow F(k, m)$ ← Derive secret "random" value using PRF.

$R \leftarrow g^r \in \mathbb{G}$

⋮

Practical Note:

ECDSA signatures are used everywhere

(c, z)

c: 128-bit challenge

z: integer in $\mathbb{Z}_q$ } 384 bits for 128-bit security

RSA-FDH signature ≈ 3072 bits " " "

Feige-Fiat-Shamir Protocol

$N = pq$ is RSA modulus

Prove knowledge of square root mod N of $A \in \mathbb{Z}_N^*$

$P(A, a \text{ s.t. } A = a^2 \bmod N)$ $\quad$ $V(A \in \mathbb{Z}_N^*)$

$r \xleftarrow{R} \mathbb{Z}_N^*$

$R = r^2 \bmod N$ $\longrightarrow$

$\longleftarrow$ $c \in \{0,1\}$ $\quad$ $c \xleftarrow{R} \{0,1\}$

$z \leftarrow r \cdot a^c \bmod N$ $\quad$ z $\longrightarrow$

$z^2 \stackrel{?}{=} R \cdot A^c$

Complete ✓

Knowledge $(R, 0, z)$ s.t. $z^2 = R$

$(R, 1, z')$ s.t. $(z')^2 = R \cdot A$

$\Rightarrow (z'/z)^2 = A$

$\hookrightarrow z'/z$ is root

ZK Need to show $\forall\, V^*$ $\exists$ ppt simulator Sim that simulates V^*'s view

Sim$(A \in \mathbb{Z}_N^*)$ {

$c' \xleftarrow{R} \{0,1\}$

$z \xleftarrow{R} \mathbb{Z}_N^*$

$R \leftarrow z^2 / A^{c'}$

run $V^*(R) \to c$

if $c \neq c'$: abort

else : output (R, c, z)

$\Rightarrow$ Turns out, computing sq roots mod N is as hard as factoring N, so this is also Σ protocol for knowledge of factors of N (with some extra work)