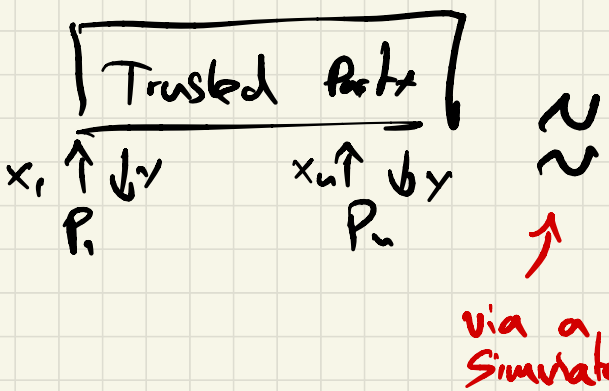
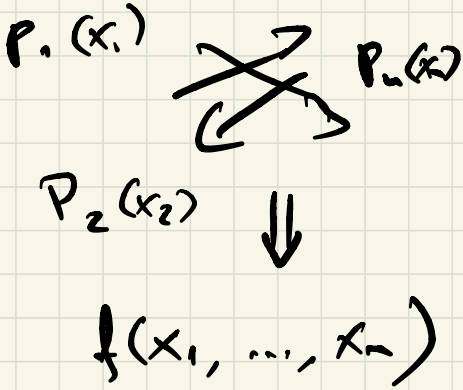
Privacy Enhancing Technologies

3. Zero-knowledge Proofs

Recap on MPC





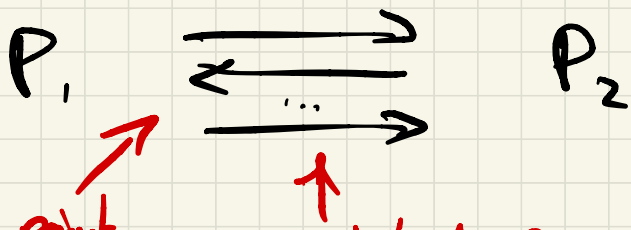
A protocol that is secure (for any function) if A is semi-honest

↳ not true in practice...

What about malicious security? what can A do?

* Abort at any point ⇒ Fairness issue
  ↳ all parties get y, or none of them

Fairness is impossible !



at this point P_1 knows γ

what if P_1 doesn't send this msg

if protocol is fair P_2 should also get γ

$\Rightarrow$ the last msg is redundant

$\Rightarrow$ recurse, all msgs are redundant

MPC properties:

* privacy & correctness

this is for free in semi-honest setting

↑ in malicious setting this is harder

* Fairness

* Input independence

↳ A can't choose it's input to depend on an honest parties'

Why not use MPC for everything?

→ We have a protocol to compute any function privately

↳ parties learn y and nothing else

→ We might want:

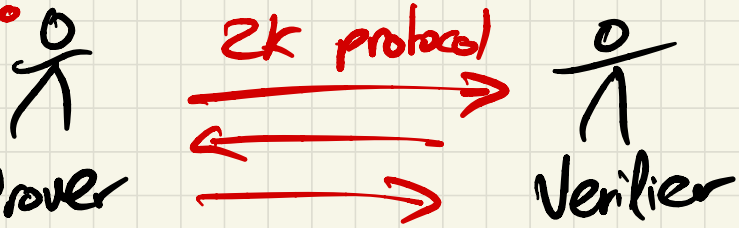
- more efficiency
- less interaction
- "more" privacy (revealing y exactly might be bad)

Zero-knowledge proof systems

⇒ MPC for 2 parties with one-sided privacy / correctness

P knows sat. assignment x of SAT formula φ

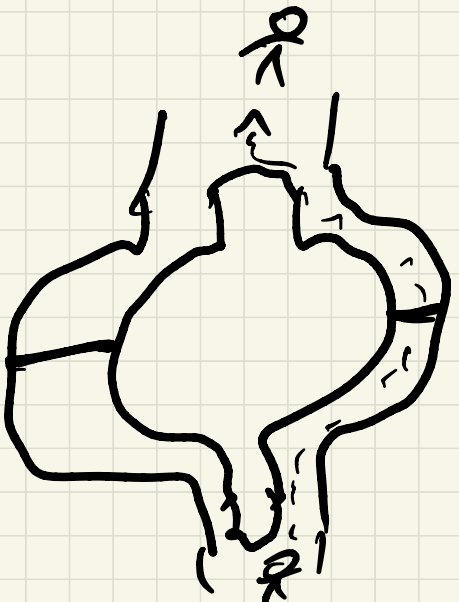
$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge \ldots$



↳ convinced that φ is satisfiable

↳ learn nothing about x

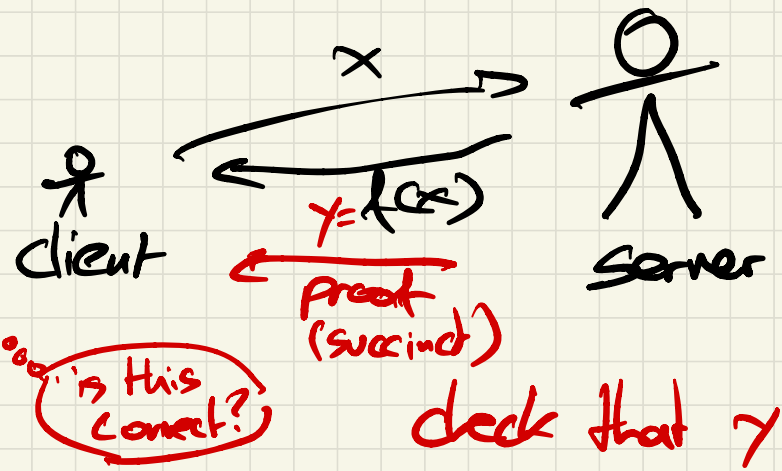
Example



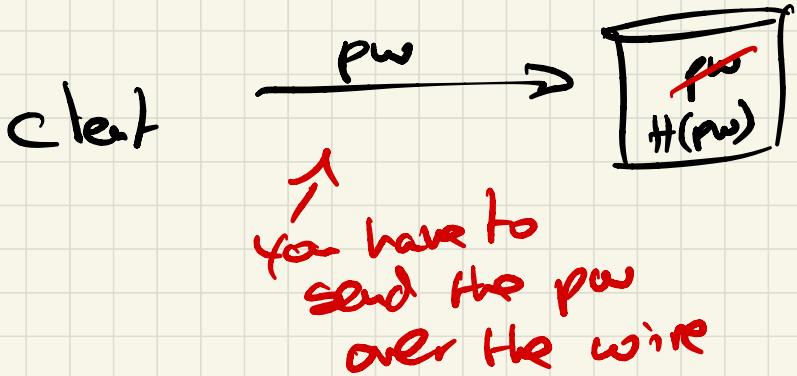
Why ZK?

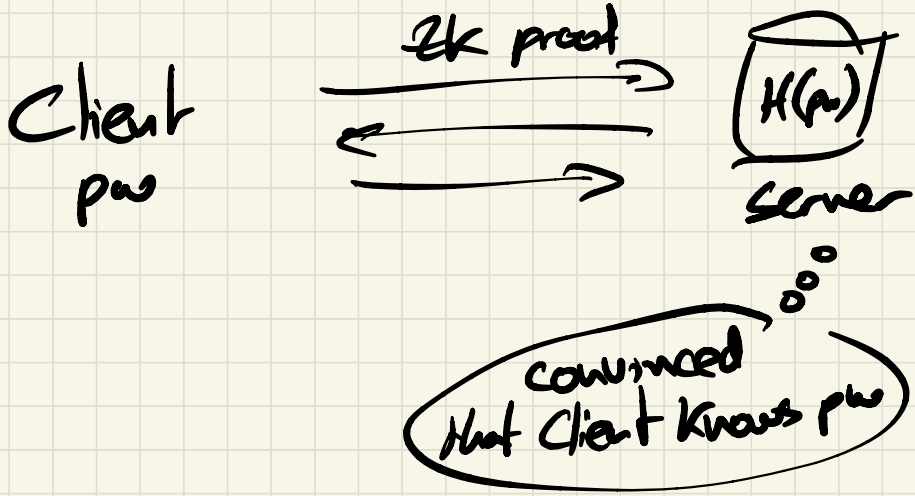
Applications

- Verifiable computation



- Verifying Passwords





* Cryptocurrencies
  - private payments
  - outsourcing work

* Malicious MPC

Semi-honest protocol + ZK proofs

↳ when P_i sends msg m, they prove (in ZK) that this is the right msg to send

Defining ZK Proofs

Circuits: $C : \mathbb{F}^n \times \mathbb{F}^u \to \mathbb{F}$ (output)

"public input" $\mathbb{F}^n$ × "witness" $\mathbb{F}^u$

we say C is **satisfiable** on input x

if $\exists\, w$ such that $C(x, w) = 0$

ZK means informally that the verifier learns that $C(x, w) = 0$ without learning w

Def A ZK proof system for a family of circuits $\mathbb{C}$ is denoted as $\langle P, V \rangle(C, x)$ satisfies:

1) **Completeness**: for all $C \in \mathbb{C}$, $x \in \mathbb{F}^m$ where C is sat. on x.

$$\Pr\left[\langle P, V \rangle(C, x) = \text{"accept"}\right] \geq 2/3$$

2) **Soundness**: $\forall C \in \mathbb{C}$, $x \in \mathbb{F}^m$ where C is **not** sat. on x and all **malicious** provers P^*

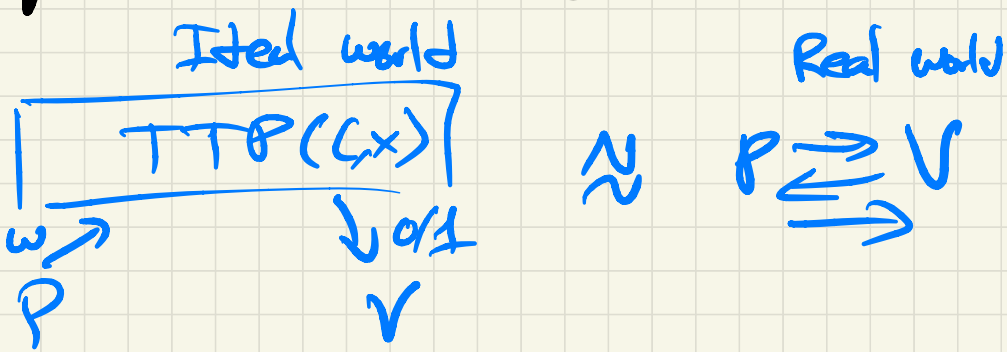
$$\Pr\left[\langle P^*, V \rangle(C, x) = \text{"accept"}\right] \leq 1/3$$

3) Zero-knowledge $\forall$ malicious V^*

Here exists a simulator Sim_{V^*} s.t.

$\forall C \in \mathcal{C}, x \in \mathbb{F}^{k_n}$ where C is sat. on x

$$View_{V^*}[\langle P, V^* \rangle(C, x)] \approx Sim_{V^*}(C, x)$$



Discussion

completeness error
soundness error

* why $2/3$, $1/3$?
  ↳ just repeat the protocol λ times
  (security parameter)

* both the prover & verifier can be malicious
  - prover → soundness
  - verifier → zk, "privacy"

Additional properties

1) "Proof of knowledge"

$$C(x, w) = 0$$

the verifier learns that there exists a witness w

not the same as "Prover knows the witness"

Password example:

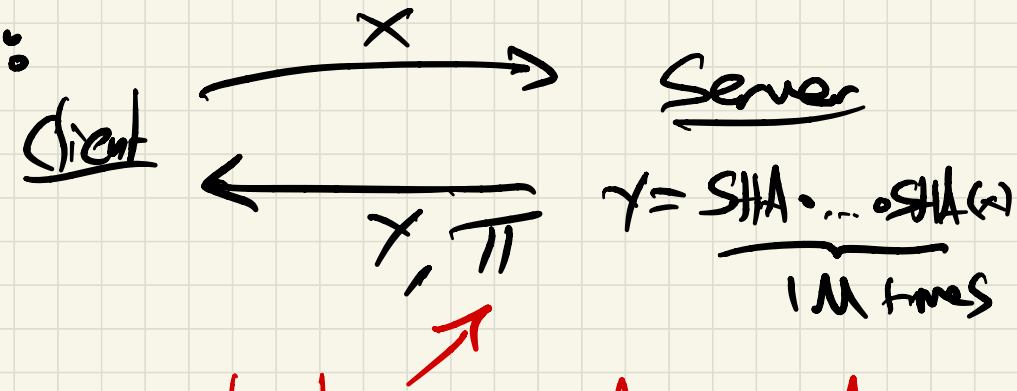
prover	verifier
pw	$y = H(pw)$

2) Non-interactivity
3) Succinctness

} SNARGs

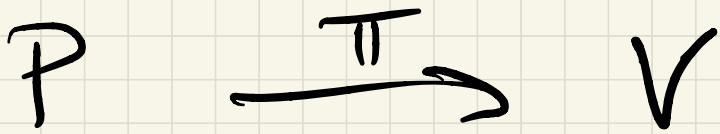
succinct non-interactive argument

Ex:



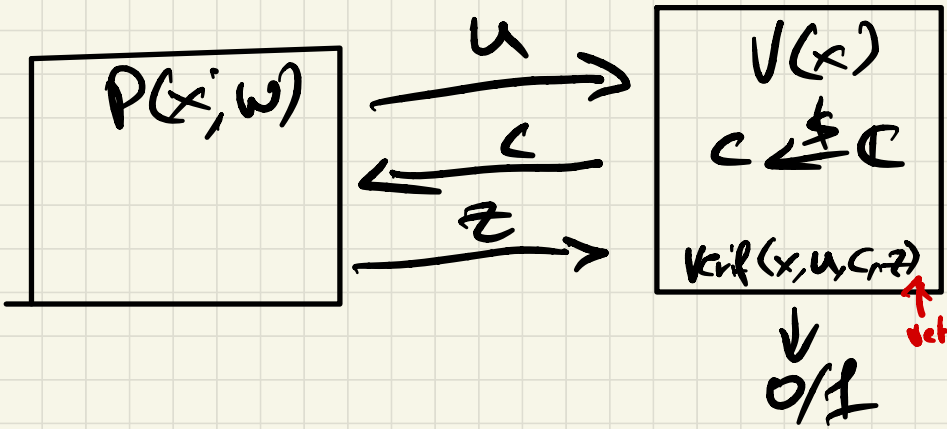
- client can verify proof π much faster than computing y
- we want the proof π to be a single msg

* Non-interactivity (NIZKs)



Fiat-Shamir transform

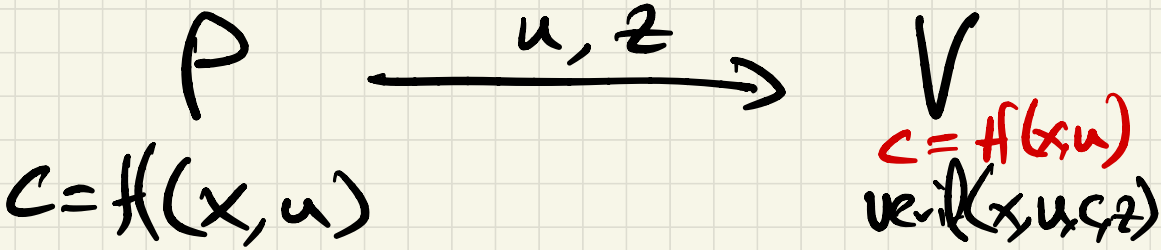
"Sigma Protocols" Σ



Idea: have the prover choose c themselves

Issue: what if P cheats?

How? Random Oracles



- to prove security assume H is a random function
- in practice, use a real hash function

Not on soundness:

Here we want soundness error to be negligible

* Succinctness

- short proofs : $|\pi| = \text{poly}\log(|C|)$
- efficient verif. : time to verify π
  $O(|x|, \text{poly}\log(|C|))$

Non-trivial : $P \xrightarrow{\omega} V$

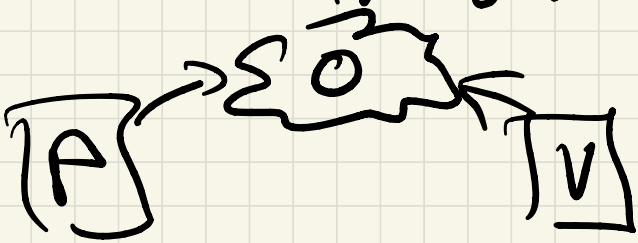
this is not succinct!

- $|\omega| = O(|C|)$
- verifier needs to run $C(x, \omega)$

Building SNARGs

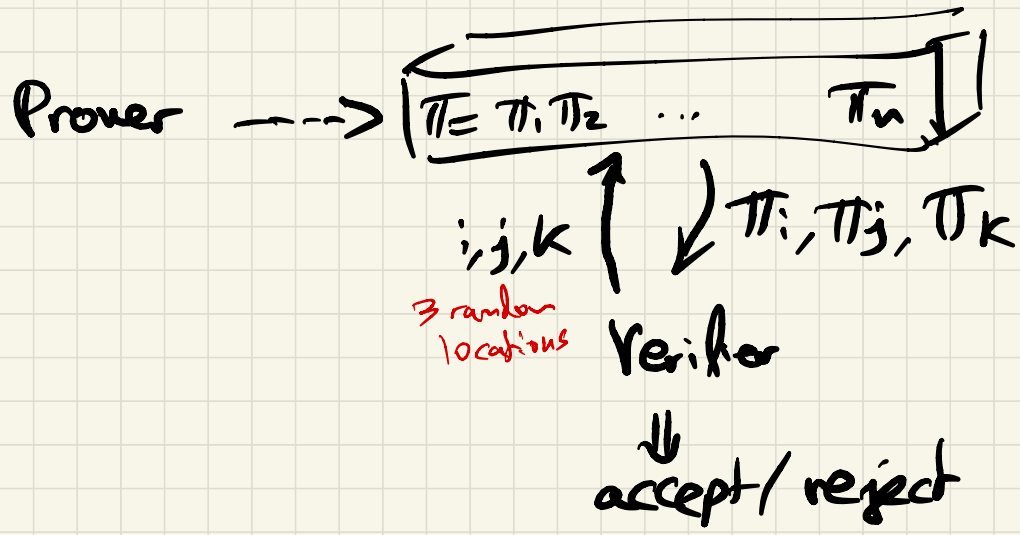
Blueprint

1) Build a proof system that is succinct in some "weird" model without cryptography



2) "emulate" this weird model using cryptography

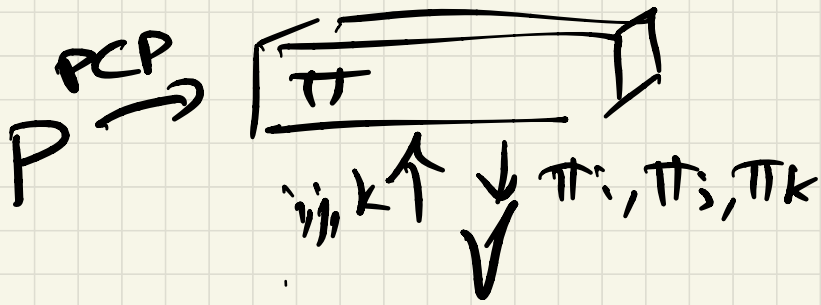
The Box game



⚠ Amazing result: This is possible

PCP Theorem: all NP languages have a proof system where $a(|\pi|) = poly(|x|)$ and V reads 3 bits of the proof π

↳ soundness error $1/2$

1) 

Issue: this box doesn't exist

Sol: Use cryptography

2) "Commitment scheme"

Special commitment scheme:

Vector comm. scheme

The box game is a "interactive oracle proof" IOP