MIT 6.875 & Berkeley CS276

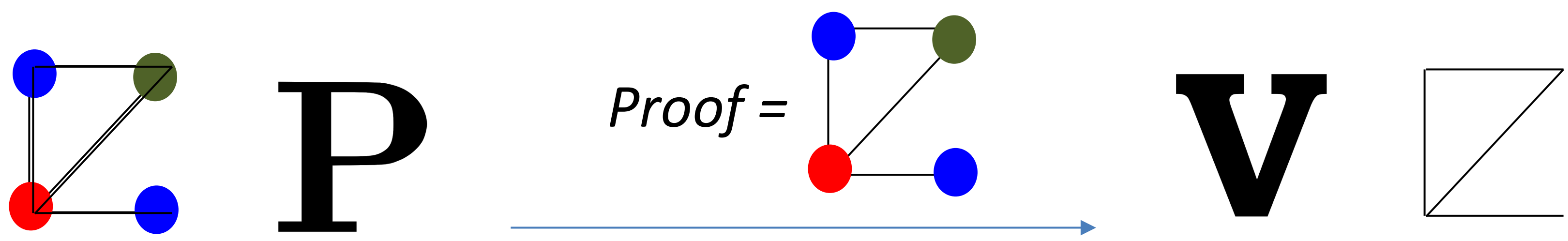
Foundations of Cryptography
Lecture 16

Today:
Non-Interactive Zero-Knowledge (NIZK)

In Two Days:
An Application of NIZK

NP Proofs

For the NP-complete problem of graph 3-coloring

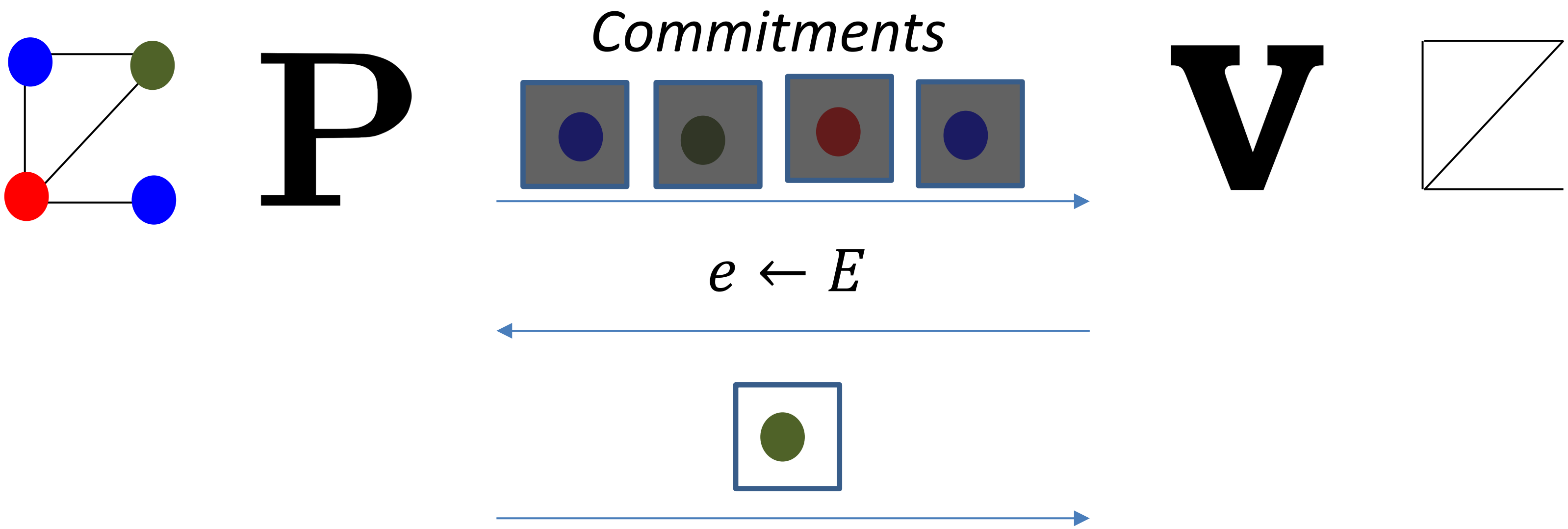


Prover P has a witness, the 3-coloring of G

Verifier V checks:
(a) only 3 colors are used &
(b) any two vertices connected by an edge are colored differently.

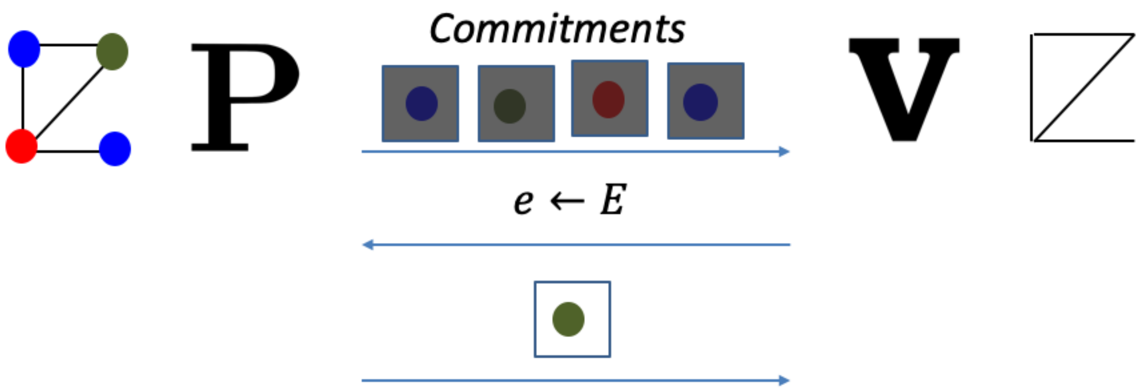
Zero-Knowledge (Interactive) Proof

Because NP proofs reveal too much



Zero-Knowledge (Interactive) Proof

Because NP proofs reveal too much



1. Completeness: For every G ∈ 3COL, V accepts P's proof.

2. Soundness: For every $G \notin$ 3COL and any cheating P^*, V rejects P^*'s proof with probability $\geq 1 - \text{neg}(n)$

3. Zero Knowledge: For every cheating V^*, there is a PPT simulator S such that for every G ∈ 3COL, S ***simulates the view*** of V^*.

TODAY:

Can we make proofs non-interactive again?

Why?

1. *V does not need to be online during the proof process.*
2. *Proofs are not ephemeral, can stay into the future.*

TODO:

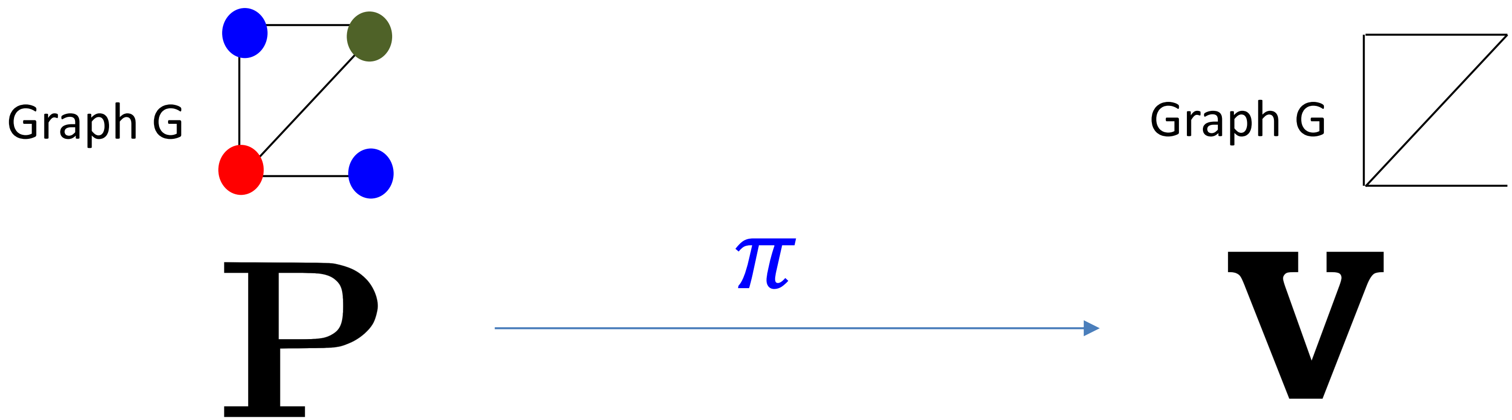
Can we make proofs non-interactive again?

YES, WE CAN!

NO!

Non-Interactive ZK is Impossible

Suppose there *were* an NIZK proof system for 3COL.

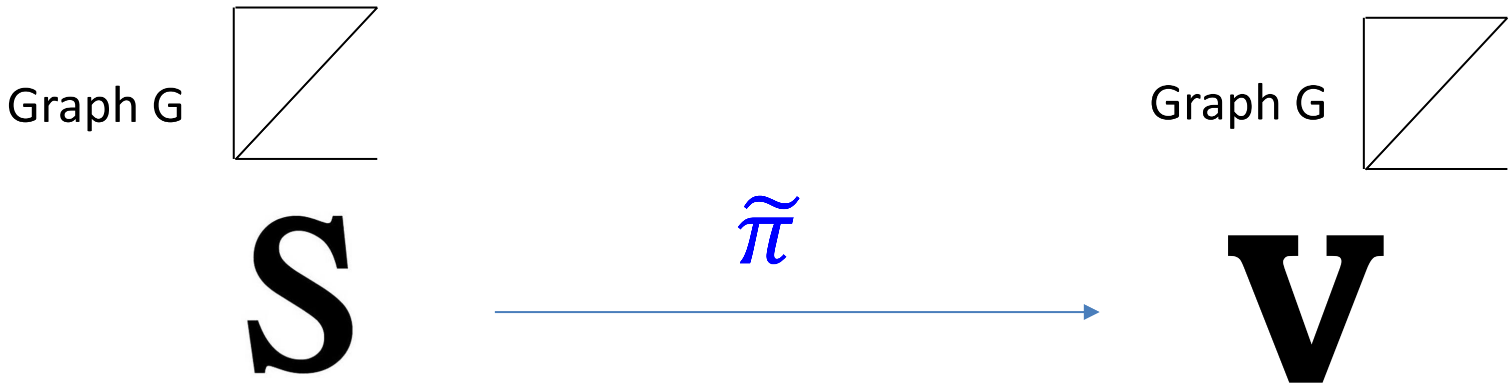


Step 1. When G is in 3COL, V accepts the proof π.

(Completeness)

Non-Interactive ZK is Impossible

Suppose there *were* an NIZK proof system for 3COL.

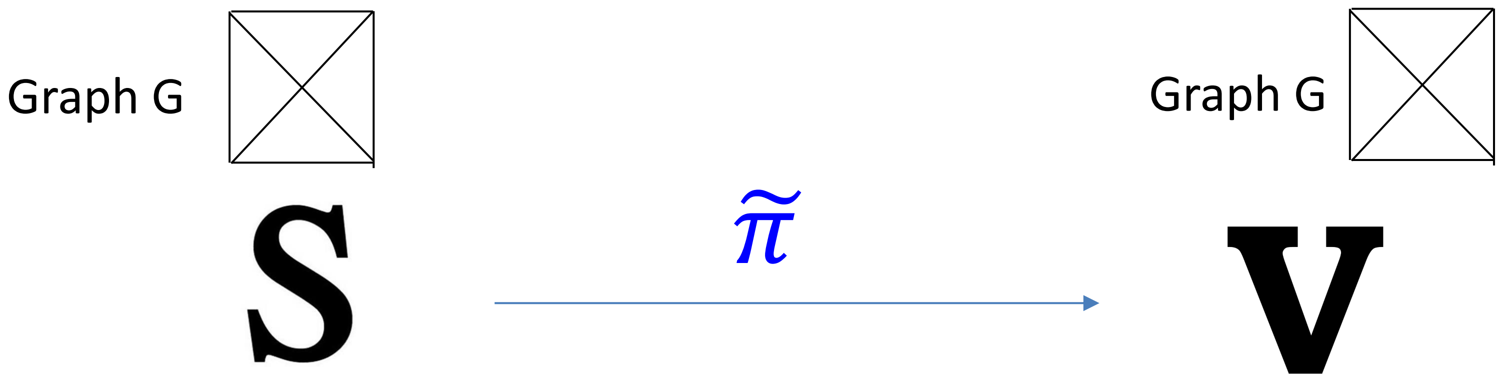


Step 2. **PPT** Simulator S, **given only G in 3COL**, produces an indistinguishable proof $\tilde{\pi}$ (Zero Knowledge).

In particular, V accepts $\tilde{\pi}$.

Non-Interactive ZK is Impossible

Suppose there *were* an NIZK proof system for 3COL.

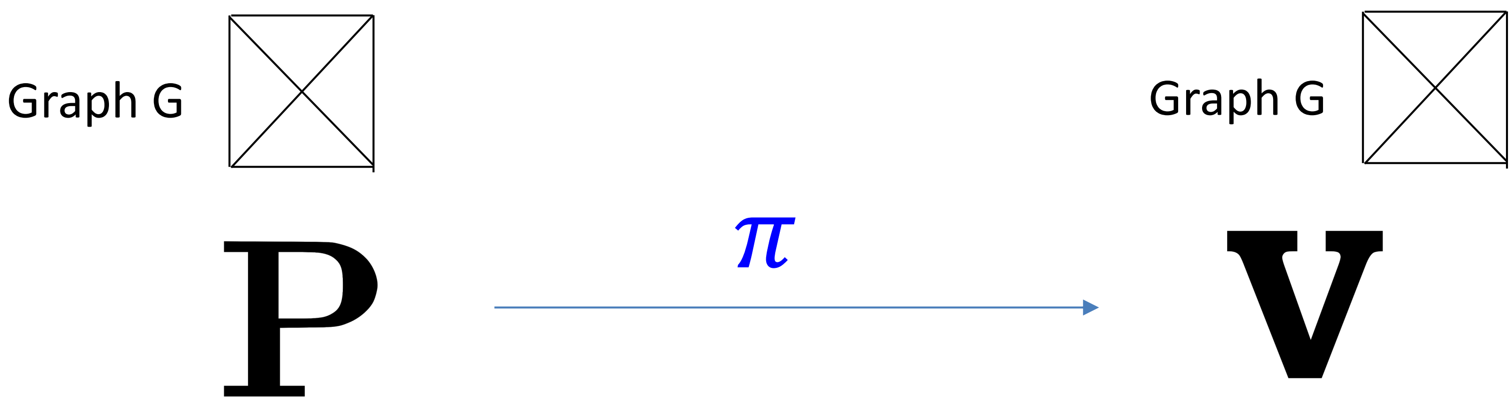


Step 3. Imagine running the Simulator S on a $G \notin$ 3COL.
It produces a proof $\tilde{\pi}$ which the verifier still accepts!

(WHY?! Because S and V are PPT. They together cannot tell if the input graph is 3COL or not)

Non-Interactive ZK is Impossible

Suppose there *were* an NIZK proof system for 3COL.



Step 4. **Therefore, S is a cheating prover!**

Produces a proof for a $G \notin$ 3COL that the verifier nevertheless accepts.

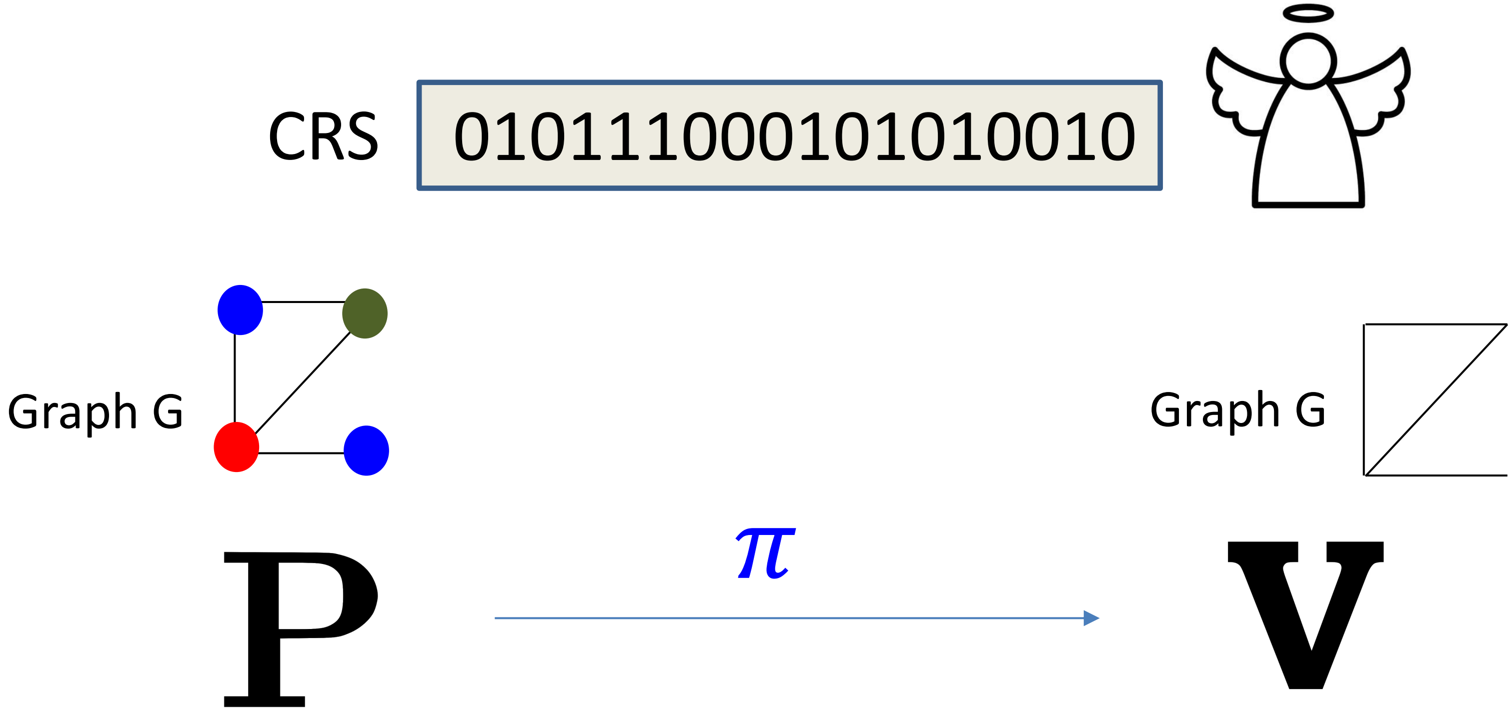
Ergo, the proof system is NOT SOUND!

∎

THE END

Or, is it?

Enter: The Common Random String

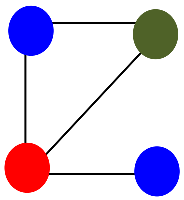


Enter: The Common *Reference* String

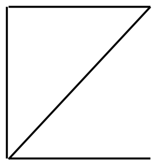
$$CRS \leftarrow D$$

(e.g., CRS = product of two primes)

Graph G



Graph G

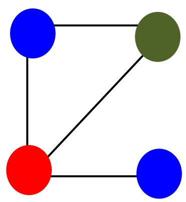


P $\xrightarrow{\pi}$ **V**

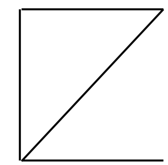
NIZK in the CRS Model

CRS 010111000101010010

Graph G



Graph G



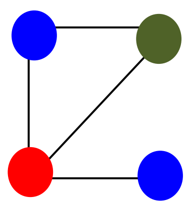
P π → **V**

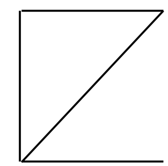
1. Completeness: For every $G \in$ 3COL, V accepts P’s proof.

2. Soundness: For every $G \notin$ 3COL and any “proof” π^*, $V(CRS, \pi^*)$ accepts with probability $\leq \text{neg}(n)$

NIZK in the CRS Model

CRS 0101110001010100 10

Graph G 

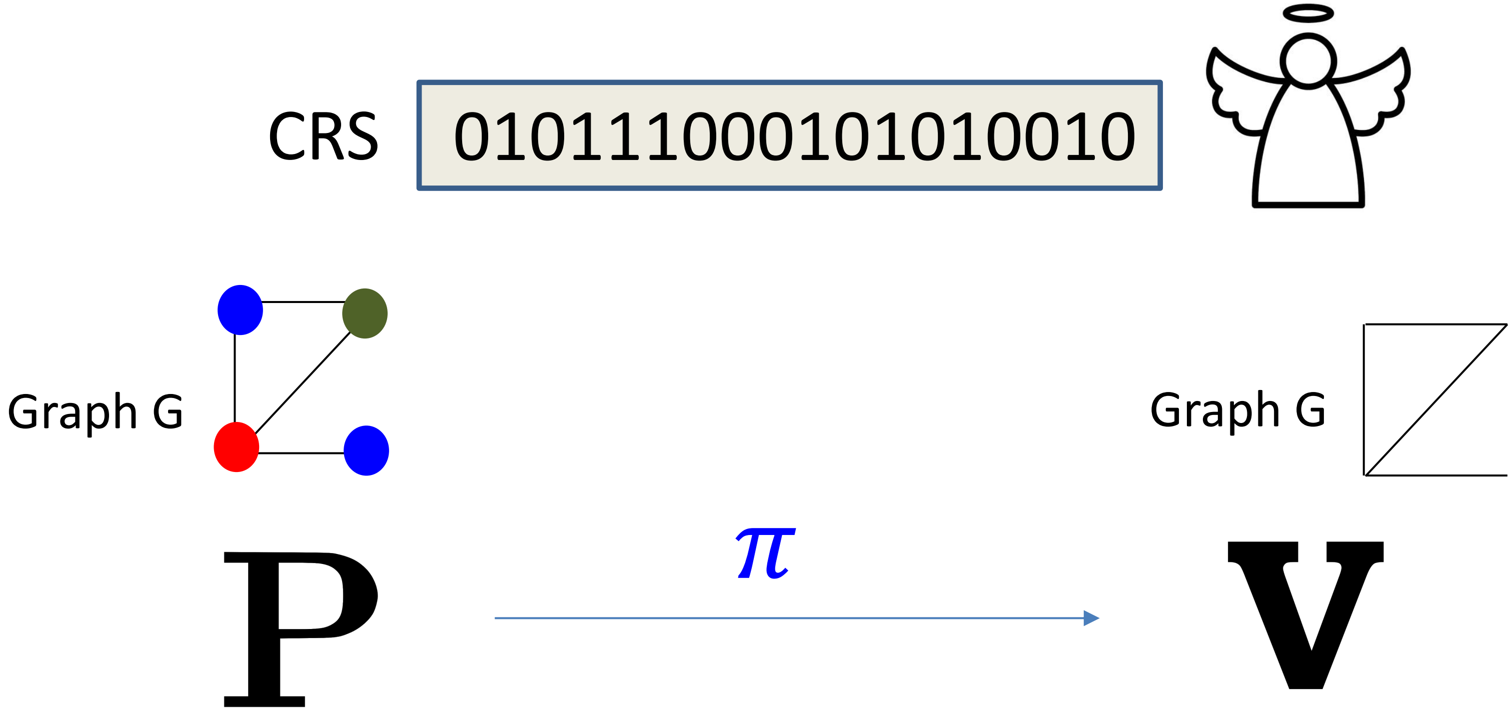
Graph G 

P $\xrightarrow{\pi}$ **V**

3. Zero Knowledge: There is a PPT simulator S such that for every G ∈ 3COL, S ***simulates the view*** of the verifier V.

$$S(G) \approx (CRS \leftarrow D, \pi \leftarrow P(G, colors))$$

NIZK in the CRS Model



3. Zero Knowledge: There is a PPT simulator S such that for every $x \in \mathrm{L}$ and witness w, S ***simulates the view*** of the verifier V.

$$S(x) \approx (CRS \leftarrow D, \pi \leftarrow P(x, w))$$

HOW TO CONSTRUCT NIZK IN THE CRS MODEL

1. **Blum-Feldman-Micali’88** ***(quadratic residuosity)***

2. Feige-Lapidot-Shamir’90 *(factoring)*

3. Groth-Ostrovsky-Sahai’06 *(bilinear maps)*

4. Canetti-Chen-Holmgren-Lombardi-Rothblum^2-Wichs’19 and Peikert-Shiehian’19 *(learning with errors)*

HOW TO CONSTRUCT NIZK IN THE CRS MODEL

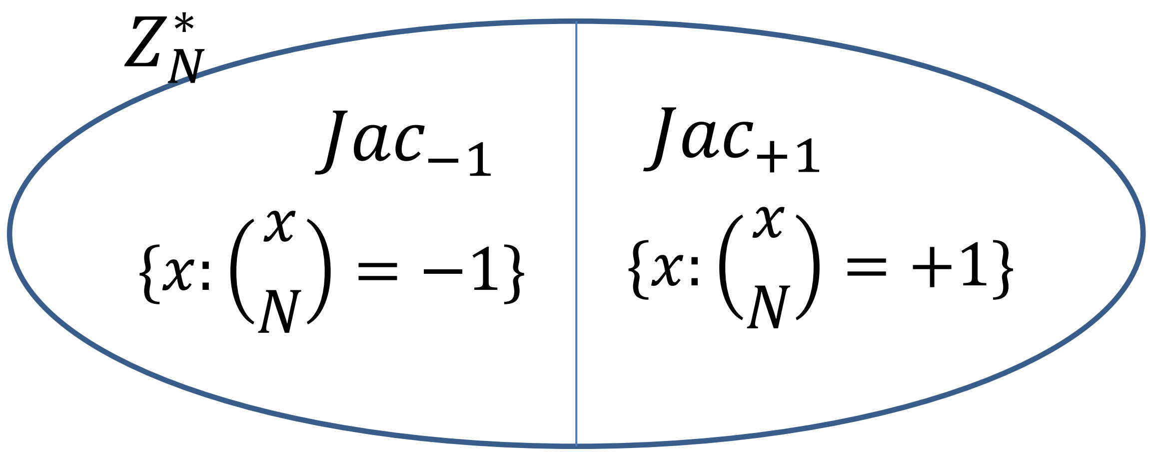
Step 1. **Review** our number theory hammers & polish them.

Step 2. **Construct** NIZK for a special NP language, namely quadratic *non*-residuosity.

Step 3. **Bootstrap** to NIZK for 3SAT, an NP-complete language.

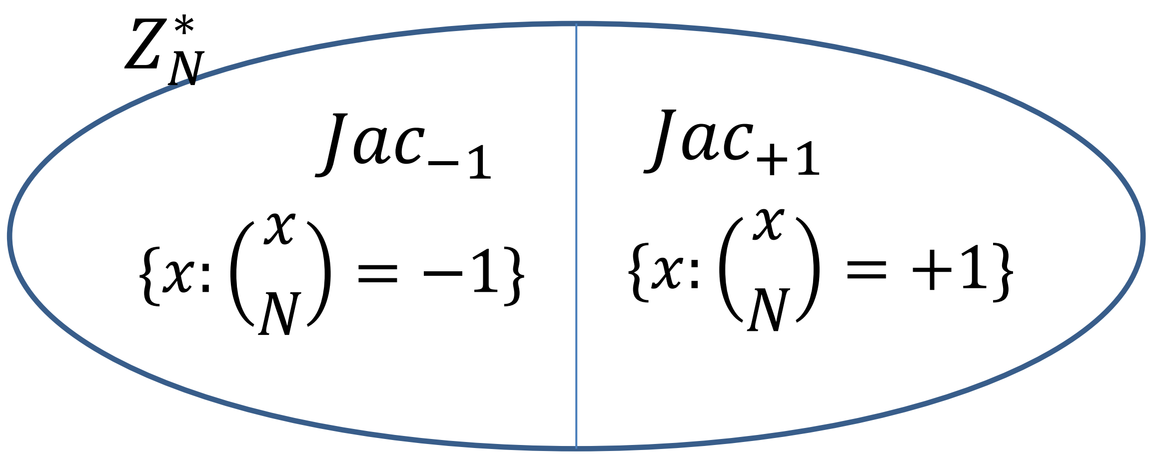
Quadratic Residuosity

Let $N = pq$ be a product of two large primes.



Quadratic Residuosity

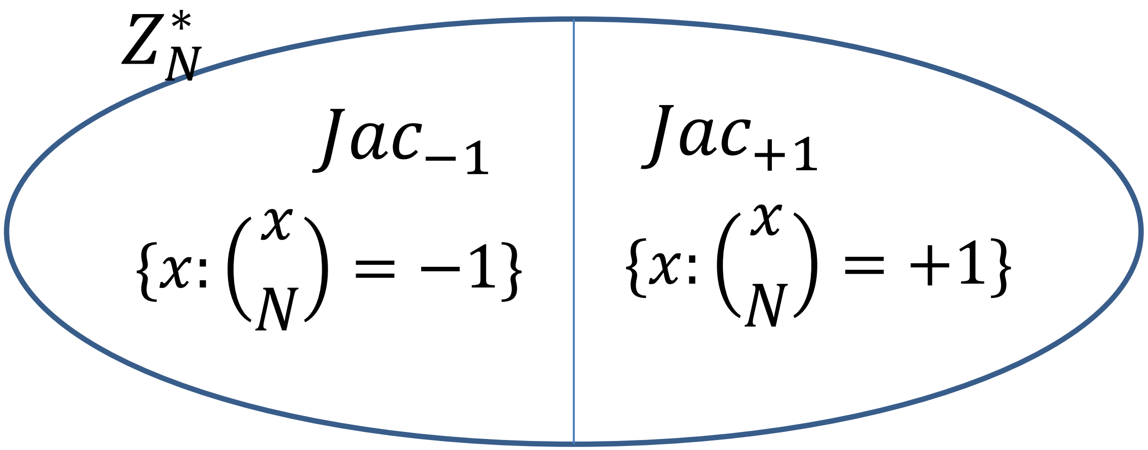
Let $N = pq$ be a product of two large primes.



Jac divides Z_N^* evenly unless N is a perfect square.

Quadratic Residuosity

Let $N = pq$ be a product of two large primes.



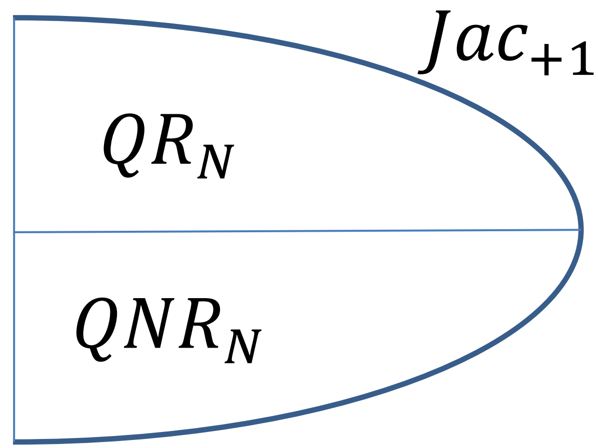
Surprising fact: Jacobi symbol $\left(\frac{x}{N}\right) = \left(\frac{x}{p}\right)\left(\frac{x}{q}\right)$ is computable in poly time **without knowing $\boldsymbol{p}$ and $\boldsymbol{q}$**.

Quadratic Residuosity

Let $N = pq$ be a product of two large primes.

So: $QR_N = \{x: \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = +1\}$

$QNR_N = \{x: \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1\}$

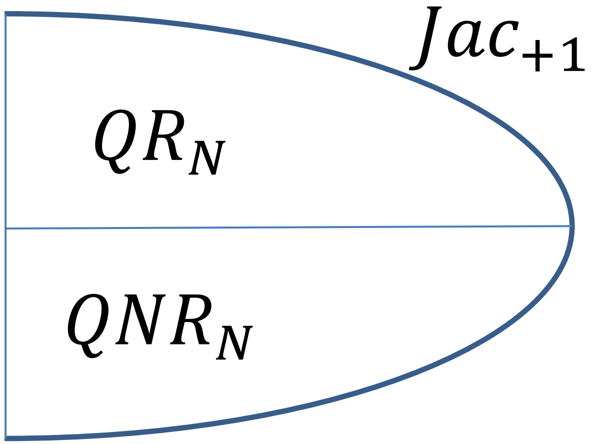


QR_N is the set of squares mod N and QNR_N is the set of non-squares mod N with Jacobi symbol +1.

Quadratic Residuosity

Exactly half residues even if

$$\boldsymbol{N = p^i q^j, i, j \geq 1,} \textbf{ not both even.}$$

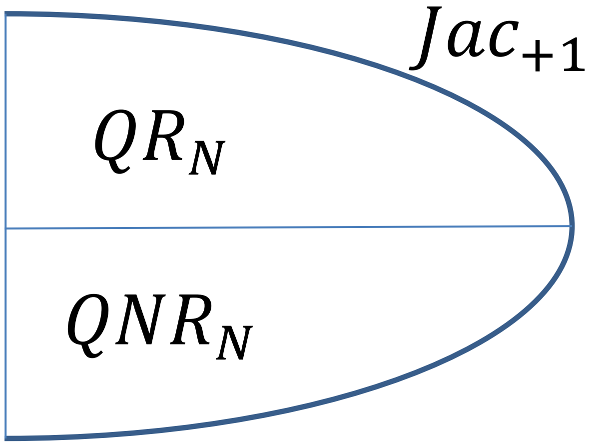


QR_N is the set of squares mod N and QNR_N is the set of non-squares mod N with Jacobi symbol +1.

Quadratic Residuosity

Exactly half residues even if

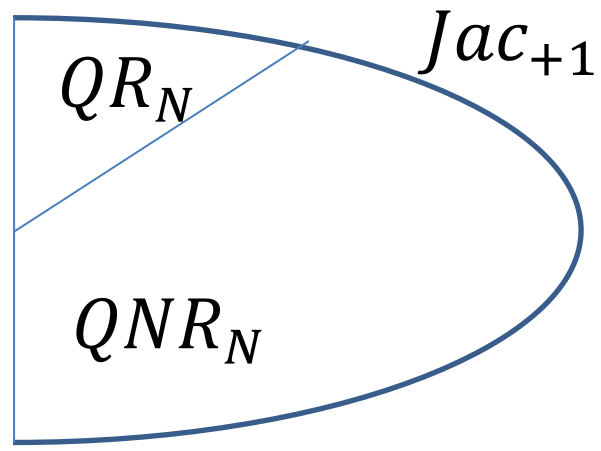
$$N = p^i q^j, i, j \geq 1, \text{not both even.}$$



<u>IMPORTANT PROPERTY</u>: If y_1 and y_2 are both in $Q\boldsymbol{N}R$, then their product $y_1 y_2$ is in QR.

Quadratic Residuosity

The fraction of residues smaller if N has three or more prime factors!



~~**<u>IMPORTANT PROPERTY</u>**: If y_1 and y_2 are both in $Q\mathbf{N}R$, then their product $y_1 y_2$ is in QR.~~

Quadratic Residuosity

Let $N = pq$ be a product of two large primes.

<u>Quadratic Residuosity Assumption (QRA)</u>

No PPT algorithm can distinguish between a random element of QR_N from a random element of QNR_N given only N.

HOW TO CONSTRUCT NIZK IN THE CRS MODEL

Step 1. **Review** our number theory hammers & polish them.

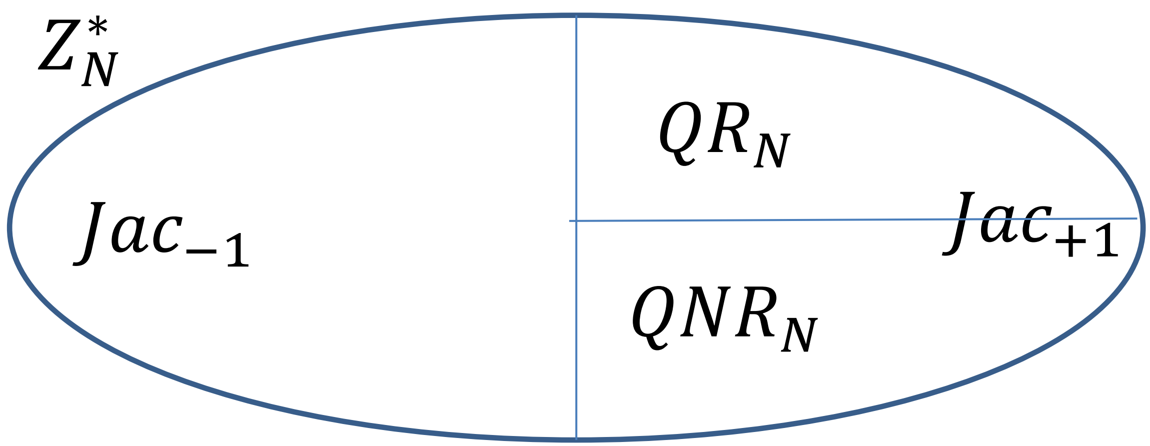
Step 2. **Construct** NIZK for a special NP language, namely quadratic *non*-residuosity.

Step 3. **Bootstrap** to NIZK for 3SAT, an NP-complete language.

NIZK for Quadratic Non-Residuosity

Define the NP language $GOOD$ with instances $(\boldsymbol{N}, \boldsymbol{y})$ where

- N is good: has exactly two prime factors and is not a perfect square; and
- $y \in QNR_N$ (that is, y has Jacobi symbol +1 but is not a square mod N)



NIZK for Quadratic Non-Residuosity

$$CRS = (r_1, r_2, \ldots, r_m) \leftarrow (Jac_N^{+1})^m$$

(N, y) (N, y)

P → **V**

If N is good and $y \in QNR_N$:

either r_i is in QR_N or yr_i is in QR_N

so I can compute $\sqrt{r_i}$ or $\sqrt{yr_i}$.

If not … I'll be stuck!

NIZK for Quadratic Non-Residuosity

$$CRS = (r_1, r_2, \dots, r_m) \leftarrow (Jac_N^{+1})^m$$

(N, y) **P** — $\forall i$: $\sqrt{r_i}$ OR $\sqrt{yr_i}$ → **V** (N, y)

Check:

- N is not a prime power,
- N is not a perfect square; and
- I received either a mod-N square root of r_i or yr_i

NIZK for Quadratic Non-Residuosity

$$CRS = (r_1, r_2, \ldots, r_m) \leftarrow (Jac_N^{+1})^m$$

(N, y) **P**

$\forall i$: $\sqrt{r_i}$ OR $\sqrt{yr_i}$ →

(N, y) **V**

Soundness (what if N has more than 2 prime factors)

No matter what y is, for half the r_i, both r_i and yr_i are ***not*** quadratic residues.

NIZK for Quadratic Non-Residuosity

$$CRS = (r_1, r_2, \ldots, r_m) \leftarrow (Jac_N^{+1})^m$$

(N, y) **P** $\xrightarrow{\forall i:\ \sqrt{r_i}\ \text{OR}\ \sqrt{yr_i}}$ **V** (N, y)

Soundness (what if N has more than 2 prime factors)

No matter what y is, ***for half the*** r_i, both r_i and yr_i are ***not*** quadratic residues.

NIZK for Quadratic Non-Residuosity

$$CRS = (r_1, r_2, \ldots, r_m) \leftarrow (Jac_N^{+1})^m$$

(N, y) **P** $\xrightarrow{\forall i:\ \sqrt{r_i} \text{ OR } \sqrt{yr_i}}$ **V** (N, y)

Soundness (what if y is a residue)

Then, if r_i happens to be a non-residue, both r_i and yr_i are ***not*** quadratic residues.

NIZK for Quadratic Non-Residuosity

$$CRS = (r_1, r_2, \ldots, r_m) \leftarrow (Jac_N^{+1})^m$$

(N, y) (N, y)

P $\forall i: \pi_i = \sqrt{r_i}$ OR $\sqrt{yr_i}$ → **V**

(Perfect) Zero Knowledge Simulator S:

First pick the proof π_i to be random in Z_N^*.

Then, *reverse-engineer* the CRS, letting $r_i = \pi_i^2$ or $r_i = \pi_i^2/y$ randomly.

$\blacksquare$

NIZK for Quadratic Non-Residuosity

$$CRS = (r_1, r_2, \dots, r_m) \leftarrow (Jac_N^{+1})^m$$

(N, y) (N, y)

P ⟶ **V**

CRS depends on the instance N. Not good.

Soln: Let CRS be random numbers. Interpret them as elements of Z_N^* and both the prover and verifier filter out Jac_N^{-1}.

NEXT LECTURE

Step 1. **Review** our number theory hammers
& polish them.

Step 2. **Construct** NIZK for a special NP language, namely
quadratic *non*-residuosity.

Step 3. **Bootstrap** to NIZK for 3SAT, an NP-complete
language.