Protocol 1

A sends $D((E(m, k_{E,B}), i_A), k_{D,A})_{i_A}$ to B

B sends $E(m, k_{E,A})$ to A

C can intercept and send B

$$D((E(m, k_{E,B}), i_C), k_{D,C})_{i_C}$$

Then B would send $E(m, k_{E,C})$ to C.

Protocol 2

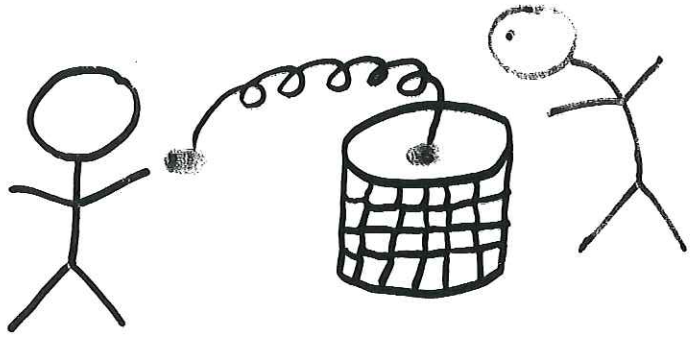
A sends $D(E((m, i_A), k_{E,B}), k_{D,A})_{i_A}$ to B

B sends $E(m, k_{E,A})$

Secure!

Coin flipping into a well

(Goldwasser - Micali)



Protocol

A
1. generate large random primes $p \equiv q \equiv 3 \pmod 4$
2. set $N = p \cdot q$ and publicize N
3. pick x with Jacobi symbol $+1 \pmod N$ at random and send x to B

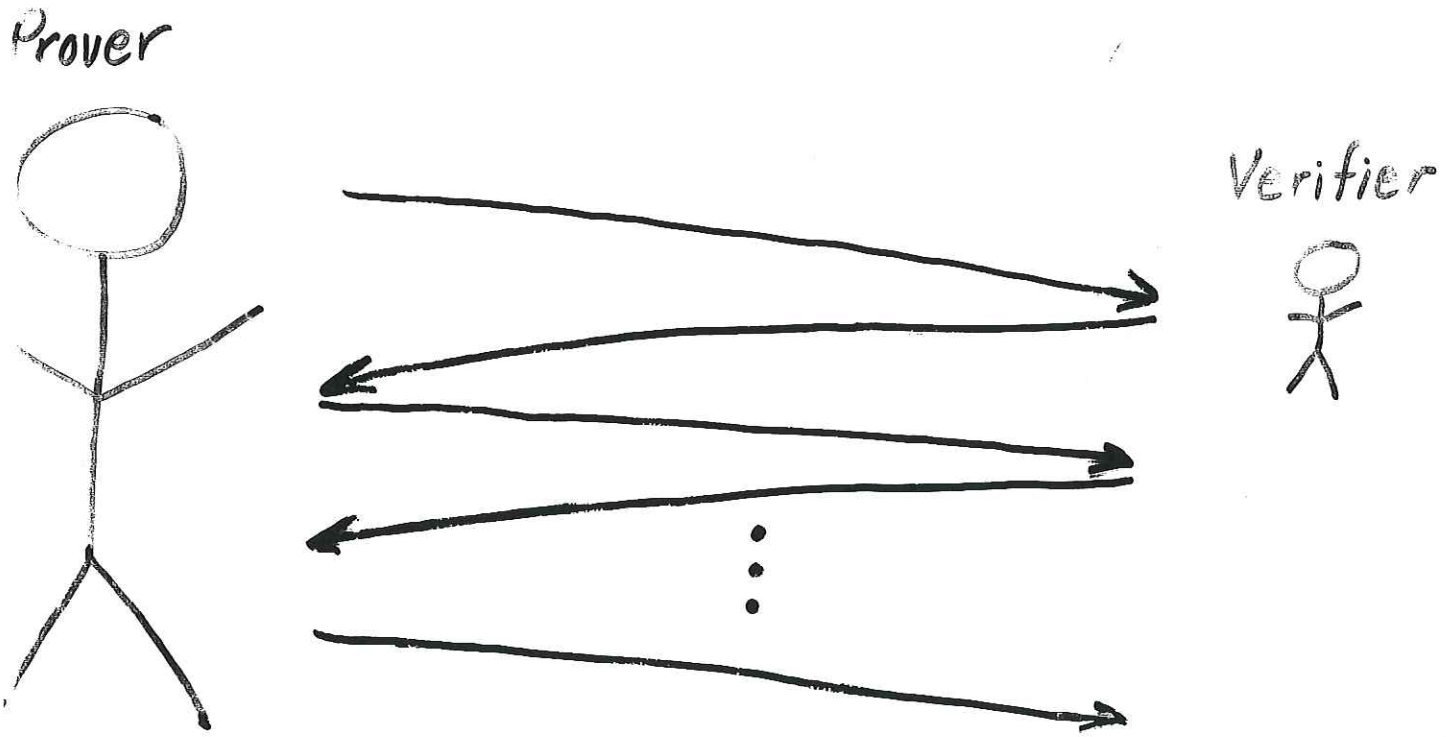
B
1. guess whether x is a residue or not → a square
2. send guess to A

A
- knows if B guessed correctly or not
- can later prove it by revealing p and q

Can A "prove" it without revealing anything about p or q?

Zero-Knowledge Proof Systems
(Goldwasser-Micali-Rackoff)

Claim: $x \in L$



Verifier is convinced.

Verifier learns nothing more.

Proof of Quadratic Residuosity
(Goldwasser - Micali - Rackoff)

y, N

Prover $(y = x^2)$ Verifier

$z = a^2 \pmod N$

flips a coin

$\begin{cases} \text{Form } z = a^2 \pmod N \\ \text{Form } z = b^2 y \pmod N \end{cases}$

a or b, whichever was requested

Repeat $|N| = \log_2 N$ times.

(With different random a's.)

Can the Prover Cheat?

Suppose y is not a square...

Case 1: z is a square.

If V asks for form $z = b^2 y \pmod N$, P can't produce b.

Case 2: z is not a square.

If V asks for form $z = a^2 \pmod N$, P can't produce a.

Each time P has only a 50% chance of cheating. So P can cheat $2^{-|N|}$ of the time.

<u>Def.</u> An interactive proof system is a protocol in which, if V follows its program

1. If $y \in L$ and if P follows its program, then V will accept with probability $\geq 1 - |y|^{-c}$ for every constant c.
2. If $y \notin L$, there is no program P could run which would cause the verifier to accept with probability $> |y|^{-c}$ for any constant c.

What does the Verifier learn?

Transcripts	Prover's bits z	Verifier's bits Form?	Verifier's private bits 0 or 1	Prover's bits a or b	...

Can a probabilistic polytime Simulator produce transcripts with same distribution as with true transcripts?

Simulator - flip coin to guess Form and choose a.
if 0, $z \leftarrow a^2 \pmod N$
if 1, $z \leftarrow a^2 y \pmod N$

send z to program for Verifier

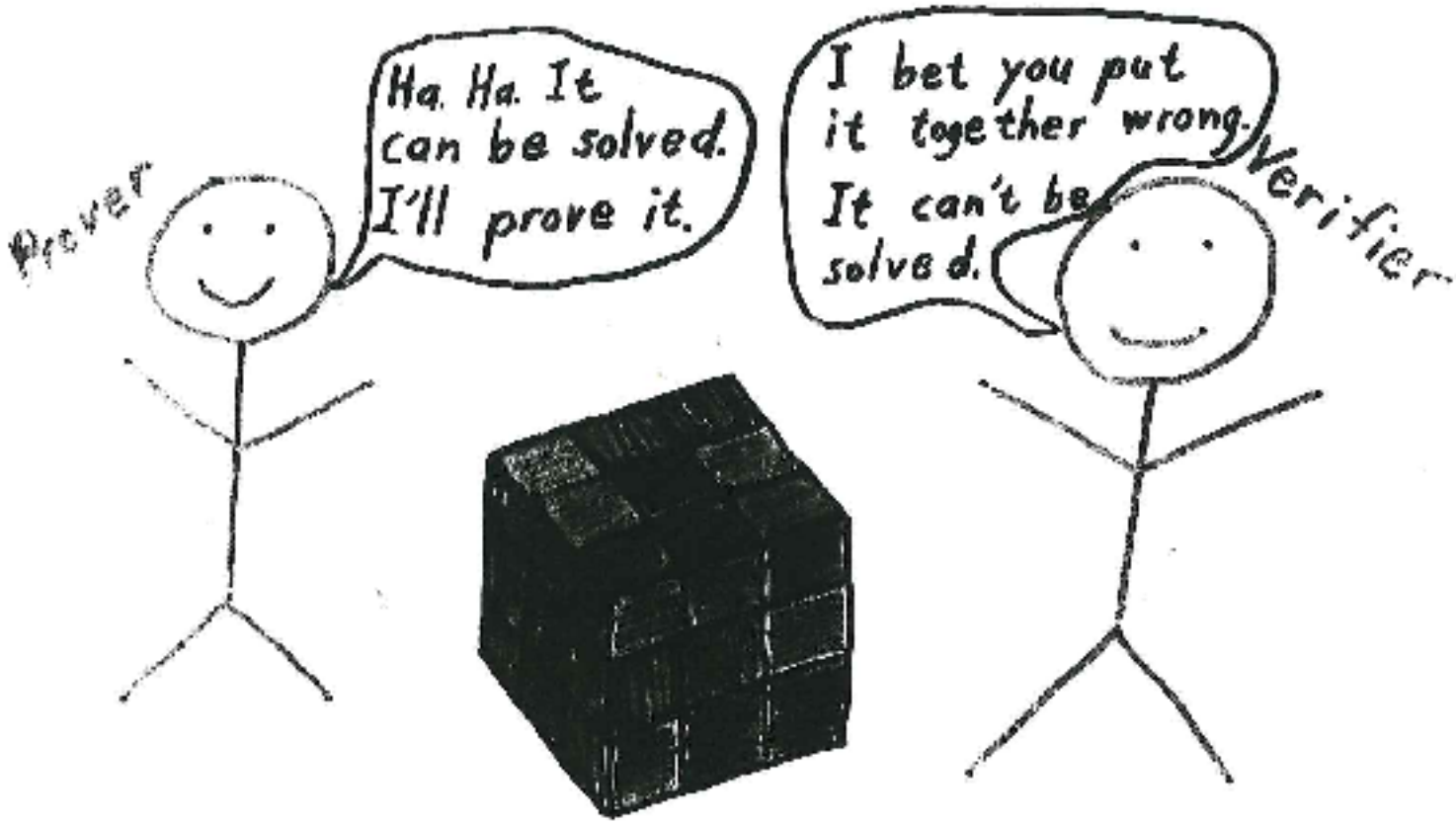
if V chooses guessed Form, OK
otherwise, back up tape and repeat

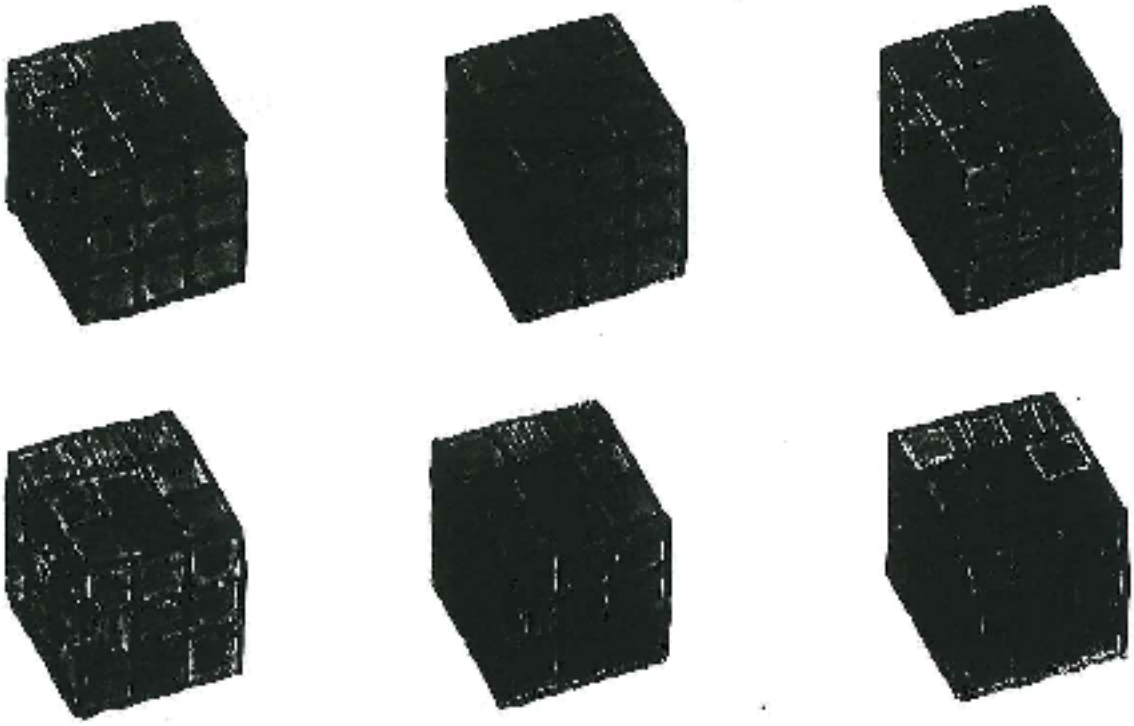
Expectation - 2 tries per round

$\therefore$ Simulation is polytime.

$\therefore$ This proof system is perfect zero-knowledge.

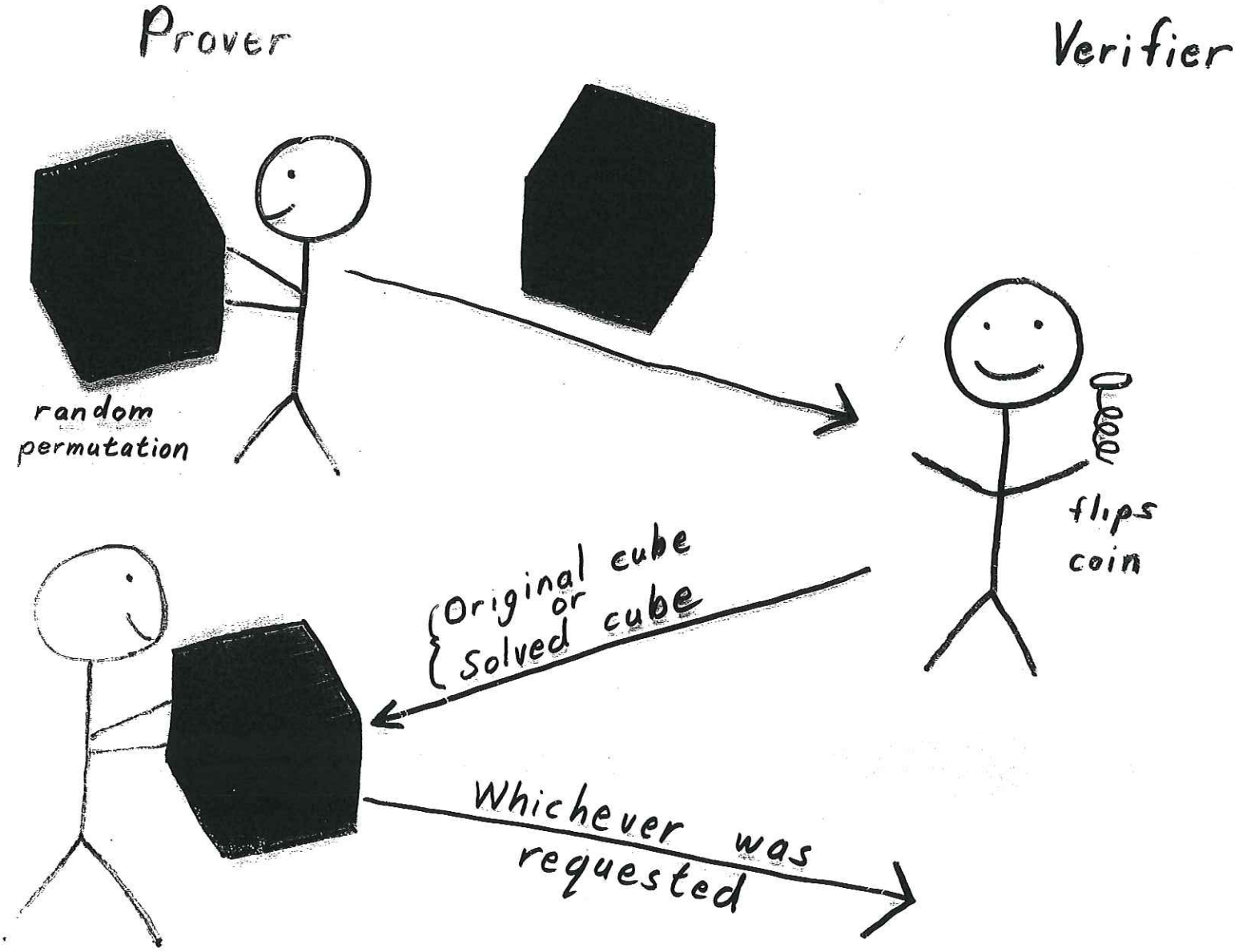
The Verifier could have done it on its own.


Prover
Ha. Ha. It can be solved. I'll prove it.
I bet you put it together wrong. It can't be solved.
Verifier



The Proof

Input: Cube, unsolved



Repeat n times.
Use different permutation each time.

Can the Prover Cheat?

Suppose it can't be solved.

From any position either

1.) It is impossible to get back to the original position with proper moves

or 2.) It is impossible to get to the solved position with proper moves.

Verifier has a 50-50 chance of catching the Prover each time.

$\therefore$ After n repetitions

≤ 1 chance in 2^n of cheating.

What does the Verifier learn?

Transcripts

Random cube position	V's random bits 	V's position request	Original or solved cube	...

Could the Verifier have produced a similar transcript on its own?

Can a probabilistic polytime Simulator produce transcripts with the same distribution as with "true" transcripts?

Simulator

- Flip coins to guess final position and to choose random moves.
- If 0, start with original cube
  If 1, start with solved cube
- Do moves. Send cube to Verifier.
- If V chooses guessed position, undo. Otherwise backtrack and repeat.

Expectation: 2 tries per round.

∴ Simulation is expected polytime.

∴ This protocol is perfect zero-knowledge.

$f(3, r_3)$

$f(2, r_2)$

$f(1, r_4)$

$f(1, r_1)$

$f(3, r_5)$

$f(2, r_6)$

Prover: 1. Choose random $\sigma \in S_3$. Recolor the graph.
2. Using a probabilistic encryption function f, encrypt the color for each vertex separately. Send the encryptions to V.

Verifier: 1. Randomly select an edge (u,v). Send (u,v) to P.

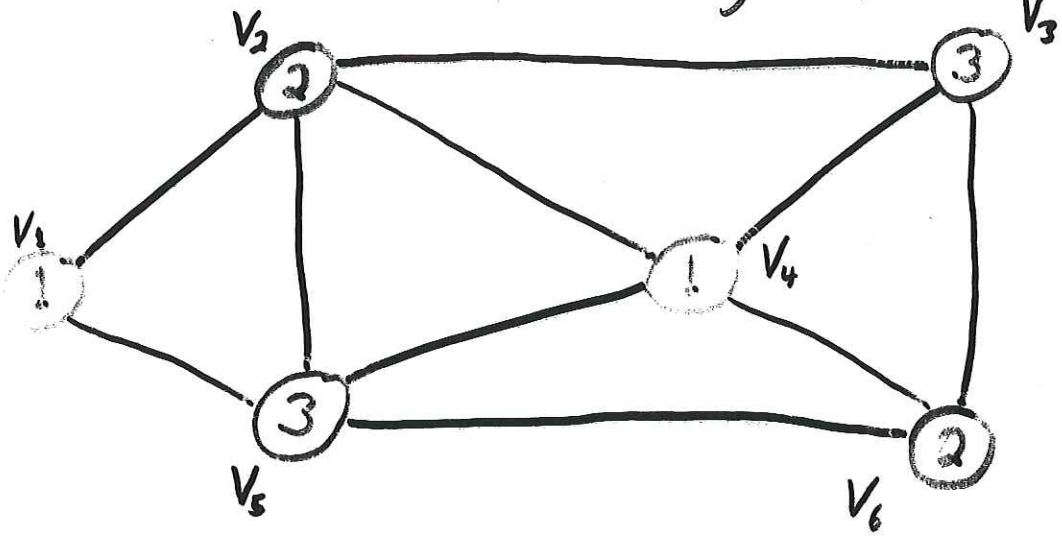
Prover: 1. Decode the colors on u and v.

Verifier: 1. Check the decoding.
2. Check that the colors are different.

Repeat $|E|^2$ times.

(With different σ's and encryptions.)

Graph 3-Colorability
(Goldreich-Micali-Wigderson)



Zero-Knowledge?

Simulator:

1. Flip coins to guess which edge V will choose.
2. Flip coins to give 2 different colors to those vertices.
3. Assign the color 1 to all other vertices.
4. Use f to encrypt. Send encryptions to V.
5. If V chooses guessed edge, OK. Otherwise back up tape and repeat.

Expectation - $|E|$ tries per round.

$\therefore$ Simulation is polytime.

The distribution is not identical.

It is polytime indistinguishable.

$\therefore$ This proof system is zero-knowledge, but not perfect zero-knowledge.

Theorems

1. (Goldreich, Micali, Wigderson) Every $L \in NP$ has a 0-knowledge proof system, under a cryptographic assumption. (Graph 3-colorability)

2. (Brassard, Chaum, Crépeau) Every $L \in NP$ has a 0-knowledge protocol proof, under a cryptographic assumption. (SAT)

   - Shows a circuit has a satisfying assignment.

Proofs of Knowledge

(Feige - Fiat - Shamir)

Prover and Verifier are BPP in power.

Prover has a knowledge tape.
It proves that, given input I, it "knows" a W, satisfying the predicate $P(I, W)$.

<u>Ex</u> "Knowing" a sq. rt. of y mod n.

Completeness - $\forall a \, \exists c \, \forall |I| > c$
if $\bar{A}$ has W on its knowledge tape such that $P(I, W)$
and $\bar{B}$ has the empty string on it knowledge tape
then $Pr((\bar{A}, \bar{B}) \text{ accepts } I) > 1 - 1/|I|^a$

Soundness - $\exists$ a prob. polytime TM M with control over A s.t. $\forall A$, $\forall$ initial contents KA of A's knowledge tape and RA of A's random tape, and sufficiently large $|I|$
if $(A, \bar{B})$ on input I accepts with nonnegligible prob, then $W = M(A, RA, KA, I)$ satisfies $P(I, W)$ with overwhelming probability

$\forall a \, \exists M \, \forall b \, \forall A \, \exists c \, \forall |I| > c \quad \forall RA \quad \forall KA$
$Pr((A, \bar{B}) \text{ accepts } I) > 1/|I|^a \implies$
$Pr(\text{output of } M(A, RA, KA) \text{ on } I \text{ satisfies } P)$
$> 1 - 1/|I|^b$

Nonresiduosity (GMR)

$y \in \mathbb{Z}_N^*$ a nonresidue

V: - chooses random $r_1 \in \mathbb{Z}_N^*$ and bit b
- sends $x_1 = r_1^2 y^b \pmod N$ to P
- proves that it knows r_1 and b

P: - sends 1 if x is a nonresidue
- " 0 " " " " residue

repeat $|N|$ times

How to show x has correct form:
Subprotocol: (Benaloh)

V: - chooses random $r_2, r_3 \in \mathbb{Z}_N^*$ and bit b'
- sends $(x_2, x_3) = (r_2^2 y^{b'} \pmod N, r_3^2 y^{1-b'} \pmod N)$ to P

P: - chooses random bit $c \rightarrow V$

V: - if $c = 0$, reveal r_2, r_3, b'
- if $c = 1$, reveal $\sqrt{x_1 x_2} \pmod N$ or $\sqrt{x_1 x_3} \pmod N$

repeat $|N|$ times

Why do we need the subprotocol?
- otherwise V could learn whether or not $x \in \mathbb{Z}_N^*$ was a residue.

Is the subprotocol a proof of knowledge?

Completeness - Suppose V knows r_1, b.
If $1=b=b'$, for $c=1$, $\sqrt{X_1 X_2} = r_1 r_2 y \pmod N$.
If $0=b=b'$, for $c=1$, $\sqrt{X_1 X_2} = r_1 r_2 \pmod N$.
If $1=b$, $0=b'$, $\sqrt{X_1 X_3} = r_1 r_3 y \pmod N$.
If $0=b$, $1=b'$, $\sqrt{X_1 X_3} = r_1 r_3 \pmod N$.

Soundness - Suppose (V,P) accepts x_1 with nonnegligible probability.
M runs V, trying to get an answer to both $c=0$ and $c=1$ for the same (X_2, X_3). It succeeds in expected polytime. From r_2, r_3, b' and the square root, it can find r_1 and b. ✓

Can the subprotocol help P? No, it's perfect O-K.

Simulator: guess whether $\bar{c}=0$ or $\bar{c}=1$
If guess $=0$, form (X_2, X_3) correctly.
If guess $=1$, flip coin to get b'.
If $b'=0$, form $(r_2^2 X_1 \pmod N, r_3^2 X_1 y \pmod N)$.
If $b'=1$, form $(r_2^2 X_1 y \pmod N, r_3^2 X_1 \pmod N)$.

Distributions?
Expected time?

Is the entire protocol perfect zero-knowledge?

Yes. The Simulator uses the Verifier's proof of knowledge to learn the bit b.

When Prover is asked for a bit,
- produce one randomly
- get result
- write it on transcript (View)
- If result is incorrect, quit
  - Otherwise, back up V's program and give the other bit
  - If V ever answers both questions, S can find b.
  - If V never answers both, there was only one set of questions V could answer out of $2^{|N|}$ sets. In this case, run a factoring algorithm to determine if x is a residue.

Expected time: $\frac{1}{2^{|N|}} \cdot 2^{|N|} + \left(1 - \frac{1}{2^{|N|}}\right)(\text{poly})$ which is polynomial.

Distribution?