Lecture 12: NIZKs from the BGN cryptosystem

Zero-knowledge proofs

263-4665-00L

Lecturer: Jonathan Bootle

Announcements

- Exercise sheet 13 posted on Moodle
- Graded, 10% of final grade
- Submit through Moodle on or before 23:59, 15/12/2023
- Please email if you think you've found a typo or mistake

- 15/12/2023 exercise session to be used for exploring libraries for implementing zero-knowledge
- Still working on extra (optional, non-examinable) video about reducing verification costs for polynomial commitments.

Last time

- Compiling IP/IOP protocols into zero-knowledge argument.

- NIZKs without setup only cover languages in **BPP**.

Agenda

- **Non-interactive zero-knowledge (NIZK) definitions**

Pairing-based constructions of NIZK

- From reasonable cryptographic assumptions

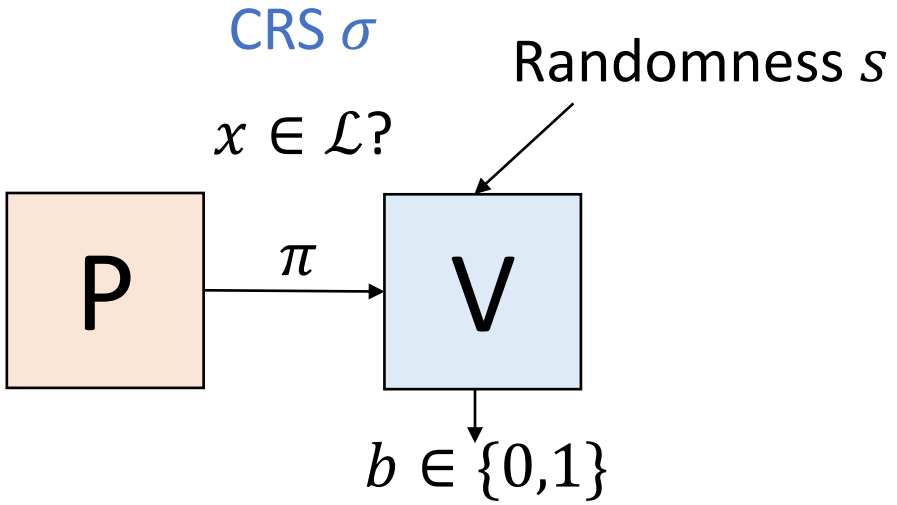
$O(N)$ proof size for Boolean circuits

- From strong cryptographic assumptions

$O(1)$ proof size for Arithmetic circuits

Syntax for non-interactive zero-knowledge

- Include a common reference string (CRS) containing ingredients used in the proof.



Definition:

A *non-interactive proof system* for an NP relation $\mathcal{R}$ consists of three efficient algorithms (K, P, V) which are

- the CRS generator $K(1^\lambda) \to \sigma$,
- the prover $P(\sigma, x, w) \to \pi$,
- the verifier $V(\sigma, x, \pi) \to b$.

Suppressing random inputs

K may take $|x|$ or even x as input

Ideally σ is uniformly random but may be structured

Security of non-interactive proofs

Easy to modify to get computational and statistical security notions

- **Completeness:** $\forall (x, w) \in \mathcal{R}$,

$$\Pr[b = 1 | \, \sigma \leftarrow K(1^\lambda), \pi \leftarrow P(\sigma, x, w), b \leftarrow V(\sigma, x, \pi)] = 1$$

Adaptive

- **Soundness:** $\forall P^*$,

$$\Pr[x \notin \mathcal{L}_\mathcal{R}, b = 1 | \, \sigma \leftarrow K(1^\lambda), (x, \pi) \leftarrow P^*(\sigma), b \leftarrow V(\sigma, x, \pi)] \approx 0$$

Adaptive

- **Zero-knowledge:** $\exists$ efficient simulators (S_1, S_2) such that $\forall A$ producing $(x, w) \in \mathcal{R}$,

Simulated σ indistinguishable from normal σ

$$\{(\sigma, \pi) : \sigma \leftarrow K(1^\lambda), (x, w) \leftarrow A(\sigma), \pi \leftarrow P(\sigma, x, w)\}$$
$$\approx \{(\sigma, \pi) : (\sigma, \tau) \leftarrow S_1(1^\lambda), (x, w) \leftarrow A(\sigma), \pi \leftarrow S_2(\sigma, x, \tau)\}$$

Simulation trapdoor τ (replaces oracle access to V^*)

In non-adaptive definitions, x is not chosen based on σ

These are *single-theorem* definitions. No security guarantees reusing σ for many x.

Knowledge soundness

Definition:

(K, P, V) is a *proof of knowledge* for a relation $\mathcal{R}$ if ∃ efficient extractors E_1, E_2 such that for all P^*,

Extractor's σ indistinguishable from normal σ

Extraction trapdoor ξ
(replaces oracle access to P^*)

- $\{\sigma : (\sigma, \xi) \leftarrow E_1(1^\lambda)\} \approx \{\sigma : \sigma \leftarrow K(1^\lambda)\}$, and

- $$\Pr\left[\begin{array}{c} V(\sigma, x, \pi) = 0 \\ \vee\ (x, w) \in \mathcal{R} \end{array} : \begin{array}{c} (\sigma, \xi) \leftarrow E_1(1^\lambda), (x, \pi) \leftarrow P^*(\sigma) \\ w \leftarrow E_2(\sigma, \xi, x, pi) \end{array}\right] \approx 1.$$

$V(\sigma, x, \pi) \Rightarrow (x, w) \in \mathcal{R}$

How can we trust the CRS?

- Simulation trapdoors let us produce proofs without knowing witnesses (breaking soundness)
- Extraction trapdoors let us extract witnesses from proofs (breaking ZK)
- Trapdoor σ are indistinguishable from normal σ.

Mitigate risks using

- “Subversion resistant” NIZK constructions
- “Updatable CRS” NIZK constructions
- “Verifiable CRS” NIZK constructions
- MPC protocols to generate σ

How can we trust the CRS?

[2] This curious property makes our result potentially applicable. For instance, all libraries in the country possess identical copies of the random tables prepared by the Rand Corporation. Thus, we may think of ourselves as being already in the scenario needed for noninteractive zero-knowledge

Agenda

- Non-interactive zero-knowledge (NIZK) definitions ✓

Pairing-based constructions of NIZK

- **From reasonable cryptographic assumptions**
  - The BGN cryptosystem
  - BGN bit proofs
  - BGN proofs for CSAT

$O(N)$ proof size for Boolean circuits

- From strong cryptographic assumptions

$O(1)$ proof size for Arithmetic circuits

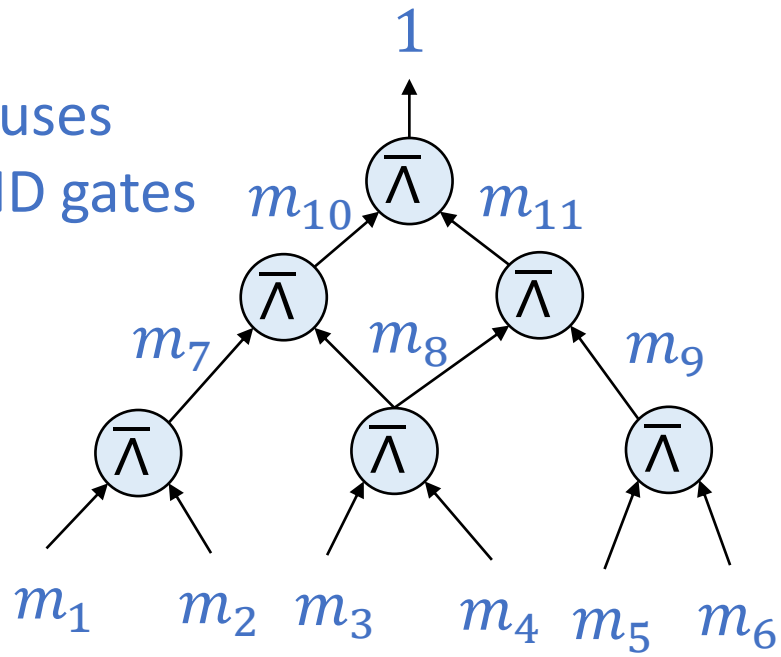
Boolean circuit NIZK idea

$\mathbb{x}$ = circuit description
$\mathbb{w}$ = satisfying wire values

Instance: circuit over $\mathbb{Z}_2$ with output.

Witness: input wire values giving correct output.

W.L.O.G. uses only NAND gates



a	b	c	$\overline{a \wedge b} == c$	$a + b + 2c - 2$
0	0	0	0	-2
0	0	1	1	0
0	1	0	0	-1
0	1	1	1	1
1	0	0	0	-1
1	0	1	1	1
1	1	0	1	0
1	1	1	0	2

$$\overline{a \wedge b} = c \Leftrightarrow a + b + 2c - 2 \in \{0,1\}$$

Proof idea:

- Commit to each wire value.
- Prove each wire value $\in \{0,1\}$.
- Prove $a + b + 2c - 2 \in \{0,1\}$ for wires around each gate.

Need a commitment scheme with NI bit proofs

Composite-order symmetric pairings

Definition:

A *symmetric bilinear group* is a triple of two groups of order $n = pq$ (where p, q are distinct primes) and a *bilinear map* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfying

$$\forall a, b \in \mathbb{Z}_p, \forall G_1, G_2 \in \mathbb{G},$$
$$e(a \cdot G_1, b \cdot G_2) = ab \cdot e(G_1, G_2)$$

Pairing maps
'multiply DLOGs'

which is non-degenerate i.e.

$$\text{If } \mathbb{G} = \langle G_1 \rangle = \langle G_2 \rangle, \text{ then } \mathbb{G}_T = \langle e(G_1, G_2) \rangle$$

Changes from last time:

- Order n instead of p
- "Type 1" symmetric setting with $\mathbb{G}_1 \cong \mathbb{G}_2 \cong \mathbb{G}$

Facts

Claim: (symmetry)

$\forall G_1, G_2 \in \mathbb{G}, e(G_1, G_2) = e(G_2, G_1)$

Proof:

Let $\mathbb{G} = \langle G \rangle$. Write $G_1 = a \cdot G$ and $G_2 = b \cdot G$. Then

$e(G_1, G_2) = ab \cdot e(G, G) = ba \cdot e(G, G) = e(G_2, G_1)$ by bilinearity.

Both claims together imply homomorphism in the right input

Claim:

$\forall G_1, G_2, H \in \mathbb{G}, e(G_1 + G_2, H) = e(G_1, H) + e(G_2, H).$

Proof: exercise

The Boneh-Goh-Nissim Cryptosystem

Large enough that n is difficult to factor e.g. $O(\lambda^3)$ bits

Setup: on input $\lambda \in \mathbb{N}$, sample distinct primes p, q and composite order symmetric bilinear group $e, \mathbb{G}, \mathbb{G}_T$ of order $n = pq$, and $G, H \in \mathbb{G}$.

Sample $B \in \mathbb{N}$. Output $pp := (e, \mathbb{G}, \mathbb{G}_T, G, H, n, B)$.

$B \leq \text{poly}(\lambda)$

Exact sampling method to be discussed

Commit: given $m \in \{0, \dots, B-1\}, pp$, sample $r \leftarrow \mathbb{Z}_n$.

Compute $C = m \cdot G + r \cdot H$. Output (C, r).

Verify: check $m \in \{0, \dots, B-1\}$ and $C == m \cdot G + r \cdot H$.

Homomorphic (looks like Pedersen)

For us, a commitment scheme (but can be used as an encryption scheme)

Dual-mode parameter generation

- $\mathbb{G}$ has order $n = pq$.
- $\text{Setup}_{binding}: G \leftarrow \mathbb{G},\ s \leftarrow \mathbb{Z}_n^*,\ H = ps \cdot G.$

a generator (w.h.p.)

random generator of order q subgroup

- $\text{Setup}_{hiding}: G \leftarrow \mathbb{G},\ s \leftarrow \mathbb{Z}_n^*,\ H = s \cdot G.$

random generator of whole group $\mathbb{G}$

Definition:

The subgroup hiding assumption holds if

$$\{\text{Setup}_{binding}(1^\lambda)\} \approx_c \{\text{Setup}_{hiding}(1^\lambda)\}$$
$$\{G, ps \cdot G\} \approx_c \{G, s \cdot G\}$$

The BGN cryptosystem is hiding

Proof:

- Using Setup_{hiding}: $G \leftarrow_{\$} \mathbb{G},\ s \leftarrow_{\$} \mathbb{Z}_n^*,\ H = s \cdot G$.

random generator of whole group $\mathbb{G}$

- For $r \leftarrow_{\$} \mathbb{Z}_n$, $r \cdot H$ is uniformly random in $\mathbb{G}$.
- Hence $C = m \cdot G + r \cdot H$ is uniformly random in $\mathbb{G}$.
- Therefore Setup_{hiding} gives *perfect* hiding.

With Setup_{hiding}, BGN is *equivocable* with equivocation key s.

$C = m \cdot G + r \cdot H = m' \cdot G + r' \cdot H$ where $r' := \frac{m-m'}{s} mod\ n$.

- The output of $\text{Setup}_{binding}$ is computationally indistinguishable from Setup_{hiding} under the subgroup hiding assumption.
- Therefore $\text{Setup}_{binding}$ still gives *computational* hiding.

The BGN cryptosystem is perfectly binding

Proof:

- Using $\text{Setup}_{binding}$: $G \leftarrow_{\$} \mathbb{G},\ s \leftarrow_{\$} \mathbb{Z}_n^*,\ H = ps \cdot G$.

random generator of order q subgroup

- Suppose $C = m \cdot G + r \cdot H = m' \cdot G + r' \cdot H$ for distinct $m, m' \in \{0, \dots, B-1\}$.
- Then $e(C, q \cdot G) = e(m \cdot G + r \cdot H, q \cdot G) = e(m \cdot G + rps \cdot G, q \cdot G)$
$= qm \cdot e(G,G) + rspq \cdot e(G,G) = qm \cdot e(G,G)$
- Similarly, $e(C, q \cdot G) = qm' \cdot e(G,G)$. Hence $q(m - m') \cdot e(G,G) = 0$.
- By non-degeneracy, $e(G,G)$ has order n so $n \mid q(m - m')$.
- Hence $q(m - m') = kn$, so $(m - m') = kp$.
- $m \equiv m' mod\ p$ but $B \ll p$ so $m = m'$.

With $\text{Setup}_{binding}$, BGN is *extractable* with extraction key s.

Compute $e(C, q \cdot G)$, check whether it is equal to $qm \cdot e(G,G)$ for each $m \in \{0, \dots, B-1\}$.

- The output of Setup_{hiding} is computationally indistinguishable from $\text{Setup}_{binding}$ under subgroup hiding, so Setup_{hiding} gives *computational* binding.

Agenda

- Non-interactive zero-knowledge (NIZK) definitions ✓

Pairing-based constructions of NIZK

- From reasonable cryptographic assumptions
  - The BGN cryptosystem ✓
  - **BGN bit proofs**
  - BGN proofs for CSAT

$O(N)$ proof size for Boolean circuits

- From strong cryptographic assumptions

$O(1)$ proof size for Arithmetic circuits

Proof for committed bits

Setup choice didn't matter

- $\mathcal{R} \coloneqq \{((\sigma, C), (m, r)) : C \in \mathbb{G},\ m \in \{0,1\}, r \in \mathbb{Z}_n, C = m \cdot G + r \cdot H\}.$

$K(1^\lambda)$ Hiding or binding
Output $\sigma = (e, \mathbb{G}, \mathbb{G}_T, G, H, n)$ from BGN setup.

$P(\sigma, C, m, r)$
Compute $\pi \coloneqq r(2m-1) \cdot G + r^2 \cdot H \in \mathbb{G}$.

$V(\sigma, C, \pi)$
Output $(e(C, C-G) == e(\pi, H))$.

Proof idea: C commits to m
$C - G$ commits to $m - 1$

$m \in \{0,1\}$
$\Updownarrow$
$m(m-1) = 0$

Pairing 'multiplies' committed values.
π is the leftover terms.

Completeness:

Suppose $m \in \{0,1\}, r \in \mathbb{Z}_n$,
$C = m \cdot G + r \cdot H$, and
$\pi = r(2m-1) \cdot G + r^2 \cdot H$.

We have $e(C, C-G)$
$= e(m \cdot G + r \cdot H, (m-1) \cdot G + r \cdot H)$
$= m(m-1) \cdot e(G,G)$ Bilinearity
$+mr \cdot e(G,H) + (m-1)r \cdot e(H,G)$
$+r^2 \cdot e(H,H)$ Symmetry and bilinearity
$m(m-1) = 0$
$= r(2m-1) \cdot e(G,H) + r^2 \cdot e(H,H)$
$= e\left(r(2m-1) \cdot G + r^2 H\right) = e(\pi, H)$

So V always accepts.

m is uniquely determined

Soundness analysis

$C = m \cdot G + r \cdot H$
$= m \cdot G + \frac{m_* - m}{ps} ps \cdot G$
$= m_* \cdot G$

- Using $\text{Setup}_{binding}$: $G \leftarrow_\$ \mathbb{G},\ s \leftarrow_\$ \mathbb{Z}_n^*,\ H = ps \cdot G.$
- $\mathbb{G} = \langle G \rangle$ so $\exists m_* \in \mathbb{Z}_n$ with $C = m_* \cdot G.$
- Let $m := m_* mod\ p$ and $r := \frac{m_* - m}{ps} mod\ n.$ Then $C = m \cdot G + r \cdot H.$
- $\forall G' \in \mathbb{G}, q \cdot e(G', H) = spq \cdot e(G', G) = 0.$ Group has order $n = pq.$

- If $e(C, C - G) = e(\pi, H)$ then expanding $e(C, C - G)$ gives

$m(m-1) \cdot e(G,G) + e\big(r(2m-1) \cdot G + r^2 \cdot H, H\big) = e(\pi, H).$

- $qm(m-1) \cdot e(G,G) + q \cdot e\big(r(2m-1) \cdot G + r^2 \cdot H, H\big) = q \cdot e(\pi, H).$
- $\Rightarrow qm(m-1) \cdot e(G,G) = 0.$ $e(G,G)$ has order n by non-degeneracy
- $\Rightarrow qm(m-1) = kn$ for some $k \in \mathbb{Z}$. Hence $m(m-1) = kp.$
- $m(m-1) = 0\ mod\ p.$ $m < p$ so $m \in \{0,1\}$

‘Almost’ knowledge soundness analysis

$E_1(1^\lambda) \to (\sigma, \xi := q \cdot G)$
Using BGN binding setup.

$E_2(\sigma, C, \pi) \to m \in \{0,1\}$
Using commitment extraction.

‘almost’ knowledge soundness because we only get m and not r satisfying $C = m \cdot G + r \cdot H$.

We want $\{\sigma : (\sigma, \xi) \leftarrow E_1(1^\lambda)\} \approx \{\sigma : \sigma \leftarrow K(1^\lambda)\}$, and $\Pr\left[\begin{matrix} V(\sigma, x, \pi) = 0 \\ \vee\ (x, w) \in \mathcal{R} \end{matrix} : \begin{matrix} (\sigma, \xi) \leftarrow E_1(1^\lambda), (x, \pi) \leftarrow P^*(\sigma) \\ w \leftarrow E_2(\sigma, \xi, x, pi) \end{matrix}\right] \approx 1.$

Why are σ from K, E_1 indistinguishable?

- Trivially if K uses binding setup. Computationally if K uses hiding setup.

Why is the output of E_2 a witness?

- E_2 extracts the unique m from the previous slide.
- The soundness analysis shows that $m \in \{0,1\}$.

Proof uniqueness (witness indistinguishabilility)

- Using Setup_{hiding}: $G \leftarrow_{\$} \mathbb{G},\ s \leftarrow_{\$} \mathbb{Z}_n^*,\ H = s \cdot G$.

- Suppose $\pi_1, \pi_2 \in \mathbb{G}$ satisfy $e(\pi_1, H) = e(\pi_2, H) = e(C, C - G)$.
- Writing $\pi_i = a_i \cdot G$, we have $a_1 \cdot e(G, H) = a_2 \cdot e(G, H)$.
- Hence $(a_1 - a_2) \cdot e(G, H) = 0$.
- Hence $a_1 - a_2 \equiv 0\ mod\ n$, so $a_1 = a_2$ and $\pi_1 = \pi_2$.

Non-degeneracy ⇒ $e(G, H)$ has order n since both G, H are generators.

- Therefore, $\forall C \in \mathbb{G}$, there is at most one accepting proof $\pi \in \mathbb{G}$.

With Setup_{hiding}, BGN is *equivocable* with equivocation key s.
$\mathbb{G} = \langle G \rangle$ so $\exists m \in \mathbb{Z}_n$ with $C = m \cdot G$.
$C = m \cdot G = m' \cdot G + r' \cdot H$ where $r' = \frac{m - m'}{s} mod\ n$.

⇒ a valid proof exists based on $C = 0 \cdot G + r' \cdot H$

Agenda

- Non-interactive zero-knowledge (NIZK) definitions ✓

Pairing-based constructions of NIZK

- From reasonable cryptographic assumptions
  - The BGN cryptosystem ✓
  - BGN bit proofs ✓
  - **BGN proofs for CSAT**

$O(N)$ proof size for Boolean circuits

- From strong cryptographic assumptions

$O(1)$ proof size for Arithmetic circuits

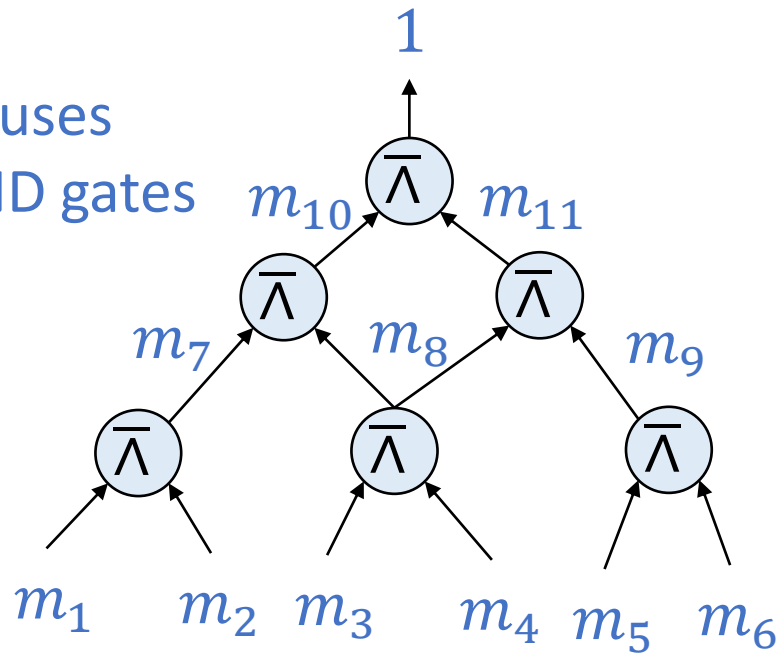
Boolean circuit NIZK idea

$\mathbb{x}$ = circuit description
$\mathbb{w}$ = satisfying wire values

Instance: circuit over $\mathbb{Z}_2$ with output.

Witness: input wire values giving correct output.

W.L.O.G. uses only NAND gates



a	b	c	$\overline{a \wedge b} == c$	$a + b + 2c - 2$
0	0	0	0	-2
0	0	1	1	0
0	1	0	0	-1
0	1	1	1	1
1	0	0	0	-1
1	0	1	1	1
1	1	0	1	0
1	1	1	0	2

$$\overline{a \wedge b} = c \Leftrightarrow a + b + 2c - 2 \in \{0,1\}$$

Proof idea:

- Commit to each wire value.
- Prove each wire value $\in \{0,1\}$.
- Prove $a + b + 2c - 2 \in \{0,1\}$ for wires around each gate.

Need a commitment scheme with NI bit proofs

$O(N)$ proof size and verification time

NIZK for Boolean satisfiability

$K(1^\lambda)$ Hiding or binding

Output $\sigma = (e, \mathbb{G}, \mathbb{G}_T, G, H, n)$ from BGN setup.

$P(\sigma, \mathbb{x}, \mathbb{w})$: For each m_i

- Sample $r_i \leftarrow_\$ \mathbb{Z}_n$. Compute $C_i = m_i \cdot G + r_i \cdot H \in \mathbb{G}$.
- Compute $\pi_i := r_i(2m_i - 1) \cdot G + r_i^2 \cdot H \in \mathbb{G}$.
- Compute $C_{out} = 1 \cdot G + 0 \cdot H \in \mathbb{G}$.
- For each gate $\overline{m_i \wedge m_j} = m_k$, compute

$C_{ijk} = m_{ijk} \cdot G + r_{ijk} \cdot H := C_i + C_j + 2 \cdot C_k - 2 \cdot G$.

- Compute $\pi_{ijk} := r_{ijk}(2m_{ijk} - 1) \cdot G + r_{ijk}^2 \cdot H \in \mathbb{G}$.
- Output $((C_i)_i, (\pi_i)_i, (\pi_{ijk})_{ijk})$.

$V(\sigma, C, \pi)$: Output 1 if and only if $e(C_i, C_i - G) = e(\pi_i, H)$ for each i and similarly for each gate.

Completeness, soundness:
Immediate from the properties of the bit proofs.

Knowledge soundness:

- E_1 is the same as in the bit proofs (binding setup).
- For E_2, we use the bit proof extractor to extract each wire value.
- Bit proof soundness implies that each gate is satisfied.

No need to extract commitment randomness. 'Almost' knowledge soundness of bit proofs suffices.

Use equivocation key s to cheat openings and satisfy NANDs

Zero-knowledge analysis

$S_1(1^\lambda) \to (\sigma, \tau := s)$
Using BGN hiding setup.

Using Setup_{hiding}: $G \leftarrow_\$ \mathbb{G}$, $s \leftarrow_\$ \mathbb{Z}_n^*$, $H = s \cdot G$.

What is the verifier's view?

$((C_i)_i, (\pi_i)_i, (\pi_{ijk})_{ijk})$

- Each C_i is uniformly random in $\mathbb{G}$.
- Each π_i and π_{ijk} is uniquely determined by the C_i.

Why is the simulator valid?

- The distributions of σ and C_i are identical and the bit proofs uniquely determined from these.

$S_2(\sigma, \mathbb{x}, \tau)$: For each wire, set $m_i := 0$.

- Sample $r_i \leftarrow_\$ \mathbb{Z}_n$. Compute $C_i = m_i \cdot G + r_i \cdot H \in \mathbb{G}$.
- Compute $\pi_i := r_i(2m_i - 1) \cdot G + r_i^2 \cdot H \in \mathbb{G}$.
- Compute $C_{out} = 1 \cdot G + 0 \cdot H \in \mathbb{G}$.
- For each gate $\overline{m_i \wedge m_j} = m_k$, use the trapdoor s to open C_k as $1 \cdot G + r_i' \cdot H$.
- Compute $C_{ijk} = m_{ijk} \cdot G + r_{ijk} \cdot H := C_i + C_j + 2 \cdot C_k - 2 \cdot G$.
- Compute $\pi_{ijk} := r_{ijk}(2m_{ijk} - 1) \cdot G + r_{ijk}^2 \cdot H \in \mathbb{G}$. Use $m_i = m_j = 0, m_k' = 1$.
- Output $((C_i)_i, (\pi_i)_i, (\pi_{ijk})_{ijk})$.

Summary of BGN-based NIZK

- $O(N)$ proof size and verification time for CSAT (hence all NP).

	Binding setup	Hiding setup
Knowledge soundness	Perfect	Computational
Zero-knowledge	Computational	Perfect

- Composite order symmetric pairings $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, order $n = pq$.

- Subgroup hiding assumption $\{G, ps \cdot G\} \approx_c \{G, s \cdot G\}$.

Agenda

- Non-interactive zero-knowledge (NIZK) definitions ✓

Pairing-based constructions of NIZK

- From reasonable cryptographic assumptions
  - The BGN cryptosystem ✓
  - BGN bit proofs ✓
  - BGN proofs for CSAT ✓
- **From strong cryptographic assumptions**
  - Arithmetisation of R1CS into QAP
  - Polynomial IOP and pairing-based compiler

$O(N)$ proof size for Boolean circuits

$O(1)$ proof size for Arithmetic circuits

From R1CS to strong R1CS

$$\mathcal{R}_{R1CS} = \left\{ \left((\mathbb{F}, A, B, C, \vec{x}), \vec{w}\right): \begin{array}{c} A, B, C \in \mathbb{F}^{N_r \times N_c}, \vec{x} \in \mathbb{F}^k \\ \vec{w} \in \mathbb{F}^{N_c - k}, \vec{z} \coloneqq \vec{x} || \vec{w} \\ A\vec{z} \circ B\vec{z} = C\vec{z} \end{array} \right\}.$$

entry-wise product

$\vec{x}$ makes the problem non-trivial. W.L.O.G first entry is 1.

Definition: *strong* R1CS instances are as above, and additionally, if $\vec{z}_i \coloneqq \vec{x}||\vec{w}_i$ for $i \in [3]$, $A\vec{z}_1 \circ B\vec{z}_2 = C\vec{z}_3$ implies that $\vec{z}_1 = \vec{z}_2 = \vec{z}_3$.

Lemma: for each R1CS instance, there is a *strong* R1CS instance with exactly the same witnesses and dimensions $N_r + 2N_c, N_c$.

Proof:

$$\begin{pmatrix} A \\ I_{N_c} \\ 1^{N_C} \quad 0_{N_c \times (N_c - 1)} \end{pmatrix} \vec{z}_1 \circ \begin{pmatrix} B \\ 1^{N_C} \quad 0_{N_c \times (N_c - 1)} \\ I_{N_c} \end{pmatrix} \vec{z}_2 = \begin{pmatrix} C \\ I_{N_c} \\ I_{N_c} \end{pmatrix} \vec{z}_3$$

$$\begin{pmatrix} A\vec{z}_1 \\ 1^{N_C} \\ \vec{z}_1 \end{pmatrix} \circ \begin{pmatrix} B\vec{z}_2 \\ \vec{z}_2 \\ 1^{N_C} \end{pmatrix} = \begin{pmatrix} C\vec{z}_3 \\ \vec{z}_3 \\ \vec{z}_3 \end{pmatrix}$$

Polynomial definitions and facts

- Let $H \subseteq \mathbb{F}$ with $|H| = N$.

$$L_{h,H}(\omega) = (\omega == h)$$

Definition:

- The *Lagrange polynomials* on H are defined, for $h \in H$, by

$$L_{h,H}(X) \coloneqq \prod_{h' \in H \setminus \{h\}} \frac{X - h'}{h - h'}$$

Degree $|H| - 1$.

- The *vanishing polynomial* on H is defined as $v_H(X) \coloneqq \prod_{h \in H}(X - h)$.

Fact:

Degree $|H|$.

For $f \in \mathbb{F}[X]$, we have $f(h) = 0 \; \forall h \in H \Leftrightarrow v_H(X) \mid f(X)$.

R1CS as polynomial divisibility

- Choose $H = \{1, \dots, N_r\} \subseteq \mathbb{F}$ (there are better choices).
- For each $j \in [N_c]$, define $a_j(X) \coloneqq \sum_{i \in [N_r]} a_{ij} L_{i,H}(X)$.
- Define $b_j(X)$, $c_j(X)$ similarly.
- Let $\vec{z} = (z_1, \dots, z_{N_c})$ be an R1CS witness.
- Define $A(X) \coloneqq \sum_{j \in [N_c]} z_j a_j(X)$ and $B(X), C(X)$ similarly.

$A = (a_{i,j})$

Note that $a_j(i) = a_{i,j}$.

Lemma: $v_H(X) \mid A(X) \cdot B(X) - C(X) \iff A\vec{z} \circ B\vec{z} = C\vec{z}$.

Proof: $v_H(X) \mid A(X) \cdot B(X) - C(X) \Leftrightarrow A(X) \cdot B(X) - C(X)$ vanishes on H.

For each $i \in H = [N_r]$,

$$A(i)B(i) - C(i) = \left(\sum_{j \in [N_c]} z_j a_j(i)\right)\left(\sum_{j \in [N_c]} z_j b_j(i)\right) - \left(\sum_{j \in [N_c]} z_j c_j(i)\right)$$

$$= \left(\sum_{j \in [N_c]} z_j a_{ij}\right)\left(\sum_{j \in [N_c]} z_j b_{ij}\right) - \left(\sum_{j \in [N_c]} z_j c_{ij}\right) = (A\vec{z})_i (B\vec{z})_i = (C\vec{z})_i.$$

ETH Zurich, Department of Computer Science — Dr. Jonathan Bootle
HS 2023 — Karen Klein, Varun Maram, Antonio Merino Gallardo

Zero-Knowledge Proofs
Exercise 12

12.1 Feasibility of Non-Interactive Zero-Knowledge

Explain why the argument in the lectures (that if L has a non-interactive zero-knowledge proof *without a CRS*, then $L \in BPP$) does not apply when the NIZK has adaptive zero-knowledge *with CRS*, as defined in the lectures.

12.2 Non-Interactive Proofs of Quadractic Non-Residuosity

Given an odd prime integer p, recall that the Legendre symbol of $a \in \mathbb{Z}_p^*$, denoted $\left(\frac{a}{p}\right) \in \{-1, 1\}$, is computed as $a^{(p-1)/2} \bmod p$ and indicates whether a is a quadratic residue modulo p (i.e., whether there exists $b \in \mathbb{Z}_p^*$ such that $a = b^2 \bmod p$).

Let n be the product of two distinct primes p and q. The set of quadratic residues modulo n is further denoted QR_n. The Jacobi symbol of $a \in \mathbb{Z}_n^*$, denoted $J_n(a)$, is defined as $\left(\frac{a}{p}\right) \cdot \left(\frac{a}{q}\right) \in \{-1, 1\}$. Note that a Jacobi symbol can be computed in polynomial time. Denote by J_n^+ the subset of Z_n^* of elements of Jacobi symbol 1 and J_n^- the subset of elements of Jacobi symbol -1.

a) Show that $|J_n^+| = |J_n^-|$ if $p = q = 3 \bmod 4$ (i.e., n is a so-called Blum integer).

An integer z is said to be *regular* if we have $|J_z^+| = |J_z^-|$. A regular integer is said to be *k-regular*, for $k \geq 1$, if it is the product of k distinct primes. We denote by $R(k)$ the set of k-regular integers. The previous question shows that all Blum integers are 2-regular.

Let $L_{2,\overline{\mathrm{QR}}}$ denote the set

$$\left\{(n, a) \in \mathbb{Z}_{\geq 0}^2 \colon n \in R(2) \wedge a \in J_n^+ \wedge a \notin \mathrm{QR}_n\right\}.$$

Deciding whether a pair (n, a) is in this language is not a trivial task as a standard computational assumption in cryptography is precisely if n is a Blum integer, then it is hard to distinguish quadratic residues (modulo n) from non-quadratic residues with Jacobi symbol 1.

Let $(n, a) \in L_{2,\overline{\mathrm{QR}}}$.

b) Show that if $r \in J_n^+$, then either r or ar is a quadratic residue modulo n.

This observation suggests the following non-interactive proof system for $L_{2,\overline{\mathrm{QR}}}$.

- The prover and the verifier are given as input a pair $(n, a) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}_n^*$ (set $\lambda := \lfloor \log n \rfloor + 1$) as well as a uniformly random λ^3-bit reference string $r_1 r_2 \cdots r_{\lambda^2}$ with each r_i of length λ.
- For $i \in [\![\lambda^2]\!]$, if $r_i \in J_n^+$ (the bit string r_i is identified with the integer it represents), then the prover computes and sends a uniformly random value y_i such that $r_i = y_i^2 \bmod n$ or $ar_i = y_i^2 \bmod n$.

– Upon receiving the proof, the verifier proceeds as follows.
  1. If $r_i \in J_n^+$ for less than 3λ indices i then accept.
  2. If n is not odd or $a \notin J_n^+$ then reject.
  3. If n is a perfect square then reject.
  4. If n is a prime power then reject.
  5. If $r_i = y_i^2 \bmod n$ or $ar_i = y_i^2 \bmod n$ for each y_i received from the prover then accept, otherwise reject.

  To check whether n is a prime power, assuming the existence of a randomized (with a random tape of the same length as the input tape) primality-test algorithm $\mathsf{PrimeTest}$ with perfect correctness and soundness error $3/8$[1], the verifier can proceed as follows.

  1. Compute the largest integer α such that $n = m^\alpha$ for some positive integer m. The verifier need only loop over the value $\{1, \ldots, \lfloor \log n \rfloor + 1\}$ for α, and do a binary search to find m, should it exist.
  2. Let z be such that $z^\alpha = n$.
  3. If $\mathsf{PrimeTest}(z; r_i) = 1$ for all $i \in [\![\lambda^2]\!]$, reject.

c) Show that the protocol has negligible (in λ) completeness error.

d) What is the probability that the last check from the verifier succeeds if n is 2-regular but $a \in \mathrm{QR}_n$?

e) Show that an odd integer is regular if and only if it is *not* a perfect square.

f) What is the probability that the third check of the verifier succeeds if n is not regular?

g) Consider a binary relation $\sim$ over $\mathbb{Z}_n^*$ defined as $a_1 \sim a_2$ if and only if $a_1 a_2 \in \mathrm{QR}_n$. Show that it is an equivalence relation.

h) Show that an odd integer n is k-regular for $k \geq 1$ if and only if it is regular and $\mathbb{Z}_n^*$ is partitioned in 2^k $\sim$ equivalence classes of the same cardinality.

i) What is the probability that the fourth check of the verifier succeeds if n is 1-regular?

j) For any fixed (n, a) such that n is k-regular for $k \geq 3$, show that for each $r_i \in J_n^+$, r_i or ar_i is in QR_n with probability at most $1/2$.

k) Show that the last check of the verifier succeeds with probability at most $2^{-\lambda}$ if n is k-regular for $k \geq 3$.

l) Conclude that the proof has negligible soundness error.

m) Show that the proof is zero-knowledge by designing a polynomial-time simulator that computes a common-reference string (after getting the instance as input) with the same distribution as in the proof system and can thereby compute proofs with the same distribution as those of the prover.

[1] That is, the probability that it declares a composite number to be prime is at most $3/8$.

ETH Zurich, Department of Computer Science — Dr. Jonathan Bootle
HS 2023 — Karen Klein, Varun Maram, Antonio Merino Gallardo

Zero-Knowledge Proofs
Exercise 12

12.1 Feasibility of Non-Interactive Zero-Knowledge

Explain why the argument in the lectures (that if L has a non-interactive zero-knowledge proof *without a CRS*, then $L \in BPP$) does not apply when the NIZK has adaptive zero-knowledge *with CRS*, as defined in the lectures.

Solution: In the lectures, we saw that if there was a non-interactive zero-knowledge proof system (P, V) without a *common reference string* for a language L, then $L \in BPP$. This was because given an efficient simulator $S(x)$ producing proofs indistinguishable from the prover's proofs for $x \in L$, one could use S and V to build an efficient decider for L, by first running S to produce a proof, and then using V (with fresh randomness) to verify the proof. If $x \in L$, then S would produce a proof π indistinguishable from that of P (by zero-knowledge), which would then be accepted by the verifier (by completeness). If $x \notin L$, then by the soundness of (P, V), the proof produced by S would be rejected.

The definition of adaptive zero-knowledge says that a non-interactive proof system (K, P, V) for a relation R is zero-knowledge if there exists an efficient simulator $S = (S_1, S_2)$ such that for all $(x, w) \in R$, the distribution of common reference strings σ produced by $(\sigma, \tau) \leftarrow S_1(1^\lambda)$ and proofs $\pi \leftarrow S_2(\sigma, x, \tau)$ is indistinguishable from σ and π produced by the CRS generator K and prover P. As before, the decider given above will still work when $x \in L$. What if $x \notin L$? This time, since S_1 samples common reference strings independently of x, we know that simulated common reference strings σ are indistinguishable from real ones. However, since the soundness guarantees for (K, P, V) do not include the simulation trapdoor τ, we cannot guarantee that V will reject a proof π produced by S_2 on inputs (σ, τ, x).

Note that if (K, P, V) has perfect soundness, then there are no *correctly generated* common reference strings σ with associated proofs π that cause the verifier to accept when $x \notin L$. Even so, we cannot guarantee that the verifier will reject simulated proofs, because if (K, P, V) has computational zero-knowledge, then the set of all possible common reference strings produced by the simulator may contain many more strings than just the honestly generated ones, for which there may or may not be accepting proofs.

12.2 Non-Interactive Proofs of Quadractic Non-Residuosity

Given an odd prime integer p, recall that the Legendre symbol of $a \in \mathbb{Z}_p^*$, denoted $\left(\frac{a}{p}\right) \in \{-1, 1\}$, is computed as $a^{(p-1)/2} \bmod p$ and indicates whether a is a quadratic residue modulo p (i.e., whether there exists $b \in \mathbb{Z}_p^*$ such that $a = b^2 \bmod p$).

Let n be the product of two distinct odd primes p and q. The set of quadratic residues modulo n is further denoted QR_n. The Jacobi symbol of $a \in \mathbb{Z}_n^*$, denoted $J_n(a)$, is defined as $\left(\frac{a}{p}\right) \cdot \left(\frac{a}{q}\right) \in \{-1, 1\}$. Note that a Jacobi symbol can be computed in polynomial time.

Denote by J_n^+ the subset of Z_n^* of elements of Jacobi symbol 1 and J_n^- the subset of elements of Jacobi symbol -1.

a) Show that $|J_n^+| = |J_n^-|$ if $p = q = 3 \bmod 4$ (i.e., n is a so-called Blum integer).
Solution: We start by proving a slightly more general fact: If $J_n^- \neq \emptyset$ then $|J_n^+| = |J_n^-|$. If J_n^- is not empty, we can find an element a in it. Define the following map:

$$\begin{aligned} J_n^+ &\to J_n^- \\ x &\mapsto ax \end{aligned}$$

This map is well defined, since if $x \in J_n^+$ then $\left(\frac{ax}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{x}{n}\right) = -1 \cdot 1 = -1$. It also is a bijection, since it has an inverse map $x \mapsto a^{-1}x$ (note that an element with Jacobi symbol $\neq 0$ necessarily has an inverse).
It remains to show that $J_n^- \neq \emptyset$. Note that modulo q we have that -1 is a QNR, since by Euler's criterion $\left(\frac{-1}{q}\right) \equiv (-1)^{\frac{q-1}{2}} \equiv -1 \pmod q$ since $q \equiv 3 \pmod 4$. Next, note that since $p \neq q$ and they are both primes, we can, by the Chinese Remainder Theorem, find an $a \in \mathbb{Z}_n$ such that

$$\begin{aligned} a &\equiv 1 \pmod p \\ a &\equiv -1 \pmod q \end{aligned}$$

And as such $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right) = 1 \cdot -1 = -1$ so $a \in J_n^-$.

An odd integer z is said to be *regular* if we have $|J_z^+| = |J_z^-|$. A regular integer is said to be *k-regular*, for $k \geq 1$, if it is the product of powers of k distinct primes. We denote by $R(k)$ the set of k-regular integers. The previous question shows that all Blum integers are 2-regular.
Let $L_{2,\overline{\mathrm{QR}}}$ denote the set

$$\left\{(n,a) \in \mathbb{Z}_{\geq 0}^2 \colon n \in R(2) \wedge a \in J_n^+ \wedge a \notin \mathrm{QR}_n\right\}.$$

Deciding whether a pair (n,a) is in this language is not a trivial task as a standard computational assumption in cryptography is precisely if n is a Blum integer, then it is hard to distinguish quadratic residues (modulo n) from non-quadratic residues with Jacobi symbol 1.
Let $(n,a) \in L_{2,\overline{\mathrm{QR}}}$.

b) Show that if $r \in J_n^+$, then either r or ar is a quadratic residue modulo n.
Solution: If $r \in \mathrm{QR}_n$, we are done. So, let us assume that $r \notin \mathrm{QR}_n$. Then it must be that $\left(\frac{r}{p}\right) = \left(\frac{r}{q}\right) = \left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$. Then $\left(\frac{ar}{p}\right) = 1$ and $\left(\frac{ar}{q}\right) = 1$ and so $ar \in \mathrm{QR}_n$ as desired.

This observation suggests the following non-interactive proof system for $L_{2,\overline{\mathrm{QR}}}$.

- The prover and the verifier are given as input a pair $(n,a) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}_n^*$ (set $\lambda := \lfloor \log n \rfloor + 1$) as well as a uniformly random λ^3-bit reference string $r_1 r_2 \cdots r_{\lambda^2}$ with each r_i of length λ.
- For $i \in [\![\lambda^2]\!]$, if $r_i \in J_n^+$ (the bit string r_i is identified with the integer it represents), then the prover computes and sends a uniformly random value y_i such that $r_i = y_i^2 \bmod n$ or $ar_i = y_i^2 \bmod n$.[1]

[1] If we want the prover to run efficiently, we can change the protocol to give the prover the factors p, q of n as additional input. Note that given the factorization of $n = pq$ one can efficiently compute all four square roots of a quadratic residue modulo n.

– Upon receiving the proof, the verifier proceeds as follows.
  1. If $r_i \in J_n^+$ for less than 3λ indices i then reject.
  2. If n is not odd or $a \notin J_n^+$ then reject.
  3. If n is a perfect square then reject.
  4. If n is a prime power then reject.
  5. If $r_i = y_i^2 \bmod n$ or $ar_i = y_i^2 \bmod n$ for each y_i received from the prover then accept, otherwise reject.

  To check whether n is a prime power, assuming the existence of a randomized (with a random tape of the same length as the input tape) primality-test algorithm $\mathsf{PrimeTest}$ with perfect correctness and soundness error $3/8$[2], the verifier can proceed as follows.
  1. Compute the largest integer α such that $n = m^\alpha$ for some positive integer m. The verifier need only loop over the value $\{1, \ldots, \lfloor \log n \rfloor + 1\}$ for α, and do a binary search to find m, should it exist.
  2. Let $z > 1$ be such that $z^\alpha = n$.
  3. If $\mathsf{PrimeTest}(z; r_i) = 1$ for all $i \in [\![\lambda^2]\!]$, reject.

c) Show that the protocol has negligible (in λ) completeness error.

Solution: We analyse each step in sequence.

1. This first check does not affect completeness.
2. A valid input will always meet those two conditions
3. A product of two distinct primes is not a square so this will always pass
4. For this we have to consider the algorithm for prime power testing. Fix a z, since n is not a prime power, the algorithm will result in a false positive with probability $\frac{3}{8}$, and repeating it λ^2 times yields a false positive probability of at most $(\frac{3}{8})^{\lambda^2}$. This was for a single fixed z, so using a union bound the completeness error is at most $2^\lambda(\frac{3}{8})^{\lambda^2}$ which is negligible.
5. Finally, by part b) we know that one of the two conditions holds, and so we are done.

d) What is the probability that the last check from the verifier succeeds if n is 2-regular but $a \in \mathrm{QR}_n$?

Solution:
If $r \in \mathrm{QR}_n$ then the test will pass. In the other case, similar reasoning to part b) shows that $ar \notin \mathrm{QR}_n$. So the probability that the test passes despite $a \in \mathrm{QR}_n$ will be at most the probability that $r \in \mathrm{QR}_n$, which is at most $\frac{1}{2}$ (in fact it will be less than that). This will be repeated for at-least 3λ values, and taking the union bound over all possible input moduli n yields a final probability of $2^\lambda(\frac{1}{2})^{3\lambda} = 2^{-2\lambda}$ which again is negligible.

e) Show that an odd integer is regular if and only if it is *not* a perfect square.

Solution:
Let n be odd. Recall that an integer n is regular iff $|J_n^+| = |J_n^-|$. We show that this is equivalent to n not being a perfect square. First, if n is a perfect square (say $n = m^2$) then we have that

$$\left(\frac{x}{n}\right) = \left(\frac{x}{m}\right)^2 = 1$$

for every x with non zero Jacobi symbol in m, so $|J_n^-| = 0$ while $|J_n^+| \geq 1$, so n cannot be regular. Conversely, let us assume that n is not a perfect square. Then we can decompose $n = p_1^{\alpha_1} \ldots p_k^{\alpha_k}$ into prime factors, where at least one of those α_i

[2] That is, the probability that it declares a composite number to be prime is at most 3/8.

is not even. Without loss of generality, assume it to be α_1. Note that $\mathrm{QNR}_{p_1} \neq \emptyset$, since p_1 is prime and $\neq 2$. Let $a \in \mathrm{QNR}_{p_1}$, and use the Chinese Remainder Theorem to find an x such that $x \equiv a \pmod{p_1^{\alpha_1}}$ and $x \equiv 1 \pmod{{p_i}^{\alpha_i}}$ for $i \neq 1$. Then we have that:

$$\begin{aligned}\left(\frac{x}{n}\right) &= \left(\frac{a}{p_1}\right)^{\alpha_1}\left(\frac{1}{p_2}\right)^{\alpha_2}\cdots\left(\frac{1}{p_k}\right)^{\alpha_k} \\ &= \left(\frac{a}{p_1}\right)^{\alpha_1} \\ &= (-1)^{\alpha_1} = -1\end{aligned}$$

The last line follows from the fact that α_1 is odd. Therefore, $x \in J_n^-$ and this, by the solution of part a), shows that n is regular.

f) What is the probability that the third check of the verifier succeeds if n is not regular?

Solution:

If n is not regular, we have shown in part e) that it must be a perfect square, which this test will always detect.

g) Consider a binary relation $\sim$ over $\mathbb{Z}_n^*$ defined as $a_1 \sim a_2$ if and only if $a_1 a_2 \in \mathrm{QR}_n$. Show that it is an equivalence relation.

Solution:

Reflexivity follows at once as $a \cdot a = a^2 \in \mathrm{QR}_n$ by definition. Symmetry follows by commutativity of multiplication. Finally for transitivity, assume $a \sim b$ and $b \sim c$. Then $ab, bc \in \mathrm{QR}_n$. Assume that $ab = s^2, bc = r^2$. Then $ac = \frac{ab \cdot bc}{b^2} = \frac{r^2 s^2}{b^2} = \left(\frac{rs}{b}\right)^2 \in \mathrm{QR}_n$

h) Show that an odd integer n is k-regular for $k \geq 1$ if and only if it is regular and $\mathbb{Z}_n^*$ is partitioned in $2^k \sim$ equivalence classes of the same cardinality.

Solution:

Let $n = \prod_{i=1}^{k} p_i^{\alpha_i}$. Define the function $bin : \mathbb{Z}_n^* \to \{-1, +1\}^k$ to take:

$$x \mapsto \left(\frac{x}{p_1}\right) || \ldots || \left(\frac{x}{p_k}\right)$$

We claim that this function is constant on equivalence classes and that it is surjective. Since the Legendre symbol is multiplicative, we have that $bin(ab) = bin(a) \cdot bin(b)$, where this multiplication is done pointwise. Consequently, $bin(a^2) = 1^k$ and so $bin(a) = 1^k \iff a \in \mathrm{QR}_n$. Now, suppose that $a \sim b$. Then $ab \in \mathrm{QR}_n$ and as such $1^k = bin(ab) = bin(a)bin(b)$ which then implies that $bin(a) = bin(b)$. If instead $bin(a) = bin(b)$ then $bin(ab) = 1^k$ and then $a \sim b$. Let now $\alpha \in \{-1, +1\}^k$. We show that we can find x with $bin(x) = \alpha$. Write α_i for the i-th bit of α. Then by the Chinese Remainder Theorem we can find x such that $x \equiv \alpha_i \pmod{p_i}$. Then, it will be that $bin(x) = \alpha$ by definition of the Legendre symbol. With this, we have shown that $\sim$ divides $\mathbb{Z}_n^*$ into 2^k equivalence classes. What is left to show is that they have the same size. By the above argument, we have identified equivalence classes with binary string. Let α, β be the strings representing two classes. Then, let $\gamma = \beta\alpha$. By surjectivity, find an element a that has $bin(a) = \gamma$. Then consider the map $x \mapsto ax$. If $bin(x) = \alpha$ then clearly $bin(ax) = bin(a)bin(x) = \beta\alpha\alpha = \beta$ so this map maps elements of the first class to the second one. It is a bijection, so we are done. The above was independent of regularity, and together with the fact that the number of classes is exactly the number of distinct prime factors of n, implies the result.

i) What is the probability that the fourth check of the verifier succeeds if n is 1-regular?

Solution:

If n is 1-regular, it is a prime power. Then, the check will detect this.

j) For any fixed (n, a) such that n is k-regular for $k \geq 3$, show that for each $r_i \in J_n^+$, r_i or ar_i is in QR_n with probability at most $1/2$.

Solution:
First, recall from part h) that $\mathbb{Z}_n^*$ is partitioned into 2^k equivalence classes of the same size. A quick thought shows that in fact exactly half of them will have Jacobi symbol 1. Assume that $r_i \in J_n^+$ and that is uniformly distributed. Then, the probability of $r_i \in \mathrm{QR}_n$ is exactly $\frac{1}{2^{k-1}} \leq \frac{1}{4}$. Similarly, the probability of $ar_i \in QR_n$ is less than $\frac{1}{4}$. So the probability that either of them happens is less than $\frac{1}{2}$

k) Show that the last check of the verifier succeeds with probability at most $2^{-\lambda}$ if n is k-regular for $k \geq 3$.

Solution:
By the above, the probability that the check can be satisfied for a given r_i is at most $\frac{1}{2}$. Repeating it at-least 3λ times yields the the desired result.

l) Conclude that the proof has negligible soundness error.

Solution:
Suppose that $(n, a) \notin L_{2,\overline{\mathrm{QR}}}$. First, let us tackle the case in which n is not 2-regular. If it is even, it will be rejected with probability 1. If n is not regular, by part f) it will always be rejected. If instead it is 1-regular, it will be also be always be detected by part i). Finally, if n is k-regular with $k \geq 3$ then by part k) the verifier will only be fooled with negligible probability. Instead, if n is 2-regular but a is a quadratic residue, then by part d) the prover will succeed with negligible probability.

m) Show that the proof is non-adaptive zero-knowledge by designing a polynomial-time simulator that computes a common-reference string (after getting the instance as input) with the same distribution as in the proof system and can thereby compute proofs with the same distribution as those of the prover.

Solution:
Consider the following simulator, on input $(n, a) \in L_{2,\overline{\mathrm{QR}}}$. Randomly select λ^2 integers $s_i \in \mathbb{Z}_n$. If $s_i \notin J_n^+$, simply set $r_i \coloneqq s_i$. Else, flip a coin, and depending on the outcome, either set $r_i \coloneqq s_i^2$ or $r_i \coloneqq a^{-1}s_i^2$. The final reference string is the string $r_1 \ldots r_{\lambda^2}$. The string is uniformly distributed (as desired) and the simulator can then compute a proof by sending the s_i.