The Clean Paper

WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

Cryptographic proof zai iya ɓoye sirrinsa ta sa missing simulator ya zama mai wahalar prove

Zero-knowledge proofs suna ba mutum damar prove statement ba tare da bayyana witness ba. Classical theory tana cewa ba za ka iya samun wannan, a full sense, da message guda, babu trusted setup kuma perfect soundness ba. Paper na Rahul Ilango bai sa wannan impossibility ya ɓace ba. Yana canza target: maimakon buƙatar simulator ya wanzu da gaske, yana tambayar cewa chosen proof system ba zai iya efficiently prove cewa simulator ba ya wanzuwa ba. A ƙarƙashin manyan assumptions na proof complexity da cryptography, wannan ya isa a dawo da falsifiable, game-based consequences na zero-knowledge property by property. Point ɗin ba plug-in internet primitive ba ne. Proof-theoretic hanya ce ta juya mathematical unprovability zuwa cryptographic cover.

Written by Cinzia Vaglio · Cross-checked by Vera Rubin · Edited by Michele Renda · Figures by Nova Vela · AI-assisted, human-reviewed

Paper: *Gödel in Cryptography: Effectively Zero-Knowledge Proofs for NP with No Interaction, No Setup, and Perfect Soundness*, Rahul Ilango, FOCS 2025 / IACR ePrint 2025/1296 · DOI: 10.1109/FOCS63196.2025.00058

Dabarar ba ita ce a tabbatar cewa sirrin ya ɓoye ba

Mu fara da mafi sauƙin sigar zero-knowledge.

Alice tana son ta gamsar da Bob cewa wani Sudoku puzzle yana da mafita. Idan ta tura masa mafitar, Bob zai gamsu, amma puzzle ɗin ya lalace. Abin da take so ya fi ban mamaki: proof cewa mafita tana wanzuwa, ba tare da ta bayyana mafitar ba.

Wannan shi ne alkawarin **zero-knowledge proof**. Prover (Alice) yana gamsar da verifier (Bob) cewa statement gaskiya ne, alhali bai bayyana wani abu fiye da gaskiyar statement ɗin ba.

Matsalar ita ce wannan alkawari yana da farashi. Ordinary mathematical proof yana da comfortable features biyu. Na farko, **one message** ne: ka rubuta shi, ka miƙa, ka tafi. Na biyu, yana da **perfect soundness**: false statement ba shi da valid proof kwata-kwata. Classical impossibility sakamako suna cewa zero-knowledge dole ya bar duka waɗannan features — kuma ba wai idan an haɗa su biyu kawai ba; kowannensu ma shi kaɗai yana hana classical zero-knowledge.

Na farko, zero-knowledge proof yana buƙatar conversation. Idan Alice ta tura message guda ɗaya, ba tare da trusted setup da aka shirya tun farko ba, zero-knowledge guarantee ɗin ya rushe — komai yawan soundness da kake shirye ka sadaukar domin musayar wannan.

Na biyu, zero-knowledge proof yana buƙatar ɗan tolerance ga error. Neman perfect soundness yana ternyata lalata interaction a hankali: verifier da ba zai taɓa yarda da ƙarya ba, ko wane random choices ya yi, zai iya kawai ya kulle waɗannan choices tun farko — kuma da zarar verifier ya zama predictable, Alice za ta iya ba da dukan answers a message guda, wanda shi ne case ɗin da ya riga ya karya.

Takardar Rahul Ilango tana neman hanyar wuce wannan bango biyu. Ba ta hanyar pretending cewa bangon ba ya nan ba, kuma ba ta hanyar samar da classical zero-knowledge a setting da aka san ba zai yiwu ba. Motsin ya fi subtle: a raunana abin da “reveals nothing” yake nufi, amma a raunana shi ta hanyar da ta riƙe security properties da cryptographers za su iya gwadawa a zahiri.

Ana kiran sakamakon **effectively zero-knowledge**.

A proof without leaking the witness

The paper keeps the ordinary-proof shape by weakening what privacy must prove.

blocked door 1

interaction

blocked door 2

trusted setup

blocked door 3

imperfect soundness

the route

the rulebook cannot efficiently refute the simulator

Boundary: not classical zero-knowledge; effectively zero-knowledge under a chosen proof system.

Zero-knowledge ya toshe a ƙofofi uku — interaction, trusted setup, da imperfect soundness. Construction na llango ya bi wata ƙofa daban: rulebook ɗin ba zai iya efficiently refute simulator ɗin ba.

Original diagram — The Clean Paper CC BY 4.0

Classical privacy vs effective privacy

The relaxation is the point: the paper changes what the privacy claim has to establish.

classical zero-knowledge

positive claim

A simulator exists and can fake the verifier's view without the witness.

effectively zero-knowledge

weaker claim

Your chosen proof system cannot efficiently prove that no simulator exists.

Boundary: the construction preserves testable consequences, not the full simulator guarantee.

Classical zero-knowledge yana tambayar ko simulator yana wanzuwa; “effectively zero-knowledge” yana tambayar kawai ko chosen rulebook ɗinka zai iya efficiently prove cewa ba zai iya wanzuwa ba. Wannan

weaker question shi ne ya ba construction damar riƙe one message, no setup da perfect soundness.

Original diagram — The Clean Paper CC BY 4.0

Tsohon gwaji: simulator yana wanzuwa

Classical hanyar formalize zero-knowledge tana amfani da fictional helper da ake kira **simulator**.

Ra'ayin shi ne wannan: ka yi tunanin Jane, wadda **ba ta san** sirrin Alice ba. Idan Jane za ta iya generate, ita kaɗai, proofs da suke kama da proofs ɗin da Bob zai karɓa daga Alice, to proofs ɗin Alice ba su koya wa Bob wani sabon abu ba. Jane ta riga ta iya fake experience ɗin ba tare da sirrin Alice ba.

Saboda haka classical zero-knowledge yana neman ainihin simulator. Dole ne a sami efficient algorithm da zai iya samar da fake-looking proofs ba tare da sanin sirrin ba — wato *witness* a jargon; ga Sudoku, witness kawai solved grid ne.

Wannan definition tana da ƙarfi, amma ita ce daidai inda tsohon impossibility yake cizo. Intuition ɗin shi ne: truly non-interactive proof string ce kawai. Da zarar Bob ya sami wannan string, zai iya nuna ta ga wani: ya sami ikon prove statement ɗin ga wasu, wanda tuni ya yi kama da fiye da "nothing." Classical theorems suna sharpen wannan intuition zuwa impossibilities da aka ambata a sama.

Properties uku da wannan takarda ta nace a kansu

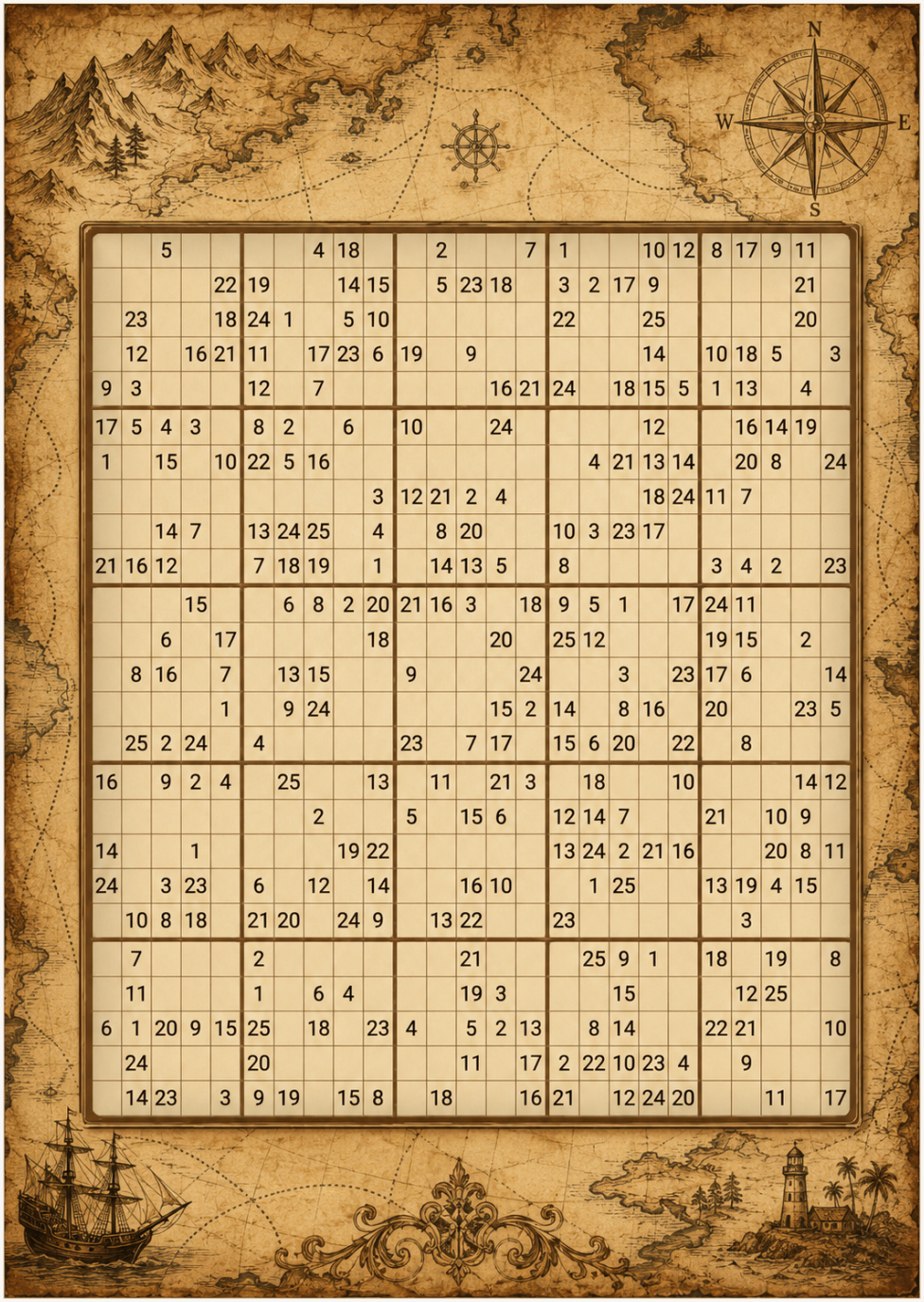
Taken takardar ya ambaci constraints uku:
No interaction: Alice tana tura proof string guda ɗaya. Babu protocol na kai-komo.
No setup: Alice da Bob ba sa dogaro da trusted common reference string ko wani pre-arranged public randomness. Systems da yawa da ake kira "non-interactive zero-knowledge" har yanzu suna dogaro da setup; a nan takardar tana nufin zero setup.
Perfect soundness: false statement ba shi da valid proof. Ba "kusan ba a taɓa accept" ba; valid proof ɗin kwata-kwata ba ya wanzuwa.
Waɗannan properties uku su ne abubuwan da ordinary written mathematics take da su — kuma, kamar yadda aka bayyana a sama, classical zero-knowledge ba zai iya riƙe su ba.

Bambancin ta hanyar MegaSudoku

Ga wata deliberately simplified hanyar jin bambancin.

Kada mu yi amfani da ordinary 9-by-9 Sudoku ga serious part na analogy. Ya yi ƙanƙanta kuma finite sosai: computer zai iya solve shi kawai, ko prove cewa ba shi da mafita. Maimakon haka, ka yi tunanin family na **MegaSudoku(n)** puzzles. Mu scale rule ɗin da aka saba: zaɓi block size n, bari $N = n^2$, sannan a gina N by N grid da aka raba zuwa n by n blocks, tare da N symbols. Ordinary Sudoku ƙaramin case ne kawai: $n = 3$, $N = 9$ — 9-by-9 grid, 3-by-3 blocks da symbols tara. Labarin proof complexity yana farawa ne kawai idan an bar n ya girma, kuma grid ɗin zai iya ɗaukar ƙarin gadgets da za su sa ya yi kamar SAT formula da aka sa masa rigar

Sudoku. SAT formula kawai jerin yes/no constraints ne: za ka iya assign true/false values ga variables ta yadda kowane constraint ya gamsu?



Sudoku 25x25: za a iya checking rules ɗinsa ba tare da bayyana finished grid ba — visual stand-in ga proof da ke verify hidden solution, wato witness.

AI-generated editorial thumbnail — The Clean Paper CC BY 4.0

Sudoku da SAT: puzzle ɗaya cikin kaya biyu

Ikirarin cewa Sudoku zai iya “behave like a SAT formula” ba metaphor ba ne. Translation ɗin yana tafiya

duka directions, kuma easy direction za a iya rubuta shi gaba ɗaya.

Daga Sudoku zuwa SAT. SAT yana magana da true/false kawai, don haka a ba shi boolean variable ɗaya ga kowane (row, column, value) triple: *x(r,c,v)* yana nufin "cell a row *r*, column *c* tana ɗauke da value *v*." Sudoku 4-by-4 (blocks 2-by-2, values 1–4) yana buƙatar 4·4·4 = 64 variables; classical 9-by-9 yana buƙatar 729. Kowane Sudoku rule sai ya zama batch na clauses. (Clause OR ne na variables ko negations ɗinsu; dukan formula kuma AND ne na dukan clauses.)

Kowane cell yana riƙe aƙalla value ɗaya — clause ɗaya ga kowane cell:

x(1,1,1) ∨ x(1,1,2) ∨ x(1,1,3) ∨ x(1,1,4)

Kowane cell yana riƙe a mafi yawa value ɗaya — "not both" clause ga kowane pair na values:

¬x(1,1,1) ∨ ¬x(1,1,2) ¬x(1,1,1) ∨ ¬x(1,1,3) … haka har dukan pairs shida.

Kowane row yana ɗauke da kowane value — ga row 1 da value 3: aƙalla sau ɗaya,

x(1,1,3) ∨ x(1,2,3) ∨ x(1,3,3) ∨ x(1,4,3)

kuma a mafi yawa sau ɗaya: ¬x(1,1,3) ∨ ¬x(1,2,3), da sauransu ga kowane pair na cells a row ɗin.

Columns da blocks — batches iri ɗaya; group na cells ne kawai ke canzawa. Ga top-left block da value 2:

x(1,1,2) ∨ x(1,2,2) ∨ x(2,1,2) ∨ x(2,2,2)

sannan pairwise "not both" clauses.

Printed clues — wannan shi ne mafi sauƙin part: kowane clue clause ne mai variable guda ɗaya. Printed 3 a top-left corner ya zama clause

x(1,1,3)

AND na dukan wannan satisfiable ne daidai lokacin da Sudoku yana da solution — kuma satisfying assignment **ita ce** solution ɗin: ka duba waɗanne x(r,c,v) suke true, ka cika grid. Ga 9-by-9 wannan yana zama variables 729 da clauses dubu kaɗan, wanda modern SAT solver zai dispatch cikin milliseconds. Ka lura da clue clause x(1,1,3): tana cewa "wannan cell daidai 3 ce," ba "waɗannan cells duk sun bambanta" ba — wannan asymmetry ɗin ne zai tilasta ƙarin trick ga clue cells a protocol note da ke ƙasa.

Daga SAT zuwa Sudoku. Takardar tana buƙatar opposite, harder direction: idan aka ba ka *arbitrary* SAT formula, ka gina mega-Sudoku da yake da solution daidai lokacin da formula ɗin ma yake da solution. Native rules na Sudoku suna iya cewa kawai "waɗannan cells duk sun bambanta," don haka arbitrary logical constraints dole a *gina* su — wannan shi ne ainihin gadgets. Gadget small pre-fabricated cluster ne na cells, ɗaya ga kowane clause na formula, inda designated cells suke taka rawar variables (symbol ɗin da suke riƙe da shi yana encode true ko false) kuma internal constraints na cluster ɗin an engineering su ta yadda legal fillings ɗinsa kaɗai suke correspond da assignments da ke satisfy clause ɗin. Wannan standard craftsmanship ne daga NP-completeness proofs; ga generalized Sudoku Yato da Seta sun yi shi a 2003. Directions biyu tare suna cewa N-by-N Sudoku da SAT problem ɗaya ne sanye da kaya daban. Wannan shi ne ya ba wannan article — da takardar — izinin ba da labari game da dukan NP ta grids da symbols.

Witness ɗin har yanzu yana da sauƙin gani a tunani. Alice ta san complete valid filling na mega-Sudoku. Bob yana

son ya gamsu cewa irin wannan filling tana wanzuwa, amma Alice ba ta son bayyana ta. Idan ta tura dukan filling, Bob ya gamsu, amma sirrin ya tafi.

A classical zero-knowledge version, Alice da Bob suna interact. Wani old-style mental model yana amfani da covered tiles. Alice tana ɓoye solved grid, tana secretly rename symbols kafin kowane round, sannan ta bar Bob ya inspect random local constraint guda ɗaya: row, column, box, ko gadget. Idan opened cells sun nuna all-different symbols, Bob yana ƙara confidence. Sannan a sake rufe komai kuma a sake fresh renaming na symbols. (Akwai wrinkle ɗaya: given clues na puzzle suna buƙatar extra trick, saboda renaming symbols yana ɓoye su ma. Note ɗin da ke ƙasa yana bayyana yadda classical protocols ke warware wannan; toy picture ɗin ya isa ga abin da zai biyo baya.)

Yadda classical protocols suke handling clue cells a zahiri

Renaming trick ɗin yana da blind spot. Row, column da box rules duk suna cewa "waɗannan cells duk sun bambanta," kuma *all different* yana tsira daga duk wani renaming na symbols. Amma clue yana cewa "wannan cell tana ɗauke da daidai 5," kuma bayan renaming Bob yana ganin $\sigma(5)$ kawai — wani masked symbol — ba tare da sanin renaming σ ba. Ba zai iya checking komai ba. Idan ba a gyara wannan ba, Alice za ta iya prove cewa *wani* valid grid yana wanzuwa tana ignoring printed clues gaba ɗaya, wanda bai prove komai game da *wannan* puzzle ba. Classical literature tana da standard repairs biyu.

Palette. A ƙara extra row guda ɗaya na N cells zuwa hidden grid — palette da Alice ta cika da symbols 1…N a fixed public order, sannan ta rename shi tare da komai, don ya ƙunshi $\sigma(1)\ldots\sigma(N)$. Random challenge na Bob yanzu yana da extra option. Bayan zaɓar row, column, box ko gadget ya buɗe, zai iya zaɓar **palette plus clue cell guda ɗaya**. Alice ta uncover duka; palette ya bayyana renaming na wannan round, kuma Bob ya check cewa clue cell tana nuna daidai renamed version na printed clue. Wannan har yanzu zero-knowledge ne saboda Bob yana koyon σ kawai — wanda ake fresh draw kowane round kuma ba shi da amfani shi kaɗai — da value na cell da ya riga ya sani daga puzzle. Babu wani abu game da secret cells da ya leak, kuma simulator zai iya fake view ta drawing random σ. Yana da soundness saboda cheating Alice ana catching ɗinta da fixed probability kowane round, kuma ana repeat rounds har doubt ya zama negligible.

Compiling clues away. Wani more structural variant yana cire special challenge maimakon ƙara shi. Maimakon *verifying* clue value, a force shi da difference constraints: a link clue cell zuwa kowane palette cell sai wanda yake ɗauke da own value ɗinsa — "different from $\sigma(1)$, different from $\sigma(2)$, …, different from everything but $\sigma(5)$." Symbol ɗaya kaɗai da cell ɗin zai iya legally hold shi ne clue ɗin. Yanzu kowane constraint ya koma "waɗannan biyu sun bambanta" — invariant under renaming, checkable daidai kamar row. Wannan shi ne manoeuvre ɗin da ake amfani da shi ga pre-colored vertices a classical graph-coloring protocol, kuma wannan shi ne ruhin kalmar *gadgets* a sama: a MegaSudoku-as-SAT picture, clues ana compile su zuwa inequality gadgets kamar kowane constraint.

Physical protocol. rayuwar yau da kullum card protocol ga Sudoku (Gradwohl, Naor, Pinkas da Rothblum, 2007) ba ya amfani da renaming kwata-kwata kuma yana settle clues kafin hiding ya fara. Ga kowane cell, Alice tana shimfiɗa identical cards uku da value na cell ɗin — face-down ga secret cells, amma **face-up ga clue cells**, don Bob ya gani da idanunsa cewa clues an respected kafin a flip cards. Sannan card ɗaya daga kowane cell ya shiga packet na row ɗinsa, ɗaya cikin column, ɗaya cikin box; kowane packet ana shuffle kuma ana reveal shi, Bob ya check cewa yana ɗauke da duk N symbols. Shuffling yana lalata position information (wannan shi ne zero-knowledge), amma clues an riga an kulle tun dealing time.

Ko wace hanya, lesson ɗin ɗaya ne da article ɗin ke komawa gare shi: zero-knowledge protocol careful

> bookkeeping ne na *waɗanne* facts suke tsira daga hiding. Renaming yana preserve “all different” kuma yana erase “equals 5” — don haka “equals 5” dole a shigar da shi ta wata hanya.

Wannan ba shi ne protocol ɗin da ke cikin takarda ba. Mental model ne ga classical zero-knowledge:

- Alice da Bob suna kai-komo.
- Bob yana zaɓar random checks.
- Alice tana bayyana local consistency kawai, ba dukan solution ba.
- Proof na privacy yana aiki ta nuna cewa view na Bob zai iya kasancewa an samar da shi ba tare da secret solution na Alice ba.

Saboda haka classical zero-knowledge an gina shi a kan positive fact:

> Simulator yana wanzuwa da gaske.

Yanzu mu cire comfortable parts. Alice tana tura proof string guda ɗaya sannan ta tafi. Babu trusted setup, babu shared random string da aka shirya tun farko, kuma Bob kada ya taɓa accept false puzzle. Wannan shi ne setting da classical zero-knowledge ba zai iya rayuwa a ciki ba.

Ana buƙatar character guda ɗaya kafin trick ɗin. Ka fix **rulebook**: formal proof system, a ma’anar logician — fixed set na axioms tare da mechanical rules na checking written mathematical proofs. ZFC, standard axioms na mathematics, shi ne canonical example. Duk abin da zai biyo baya ana bayyana shi relative ga rulebook da aka zaɓa tun farko, kuma choice ɗin flexible ne: construction ɗin yana aiki ga duk rulebook da ka fix, har da ZFC.

(Note kan words, kamar yadda takardar kansa ya yi: “proof system” a nan koyaushe yana nufin wannan rulebook — formal system da ke checking mathematical proofs — **ba** messages da Alice ke tura ba. Machinery na Alice da Bob ana kiransa “prover da verifier.”)

Gödel-style version yana riƙe MegaSudoku story amma yana canza proof ɗin.

Zaɓi na biyu constraint system mai displayed size ɗaya, mu kira shi **D**. Ga story, S da D MegaSudoku(n) puzzles biyu ne a format ɗaya. Behind the scenes, D zai iya farawa a matsayin hard logical formula mai wani size; idan ya zama dole, za a iya padding shi da harmless dummy constraints har ya dace da grid ɗin. D an gina shi daga logical formula da yake a zahiri **unsatisfiable**: babu assignment na values da zai sa dukan constraints ɗinsa su zama true, kamar broken puzzle da ba shi da legal completed grid. Toy example zai kasance formula da ke neman “X is true” da “X is false” a lokaci guda. Saboda haka D ba shi da valid filling.

Amma D bai kamata ya zama broken puzzle da yake *easy to expose* ba. Toy example da ke sama ya kasa wannan: kowane rulebook zai refute “X and not-X” da line ɗaya. D dole ya zama false ta hanyar da chosen rulebook ba zai iya certify da short argument ba. Idan rulebook zai iya refute D da short proof, story ɗin da ke ƙasa zai rushe: alternative route da zai iya samar da proofs ba tare da secret na Alice ba za a iya formally rule out, kuma tare da shi privacy guarantee ɗin zai tafi. Don haka ana zaɓar D daga family da fixed rulebook ba zai iya efficiently refute ba: babu short proof, a cikin wannan rulebook, cewa D ba shi da solution.

One-message proof na Alice sai ya shafi either/or statement:

> ko real mega-Sudoku S yana da solution, ko decoy D yana da solution.

Wannan shi ne logical link. D **ba** a generate shi ta wata magical hanya da ke sa S ya zama true ba. Proof ɗin ba yana cewa "D ba shi da solution, don haka S yana da solution" ba. Yana proving disjunction **S or D**. Perfect soundness yana cewa false disjunction ba zai iya samun valid proof ba. Tun da D false ne a reality — ba shi da solution — hanyar da disjunction zai zama true ita ce S ya zama true. Don haka idan proof aka accept, S dole ya kasance yana da solution. Decoy ba zai iya sa false S ya zama true ba.

Amma ga zero-knowledge-style part, mu tambayi abin da zai faru idan D *ya kasance* yana da solution. Wannan decoy solution zai yi aiki a matsayin alternative witness. Zai ba wani damar produce proofs ba tare da sanin real mega-Sudoku solution na Alice ba — wato simulator. A reality D ba shi da solution, don haka simulator route ɗin a rufe yake. Point ɗin shi ne rulebook ba zai iya efficiently prove cewa an rufe shi ba.

Saboda haka D yana da jobs biyu. Ga **soundness**, D false ne, don haka valid proof na "S or D" yana force S. Ga **effective zero-knowledge**, D yana da hard to refute, don haka rulebook ba zai iya da sauri rule out decoy route da zai sa simulation ta yiwu ba.

Saboda haka security test ba wannan ba ne kuma:

> Za mu iya prove cewa simulator yana wanzuwa da gaske?

Ya koma:

> Rulebook ɗinka zai iya efficiently prove cewa simulator impossible ne?

Idan answer ɗin no ne, wani surprisingly strong abu yana biyo baya: duk security guarantee da (a) za a iya observe ta running test, kuma (b) za a iya prove — a cikin wannan rulebook — cewa yana biyo daga existence na simulator, a zahiri yana hold. Successful attack a kan kowannensu zai zama missing short refutation da kansa, kuma wannan missing short refutation ba ya wanzuwa. Wannan shi ne "effective" part na effectively zero-knowledge.

Don haka classroom contrast shi ne:

Classical zero-knowledge: proofs suna safe saboda simulator yana wanzuwa.

Gödel-style effective zero-knowledge: proofs ana treating su safe ga observable security tests saboda rulebook ba zai iya efficiently prove cewa simulator impossible ne ba.

ikirari na biyu ya fi rauni. Shi ne kuma dalilin da takardar zai iya riƙe features uku da suka karya classical version: one message, no setup da perfect soundness.

Sabon gwaji: ba za ka iya prove cewa simulator baya nan ba

Relaxation na Ilango yana canza tambayar.

Classical zero-knowledge yana tambaya:

> Simulator yana wanzuwa?

Effectively zero-knowledge yana tambayar weaker thing:

> Chosen rulebook ɗinka zai iya efficiently prove cewa babu simulator?

Wannan yana iya yin kama da technical dodge, amma shi ne core idea. Construction ɗin yana rayuwa a strange state: simulator ba ya wanzuwa a zahiri — takardar ya bayyana wannan — amma rulebook da ka fix ba zai iya efficiently prove cewa ba ya wanzuwa ba. Idan kowane bad consequence da kake damuwa da shi zai buƙaci irin wannan refutation, system ɗin har yanzu yana behave kamar zero-knowledge ga waɗannan consequences.

A nan Gödel yake shiga. Ba a matsayin decoration ba, kuma ba a matsayin "Gödel makes crypto secure." Connection ɗin proof-theoretic ne. Rulebook ana kiransa **optimal** idan, a precise sense, shi ne mafi kyawun mai yiwuwa one: duk lokacin da wani rulebook zai iya refute formula na relevant kind da short proof, optimal rulebook zai iya ma, da proof da bai fi polynomially longer ba. Krajíček da Pudlák sun conjecture a 1989 cewa **no optimal proof system exists**: duk rulebook da ka fix, akwai wani rulebook da zai prove wasu family na true statements da yawa shorter. Wannan ɗaya ne daga central open conjectures na proof complexity, kuma finite, complexity-theoretic cousin ne na Gödel's incompleteness theorem: wasu true statements ba su da short proof a rulebook da ka fix — ba saboda unprovable ne in principle ba, amma saboda kowane fixed rulebook yana barin wasu short truths ba tare da short proofs ba.

takardar ya ɗauka wannan conjecture (a mildly stronger "infinitely often" form, wanda standard ne lokacin da conjectures ake amfani da su cryptographically). Payoff ɗin, ta theorem na Krajíček da Pudlák, concrete ne: ga kowane rulebook akwai sequence na formulas da genuinely unsatisfiable ne, amma rulebook ɗin ba zai iya refute su da short proofs ba — kuma, crucially, efficient algorithm zai iya *generate* su. Wannan last property, uniformity, shi ne ya canza ra'ayin daga existence ikirari zuwa ainihin algorithm da Alice za ta iya run: decoys D nata suna fitowa daga assembly line, ba daga thin air ba.

Cryptographic move ɗin shi ne a sa wannan shortage na proof power ya yi aiki.

Abin da construction ɗin yake yi

Ga construction na takarda, an cire details har ya rage shape ɗinsa.

Fix rulebook — misali ZFC. A ƙarƙashin proof-complexity assumption, akwai efficiently generatable sequence na formulas da actually unsatisfiable ne, amma rulebook ba shi da short proof cewa unsatisfiable ne.

Yanzu gina one-message proof mai wannan form:

> ko real statement satisfiable ne, ko wannan special hard formula satisfiable ne.

Special hard formula ba satisfiable ba ne. Saboda haka idan underlying proof machinery yana perfectly sound, accepting message ɗin har yanzu yana nufin real statement true ne. Wannan ya ba da perfect soundness.

Amma ga zero-knowledge-like security, ka yi tunanin special hard formula *ta kasance* satisfiable. Witness ɗinta zai iya amfani wajen simulate proofs ba tare da sanin real witness ba. Formula ɗin ba satisfiable ba ce a reality — amma rulebook ba zai iya efficiently prove wannan ba. Don haka ba zai iya efficiently prove cewa simulator impossible ne ba.

Wannan shi ne hinge. System ɗin ba ya ɓoye secret ta samar da classical simulator. Yana ɓoye secret, ga large class na observable security tests, a bayan inability na rulebook ya certify cewa simulator baya nan.

Abin da takardar yake ikirari

babban theorem tana zuwa a layers. Core sakamako shi ne wannan:

A ƙarƙashin standard cryptographic assumption — existence na **non-interactive witness indistinguishable proofs**, well-studied objects da ke follow daga several established assumption packages — da kuma proof-complexity conjecture cewa **no (infinitely often) optimal proof system exists**, takardar yana construct, ga kowane choice na rulebook, one-message prover da verifier ga NP/SAT da **perfect soundness** da no setup wanda yake effectively zero-knowledge relative ga wannan rulebook. (NP/SAT shi ne standard "hardest common denominator" na puzzle-like problems; mega-Sudoku ɗaya ne daga kayansa.)

Ga broader ikirari game da preserving falsifiable security properties, takardar yana ƙara standard assumption guda ɗaya, derandomization belief **P = BPP** (roughly: randomness ba ya ba algorithms wani essential extra power).

Idan muka fassara theorem language:

- Proof ɗin message guda ɗaya ne.
- Babu trusted setup.
- False statements ba za a iya prove su ba.
- Prover ba classical zero-knowledge ba ne — ba shi da simulator.
- Amma kowane falsifiable, game-based security consequence na classical zero-knowledge za a iya achieve shi a wannan setting.

"Falsifiable" yana da muhimmanci. Yana nufin security gazawa za a iya test ta running adversary a game. Cryptographic security definitions da yawa suna da wannan form: adversary zai iya distinguish encryptions biyu, invert function, recover witness, ko win wani specified gwaji? Theorem tana ba da prover ga kowane falsifiable property, one at a time. Single prover da yake enjoying *every* falsifiable property a lokaci guda yana yiwuwa impossible — tsohon reusability attack ("Bob zai iya nuna proof ga wasu") shi kansa falsifiable property ne, kuma genuinely fails a nan. Proposal na takarda shi ne single prover zai iya plausibly cover duk *natural* falsifiable properties — waɗanda suke faruwa a cryptographic practice — amma wannan part conditional theorem ne da yake resting a informal notion na "natural," tare da explicit conjecture. Guarantee ɗin yana target observable gazawa, ba kowane philosophical ko simulation-based meaning na secrecy ba.

Concrete corollary ɗaya ya cancanci a ambata: construction ɗin yana ba da na farko non-interactive *witness hiding* proofs tare da uniform prover — "proof na puzzle ba ya taimaka maka gano solution ɗinsa," ba tare da interaction ko setup ba — object da yana iya yin kama modest amma ya ƙi construction shekaru da dama.

Abin da wannan ba ya cewa

Wannan shi ne section da yake riƙe article ɗin da gaskiya.

Ba ya cewa tsoffin impossibility theorems sun yi kuskure. Construction ɗin yana guje musu ne ta canza definition.

Ba ya ba da ordinary, classical zero-knowledge da no interaction, no setup da perfect soundness. takardar ya faɗi a sarari cewa constructed prover ba shi da simulator.

Ba yana nufin proof ba za a iya reuse shi ba. One-message proof har yanzu za a iya nuna wa wani; takardar ba ya preserve deniability-style properties. (Non-interactive zero-knowledge da trusted setup ma yana da wannan limitation.)

Ba yana nufin wannan practical protocol ne da aka shirya domin deployment ba. Wannan complexity theory da cryptographic foundations ne. Sakamakon yana dogaro da major assumptions daga proof complexity da cryptography, kuma construction ɗin yana magana ne game da abin da zai yiwu in principle.

Ba ya mayar da "Gödel" magic security primitive. Connection da Gödel yana tafiya ne ta proof systems, optimal proof systems da finite analogues na incompleteness. Useful intuition ba "incompleteness yana kare password ɗinka" ba ne. Shi ne: idan rulebook ba zai iya efficiently prove cewa simulator impossible ne ba, attacks da za su buƙaci wannan proof za a iya block su a level na security definitions.

Me ya sa har yanzu yake da ban sha'awa

Cryptography sau da yawa tana juya hardness zuwa safety. Factoring yana da wuya, don haka RSA-style assumptions suna zama useful. Lattice problems suna da wuya, don haka lattice cryptography tana zama useful. A nan hardness ɗin ya fi strange: ba "yana da wuya a compute secret" ba, amma "yana da wuya a prove cewa wani proof object ba zai iya wanzuwa ba."

Wannan shi ya sa takardar yake jin unusual. Yana treating axioms da rulebooks kusan kamar cryptographic resources. Usual impossibility yana cewa akwai tension tsakanin soundness da simulation. Move na Ilango shi ne ya sa tension ɗin a bayan proof-theoretic curtain: simulator baya nan, amma formal system ba zai iya efficiently expose wannan absence ba.

Ga reader, surprising part ba shi ne wannan zai replace today's zero-knowledge systems ba. Da alama ba zai yi haka ba, aƙalla ba kai tsaye ba. Abin mamaki shi ne limitation daga mathematical logic za a iya amfani da shi constructively: ba kawai a matsayin bango ba, amma a matsayin wani irin cover.

Yaya ƙarfin shaidar yake?

Wannan theorem takarda ne, don haka "shaida" yana nufin wani abu daban da biology ko astronomy takarda. Tambayar ba ko gwaji ya replicate ba ne. Tambayar ita ce ko definitions, assumptions da proof chain suna goyon bayan ikirarin.

Proof ɗin formal ne, kuma takardar ya bayyana assumptions ɗinsa a sarari. Assumptions ɗin ba casual ba ne. Non-interactive witness indistinguishable proofs standard objects ne a cryptography kuma suna follow daga several established assumption packages. No-optimal-proof-system conjecture central conjecture ne a proof complexity. P = BPP standard derandomization belief ne da ake amfani da shi kawai ga broader falsifiable-property theorem.

takardar kuma yana argue cewa assumptions ɗin su ne right price, ba arbitrary scaffold ba: yana prove converse da ke nuna essentially necessary ne — idan constructions irin wannan suna wanzuwa kwata-kwata, to non-interactive witness indistinguishable proofs dole su wanzu, kuma (idan aka ba standard one-way functions) optimal proof system ba zai iya wanzuwa ba. Kuma assumptions ɗin "win-win" ne: refuting kowannensu da kansa zai zama landmark discovery a proof complexity, cryptography ko complexity theory.

Amma saboda sakamakon conditional ne, confidence ɗinsa ma conditional ne. Idan waɗannan assumptions sun kasa, interpretation na theorem zai canza. Kuma ko assumptions sun hold, guarantee ɗin ba full classical zero-knowledge ba ne; relaxed, proof-theoretic version na takarda ne.

Saboda haka right confidence shi ne high cewa takardar ya tabbatar coherent conditional possibility sakamako; moderate cewa assumptions ɗinsa suna bayyana cryptographic world da muke rayuwa a ciki; kuma low ga immediate practical consequence.

Me ya sa yake da muhimmanci

takardar ya buɗe route da ake tsammani a rufe yake.

Classical theory tana cewa: full zero-knowledge ba zai iya zama one message ba tare da setup ba, kuma ba zai iya samun perfect soundness ba. takarda na Ilango yana cewa: idan muka tambayi consequences na zero-knowledge da za a iya test a security games, kuma muka bar security definition ya dogara da abin da rulebook zai iya ko ba zai iya efficiently refute ba, to za a iya dawo da yawancin useful behaviour — da one message, no setup da perfect soundness.

Wannan ba ƙaramin definitional tweak ba ne. Wata daban ce hanyar tunani game da cryptographic guarantees. Maimakon tambayar abin da yake wanzuwa kawai, tambayi abin da rulebook ɗinka zai iya rule out. Maimakon treating unprovability a matsayin philosophical nuisance, yi amfani da shi a matsayin structure.

Practical world ba lallai ya canza gobe ba. Amma conceptual map ya canza. Yanzu akwai formal sense inda "babu wanda zai iya efficiently prove cewa secret ya leak" zai iya zama strong enough ya dawo da yawancin game-based protections da muke so daga "secret bai leak ba."

Wannan shi ne dalilin da Gödel yake cikin taken.

Taƙaitaccen bayani mai tsabta

Zero-knowledge proofs suna ba prover damar gamsar da verifier cewa statement gaskiya ne ba tare da bayyana witness ba. Classical impossibility sakamako suna cewa zero-knowledge ba za a iya matsa shi cikin one message

ba tare da setup ba, kuma ba zai iya samun perfect soundness ba. takarda na Rahul Ilango bai refute waɗannan impossibilities ba. Ya define weaker notion, effectively zero-knowledge: maimakon buƙatar simulator ya wanzu da gaske, yana buƙatar chosen proof system — formal rulebook kamar ZFC — ya kasa efficiently prove cewa babu simulator. A ƙarƙashin major assumptions daga cryptography (non-interactive witness indistinguishable proofs) da proof complexity (no optimal proof system exists), takardar yana construct one-message provers ga NP/SAT da no setup da perfect soundness, waɗanda suke achieve falsifiable, game-based consequences na zero-knowledge property by property. Single prover da zai cover duk "natural" irin waɗannan properties further, partly conjectural extension ne — kuma covering literally every falsifiable property da alama impossible ne, saboda proofs suna remain reusable. Sakamakon theoretical ne kuma conditional, ba deployed primitive ba, amma yana nuna sabuwar hanyar amfani da proof-theoretic unprovability a matsayin cryptographic resource.

Binciken No-BS

Abin da takardar ya nuna: A ƙarƙashin stated assumptions, za a iya gina one-message, no-setup, perfectly sound provers ga NP/SAT da suke effectively zero-knowledge relative ga duk chosen proof system, kuma suna achieve kowane falsifiable game-based consequence na classical zero-knowledge.

Abin da yake mai yiwuwa amma ba a prove unconditionally ba: Assumptions na proof complexity da cryptography da ake buƙata suna hold. Serious, well-studied assumptions ne — kuma takardar yana nuna essentially necessary ne kamar yadda sufficient ne — amma har yanzu assumptions ne.

Abin da ba ya nuna: Classical zero-knowledge da no interaction, no setup da perfect soundness; practical system ready for deployment; deniability ko non-reusability na proofs; ko cewa Gödel's incompleteness theorem shi kaɗai yana secure cryptography.

Manyan iyakoki: Guarantee ɗin relaxation ne na zero-knowledge; broadest version yana dogaro da multiple assumptions; single-universal-prover ikirarai suna ci gaba da zama partly conjectural; kuma sakamakon primarily foundational ne.

Yaya yawan amincewa ya dace ga mai karatu na gama gari? Babba cewa wannan muhimmi conditional theory sakamako ne idan definitions an accepted. Matsakaici cewa assumptions suna capture reality. Karami ga immediate practical deployment. Safe takeaway shi ne: takardar bai karya zero-knowledge impossibilities ba; ya sami sabuwar proof-theoretic hanya ta zagaye parts ɗinsu da suke matter ga security games da yawa.

Majiyoyi

Ilango, IACR ePrint 2025/1296
https://eprint.iacr.org/2025/1296

FOCS 2025, DOI 10.1109/FOCS63196.2025.00058
https://doi.org/10.1109/FOCS63196.2025.00058