The Clean Paper

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

Një provë kriptografike mund ta fshehë sekretin duke e bërë të vështirë të provohet mungesa e simulatorit

Provat zero-knowledge lejojnë të provosh një pohim pa zbuluar witness-in. Teoria klasike thotë se kjo, në kuptimin e plotë, nuk mund të kombinohet me një mesazh të vetëm, pa setup të besuar dhe me soundness të përsosur. Punimi i Rahul Ilango-s nuk e zhduk këtë pamundësi. Ai ndryshon objektivin: në vend që të kërkojë që simulatori të ekzistojë vërtet, kërkon që asnjë proof system i zgjedhur të mos mund të provojë në mënyrë efikase se simulatori nuk ekziston. Nën supozime të mëdha nga proof complexity dhe kriptografia, kjo mjafton për të rikuperuar, një veti nga një, pasoja të falsifikueshme dhe game-based të zero-knowledge. Pika nuk është një primitive interneti gati për përdorim; është një mënyrë proof-theoretic për ta kthyer paprovueshmërinë matematike në mbulesë kriptografike.

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

Truku nuk është të provosh se sekreti është i fshehur

Le të nisemi nga versioni më i thjeshtë i zero-knowledge.

Alice dëshiron ta bindë Bob-in se një Sudoku ka zgjidhje. Nëse ia dërgon zgjidhjen, Bob bindet, por puzzle-i prishet. Ajo kërkon diçka më të çuditshme: një provë që zgjidhja ekziston, pa e zbuluar zgjidhjen.

Ky është premtimi i një prove zero-knowledge. Prover-i (Alice) e bind verifier-in (Bob) se një pohim është i vërtetë, duke mos zbuluar asgjë përtej vetë faktit se pohimi është i vërtetë.

Problemi është se ky premtim ka një kosto. Një provë e zakonshme matematike ka dy veçori të rehatshme. Është **një mesazh i vetëm**: e shkruan, e dorëzon dhe largohesh. Dhe ka **soundness të përsosur**: një pohim i rremë nuk ka fare provë të vlefshme. Rezultatet klasike të pamundësisë thonë se zero-knowledge duhet të heqë dorë nga të dyja — dhe jo vetëm nga kombinimi i tyre; secila veçmas është e ndaluar.

Së pari, një provë zero-knowledge ka nevojë për bashkëbisedim. Nëse Alice dërgon vetëm një mesazh, pa ndonjë setup të besuar të rregulluar paraprakisht, garancia zero-knowledge shembet — pavarësisht sa soundness je i gatshëm të sakrifikosh në këmbim.

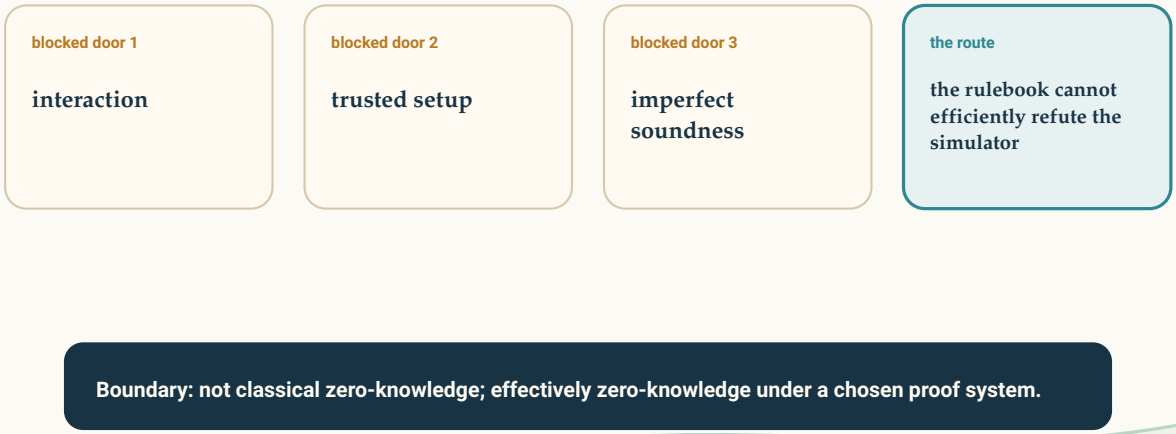
Së dyti, një provë zero-knowledge ka nevojë për një tolerancë të vogël ndaj gabimit. Të kërkosh soundness të përsosur rezulton se e shkatërron në heshtje edhe ndërveprimin: një verifier që nuk mund të mashtrohet kurrë, pavarësisht çfarë zgjedhjesh të rastësishme bën, mund t'i fiksojë ato zgjedhje që më parë — dhe sapo verifier-i bëhet i parashikueshëm, Alice mund t'u përgjigjet të gjithave me një mesazh të vetëm, pikërisht rasti që tashmë dështoi.

Punimi i Rahul Ilango-s është për një rrugë rreth këtij muri të dyfishtë. Jo duke bërë sikur muri nuk ekziston dhe jo duke prodhuar zero-knowledge klasik në një mjedis ku kjo është e pamundur. Lëvizja është më e hollë: dobësohet kuptimi i "nuk zbulon asgjë", por në një mënyrë që ruan vetitë e sigurisë që kriptografët mund t'i testojnë realisht.

Rezultati quhet **effectively zero-knowledge**.

A proof without leaking the witness

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

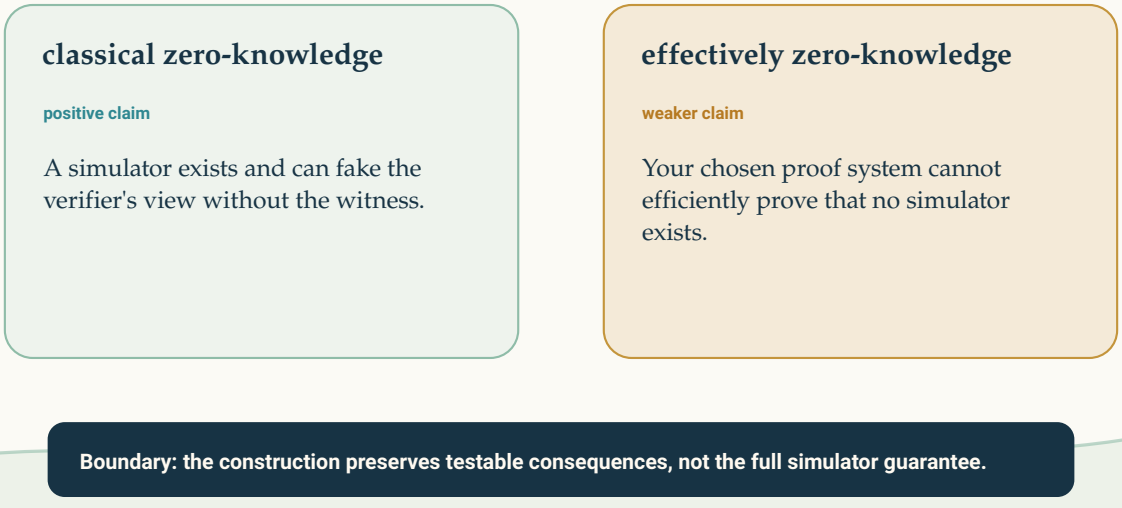


Zero-knowledge bllokohet te tri dyer — ndërveprimi, setup-i i besuar dhe soundness-i jo i përsosur. Ndërtimi i llango-s kalon nga një derë tjetër: rulebook-u nuk mund ta refuzojë në mënyrë efikase simulatorin.

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.



Zero-knowledge klasik pyet nëse një simulator ekziston; “effectively zero-knowledge” pyet vetëm nëse rulebook-u i zgjedhur mund të provojë në mënyrë efikase se simulatori nuk mund të ekzistojë. Pikërisht pyetja

më e dobët lejon një mesazh, zero setup dhe soundness të përsosur.

Testi i vjetër: ekziston një simulator

Mënyra klasike për të formalizuar zero-knowledge përdor një ndihmës imagjinar të quajtur **simulator**.

Ideja është kjo: imagjinoni Jane, e cila **nuk** e di sekretin e Alice-s. Nëse Jane mund të gjenerojë krejt vetë prova që duken tamam si provat që Bob do të merrte nga Alice, atëherë provat e Alice-s nuk i kanë mësuar Bob-it asgjë të re. Jane mund ta falsifikonte të gjithë përvojën pa sekretin e Alice-s.

Pra zero-knowledge klasik kërkon një simulator real. Duhet të ekzistojë një algoritëm efikas që mund të prodhojë prova të rreme, por bindëse, pa e ditur sekretin — *witness*-in, në zhargon; për Sudoku, witness-i është thjesht tabela e zgjidhur.

Ky përkufizim është i fuqishëm, por është edhe pikërisht vendi ku godet pamundësia e vjetër. Intuita është kjo. Një provë vërtet jo-ndërvepruese është thjesht një varg. Pasi Bob e ka atë varg, mund t'ia tregojë dikujt tjetër: ai ka fituar aftësinë t'ua provojë pohimin të tjerëve, gjë që tashmë tingëllon si më shumë se "asgjë". Teoremat klasike e mprehin këtë intuitë në pamundësitë e mësipërme.

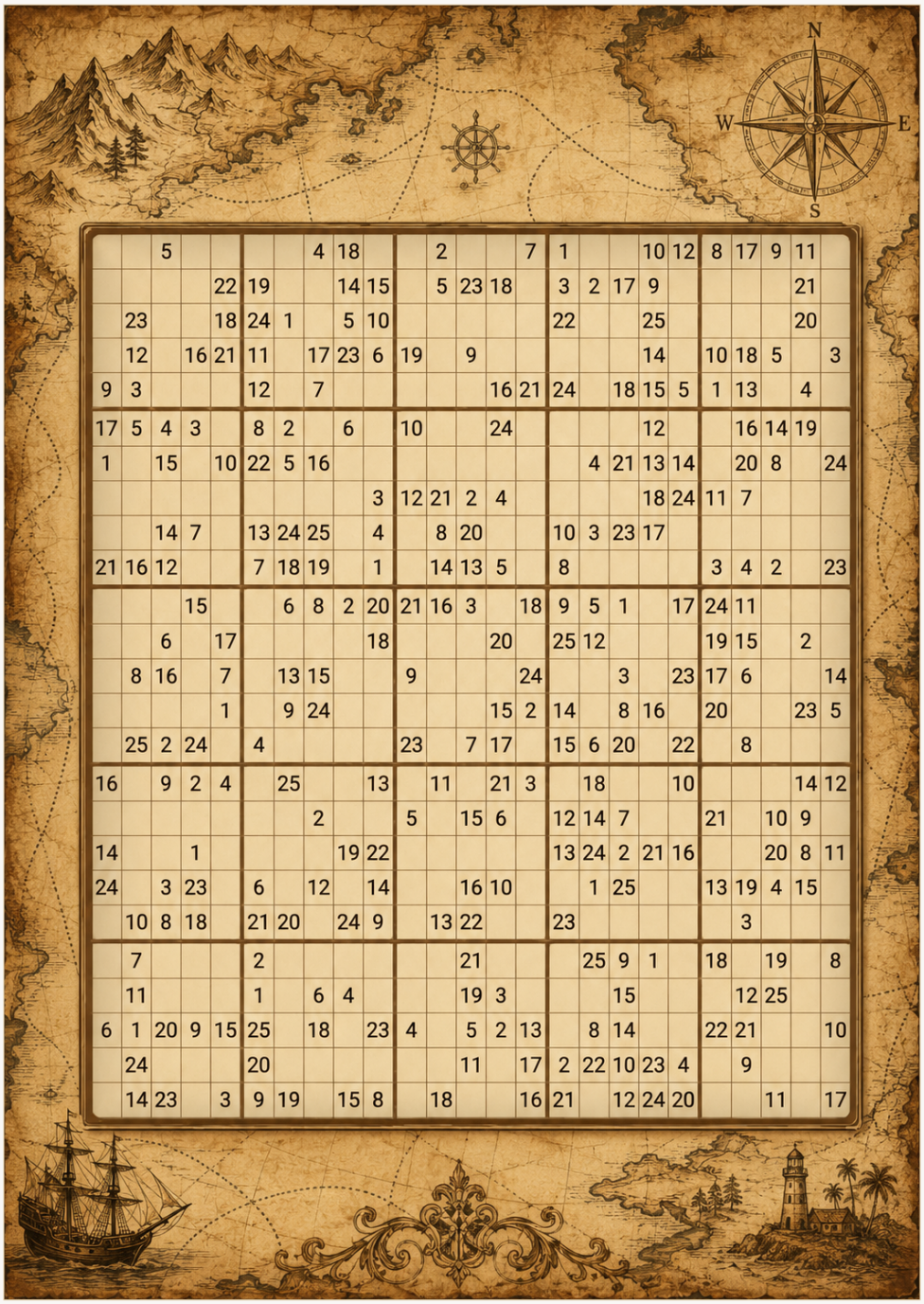
> **Tri vetitë që ky punim këmbëngul t'i mbajë**
>
> Titulli i punimit përmend tri kufizime:
> **Pa ndërveprim:** Alice dërgon një varg të vetëm prove. Nuk ka protokoll vajtje-ardhje.
> **Pa setup:** Alice dhe Bob nuk mbështeten në një common reference string të besuar ose randomness publik të përgatitur paraprakisht. Shumë sisteme të quajtura "non-interactive zero-knowledge" ende mbështeten në setup; ky punim do të thotë zero setup.
> **Soundness i përsosur:** një pohim i rremë nuk ka asnjë provë të vlefshme. Jo "pranohet pothuajse kurrë"; nuk ekziston fare provë e vlefshme.
> Këto tri veti janë pikërisht ato që ka matematika e zakonshme e shkruar — dhe, siç u shpjegua më sipër, zero-knowledge klasik nuk mund t'i mbajë.

Një version MegaSudoku i dallimit

Ja një mënyrë qëllimisht e thjeshtuar për ta ndier dallimin.

Mos përdorni një Sudoku të zakonshëm 9 me 9 për pjesën serioze të analogjisë. Është shumë i vogël dhe shumë i fundëm: një kompjuter mund ta zgjidhë thjesht, ose të provojë se nuk ka zgjidhje. Në vend të tij imagjinoni një familje puzzle-sh **MegaSudoku(n)**. Zmadhojeni rregullin e zakonshëm: zgjidhni një madhësi blloku n, vendosni $N = n^2$, dhe ndërtoni një grid N me N, të ndarë në blloqe n me n, me N simbole. Sudoku i zakonshëm është vetëm rasti i vogël $n = 3$, $N = 9$: grid 9 me 9, blloqe 3

me 3 dhe nëntë simbole. Historia e proof complexity nis vetëm kur n lejohet të rritet dhe kur gridi mund të mbajë gadget-e shtesë që e bëjnë të sillet si një formulë SAT e veshur si Sudoku. Një formulë SAT është thjesht një listë kufizimesh po/jo: a mund t'u caktoni variablave vlera true/false në mënyrë që çdo kufizim të plotësohet?



Një Sudoku 25x25: rregullat e tij mund të kontrollohen pa zbuluar gridin e përfunduar — një metaforë vizuale për një provë që verifikon një zgjidhje të fshehur, witness-in.

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

Sudoku dhe SAT: i njëjti problem me dy kostume

Pretendimi se një Sudoku mund "të sillet si një formulë SAT" nuk është metaforë. Përkthimi funksionon në të dy drejtimet, dhe drejtimi i lehtë mund të shkruhet plotësisht.

Nga Sudoku në SAT. SAT flet vetëm true/false, ndaj jepini nga një variabël boolean çdo tresheje (rresht, kolonë, vlerë): *x(r,c,v)* do të thotë "qeliza në rreshtin *r*, kolonën *c* përmban vlerën *v*." Një Sudoku 4 me 4 (blloqe 2 me 2, vlera 1–4) ka nevojë për 4·4·4 = 64 variabla; ai klasik 9 me 9 ka 729. Çdo rregull Sudoku pastaj bëhet një grup clauses. (Një clause është një OR i variablave ose mohimeve të tyre; e gjithë formula është AND-i i të gjitha clauses.)

Çdo qelizë mban të paktën një vlerë — një clause për qelizë:

$$x(1,1,1) \vee x(1,1,2) \vee x(1,1,3) \vee x(1,1,4)$$

Çdo qelizë mban maksimumi një vlerë — një clause "jo të dyja" për çdo çift vlerash:

$$\neg x(1,1,1) \vee \neg x(1,1,2) \quad \neg x(1,1,1) \vee \neg x(1,1,3)$$ … dhe kështu me radhë për të gjashtë çiftet.

Çdo rresht përmban çdo vlerë — për rreshtin 1 dhe vlerën 3: të paktën një herë,

$$x(1,1,3) \vee x(1,2,3) \vee x(1,3,3) \vee x(1,4,3)$$

dhe maksimumi një herë: $\neg x(1,1,3) \vee \neg x(1,2,3)$, e kështu për çdo çift qelizash në rresht.

Kolonat dhe blloqet — të njëjtat grupe clauses; ndryshon vetëm grupi i qelizave. Për bllokun sipër majtas dhe vlerën 2:

$$x(1,1,2) \vee x(1,2,2) \vee x(2,1,2) \vee x(2,2,2)$$

plus clauses çift "jo të dyja".

Clues e printuara — pjesa më e thjeshtë: çdo clue është një clause me një variabël të vetëm. Një 3 i printuar në këndin sipër majtas bëhet clause

$$x(1,1,3)$$

AND-i i të gjitha këtyre është satisfiable pikërisht kur Sudoku ka zgjidhje — dhe një satisfying assignment *është* zgjidhja: lexoni cilat x(r,c,v) janë true dhe mbushni gridin. Për një 9 me 9 kjo jep 729 variabla dhe disa mijë clauses, që një SAT solver modern i zgjidh në milisekonda. Vini re clause-in e clue-s x(1,1,3): ai thotë "kjo qelizë është saktësisht 3", jo "këto qeliza janë të gjitha të ndryshme" — e njëjta asimetri që do të kërkojë trukun shtesë për qelizat clue në shënimin e protokollit më poshtë.

Nga SAT në Sudoku. Punimit i duhet drejtimi i kundërt, më i vështirë: duke pasur një formulë SAT *arbitrare*, ndërto një mega-Sudoku që ka zgjidhje pikërisht kur formula ka. Rregullat natyrore të Sudoku-t mund të thonë vetëm "këto qeliza janë të gjitha të ndryshme", ndaj kufizime logjike arbitrare duhen *ndërtuar* — dhe pikërisht këtu hyjnë gadget-et. Një gadget është një grup i vogël qelizash i ndërtuar paraprakisht, nga një për çdo clause të formulës, ku qeliza të caktuara luajnë rolin e variablave (simboli që mbajnë kodon true ose false) dhe kufizimet e brendshme të grupit projektohen në mënyrë që mbushjet e vetme legale t'u

korrespondojnë assignments që e plotësojnë atë clause. Kjo është teknikë standarde nga provat e NP-completeness; për Sudoku të përgjithësuar u realizua nga Yato dhe Seta në 2003.
Të dy drejtimet së bashku thonë se Sudoku N-me-N dhe SAT janë i njëjti problem me kostume të ndryshme. Pikërisht kjo i lejon këtij artikulli — dhe punimit — të tregojë një histori për gjithë NP-në duke përdorur gride dhe simbole.

Witness-i mbetet i lehtë për t'u imagjinuar. Alice di një mbushje të plotë dhe të vlefshme të mega-Sudoku-t. Bob dëshiron të bindet se një mbushje e tillë ekziston, por Alice nuk dëshiron ta zbulojë. Nëse dërgon të gjithë gridin, Bob bindet, por sekreti humbet.

Në versionin klasik zero-knowledge, Alice dhe Bob ndërveprojnë. Një model i vjetër mendor përdor pllaka të mbuluara. Alice fsheh gridin e zgjidhur, u ndryshon fshehtas emrat simboleve para çdo raundi dhe i lejon Bob-it të kontrollojë një kufizim lokal të zgjedhur rastësisht: një rresht, kolonë, kuti ose gadget. Nëse qelizat e hapura tregojnë simbole të gjitha të ndryshme, Bob fiton besim. Pastaj gjithçka mbulohet sërish dhe simbolet riemërtohen nga e para. (Ka një hollësi: clues e dhëna të puzzle-it kërkojnë një truk shtesë, sepse riemërtimi i fsheh edhe ato. Shënimi më poshtë shpjegon si e zgjidhin protokollet klasike; për atë që vjen më tej mjafton tabloja e thjeshtuar.)

Si i trajtojnë realisht protokollet klasike qelizat clue

Truku i riemërtimit ka një pikë të verbër. Rregullat e rreshtave, kolonave dhe kutive thonë të gjitha "këto qeliza janë të gjitha të ndryshme", dhe *të gjitha të ndryshme* mbetet e vërtetë nën çdo riemërtim të simboleve. Por një clue thotë "kjo qelizë përmban saktësisht 5", dhe pas riemërtimit Bob sheh vetëm $\sigma(5)$ — një simbol të maskuar — pa e ditur riemërtimin σ. Nuk mund të kontrollojë asgjë. Nëse kjo lihet pa rregulluar, Alice mund të provojë se *një* grid i vlefshëm ekziston duke injoruar plotësisht clues e printuara, gjë që nuk provon asgjë për *këtë* puzzle. Literatura klasike ka dy riparime standarde.
Paleta. Shtoni një rresht shtesë me N qeliza te gridi i fshehur — një paletë që Alice e mbush me simbolet 1…N në një rend publik të fiksuar, dhe pastaj e riemërton bashkë me gjithçka tjetër, kështu që përmban $\sigma(1)…\sigma(N)$. Sfida e rastësishme e Bob-it tani ka edhe një mundësi. Përveç zgjedhjes së një rreshti, kolone, kutie ose gadget-i për t'u hapur, ai mund të zgjedhë **paletën plus një qelizë clue**. Alice i zbulon të dyja; paleta zbulon riemërtimin e atij raundi, dhe Bob kontrollon që qeliza clue tregon saktësisht versionin e riemërtuar të clue-s së printuar. Kjo mbetet zero-knowledge sepse Bob mëson vetëm σ — e cila zgjidhet nga e para në çdo raund dhe vetë nuk vlen asgjë — dhe vlerën e një qelize që tashmë e dinte nga puzzle-i. Asgjë nga qelizat sekrete nuk rrjedh, dhe një simulator mund ta falsifikojë pamjen duke zgjedhur një σ të rastësishme. Është sound sepse një Alice që mashtron kapet me një probabilitet të fiksuar për raund, dhe raundet përsëriten derisa dyshimi bëhet i papërfillshëm.
Kompilimi i clues. Një variant më strukturor heq sfidën speciale në vend që ta shtojë. Në vend që ta *verifikojë* vlerën e clue-s, e detyron me kufizime diference: lidh qelizën clue me çdo qelizë të paletës përveç asaj që mban vlerën e saj — "ndryshe nga $\sigma(1)$, ndryshe nga $\sigma(2)$, …, ndryshe nga gjithçka përveç $\sigma(5)$." I vetmi simbol që qeliza mund të mbajë ligjërisht është clue-ja e saj. Tani çdo kufizim është sërish i llojit "këto dy janë të ndryshme" — invariant ndaj

riemërtimit dhe i kontrollueshëm si një rresht. Ky është i njëjti manovrim që përdoret për vertex-et e para-ngjyrosura në protokollin klasik graph-coloring, dhe është në frymën e fjalës *gadgets* më sipër: në fotografinë MegaSudoku-si-SAT, clues kompilohen në inequality gadgets si çdo kufizim tjetër.
Protokolli fizik. Protokolli real me karta për Sudoku (Gradwohl, Naor, Pinkas dhe Rothblum, 2007) nuk përdor fare riemërtim dhe i zgjidh clues para se të fillojë fshehja. Për çdo qelizë, Alice vendos tri karta identike me vlerën e qelizës — me fytyrë poshtë për qelizat sekrete, por **me fytyrë lart për qelizat clue**, kështu që Bob sheh me sytë e vet se clues respektohen para se kartat të kthehen. Pastaj një kartë nga çdo qelizë futet në paketën e rreshtit, një në paketën e kolonës, një në paketën e kutisë; çdo paketë përzihet dhe zbulohet, dhe Bob kontrollon që përmban të gjitha N simbolet. Përzierja shkatërron informacionin e pozicionit (kjo është pjesa zero-knowledge), por clues ishin fiksuar tashmë gjatë shpërndarjes.
Sido që të bëhet, mësimi është i njëjti që ky artikull vazhdon të përsërisë: një protokoll zero-knowledge është një kontabilitet i kujdesshëm i *cilave* fakte mbijetojnë pas fshehjes. Riemërtimi ruan "të gjitha të ndryshme" dhe fshin "është 5" — prandaj "është 5" duhet të rikthehet me mjete të tjera.

Ky nuk është protokolli i punimit. Është modeli mendor për zero-knowledge klasik:

- Alice dhe Bob komunikojnë vajtje-ardhje.
- Bob zgjedh kontrolle të rastësishme.
- Alice zbulon vetëm konsistencë lokale, jo të gjithë zgjidhjen.
- Prova e privatësisë funksionon duke treguar se pamja e Bob-it mund të ishte gjeneruar pa zgjidhjen sekrete të Alice-s.

Pra zero-knowledge klasik ndërtohet rreth një fakti pozitiv:

> Një simulator ekziston vërtet.

Tani hiqni pjesët e rehatshme. Alice dërgon një varg të vetëm prove dhe largohet. Nuk ka setup të besuar, nuk ka shared random string të përgatitur paraprakisht, dhe Bob nuk duhet të pranojë kurrë një puzzle të rremë. Ky është mjedisi ku zero-knowledge klasik nuk mbijeton.

Para trukut duhet edhe një personazh. Fiksoni një **rulebook**: një sistem formal prove, në kuptimin e logjikës — një grup fiks aksiomash plus rregulla mekanike për kontrollimin e provave matematikore të shkruara. ZFC, aksiomat standarde të matematikës, është shembulli kanonik. Gjithçka nga këtu e tutje thuhet në raport me një rulebook të zgjedhur paraprakisht, dhe zgjedhja është fleksibël: ndërtimi funksionon për cilindo rulebook që fiksoni, përfshirë ZFC.

(Një shënim terminologjie, i marrë nga vetë punimi: "proof system" këtu do të thotë gjithmonë ky rulebook — sistemi formal që kontrollon prova matematike — kurrë mesazhet që dërgon Alice. Makineria e Alice-s dhe Bob-it quhet "prover dhe verifier".)

Versioni në stil Gödel e mban historinë e mega-Sudoku-t, por e ndryshon provën.

Zgjidhni një sistem të dytë kufizimesh me të njëjtën madhësi të shfaqur, quajeni **D**. Për historinë, S dhe D janë dy puzzle MegaSudoku(n) në të njëjtin format. Në prapaskenë, D mund të ketë nisur si formulë logjike e vështirë me madhësi tjetër; nëse duhet, mund të zgjatet me kufizime dummy të padëmshme që të futet në të njëjtin grid. D ndërtohet nga një formulë logjike që është realisht **e paplotësueshme** (*unsatisfiable*): nuk ekziston assignment vlerash që plotëson të gjitha kufizimet, njësoj si një puzzle i prishur që nuk ka grid të plotë të vlefshëm. Një shembull lodër do të ishte formulë që kërkon njëkohësisht "X është true" dhe "X është false". Pra D nuk ka mbushje të vlefshme.

Por D nuk duhet të jetë një puzzle i prishur që është *i lehtë për t'u ekspozuar*. Shembulli lodër më sipër dështon: çdo rulebook e refuzon "X dhe jo-X" me një rresht. D duhet të jetë false në një mënyrë që rulebook-u i zgjedhur nuk mund ta certifikojë me një argument të shkurtër. Nëse rulebook-u mund ta refutonte D me një provë të shkurtër, historia më poshtë do të shembej: rruga alternative që mund të prodhonte prova pa sekretin e Alice-s mund të përjashtohej formalisht, dhe bashkë me të garancia e privatësisë. Prandaj D zgjidhet nga një familje që rulebook-u i fiksuar nuk mund ta refutojë në mënyrë efikase: brenda atij rulebook-u nuk ka provë të shkurtër se D nuk ka zgjidhje.

Prova me një mesazh e Alice-s pastaj flet për një pohim ose/ose:

> ose mega-Sudoku real S ka zgjidhje, ose decoy D ka zgjidhje.

Kjo është lidhja logjike. D **nuk** gjenerohet në ndonjë mënyrë magjike që e bën S të vërtetë. Prova nuk argumenton "D nuk ka zgjidhje, pra S ka zgjidhje." Ajo provon disjunction-in **S ose D**. Soundness-i i përsosur thotë se një disjunction i rremë nuk mund të ketë provë të vlefshme. Meqë D është false në realitet — nuk ka zgjidhje — e vetmja mënyrë që disjunction-i të jetë i vërtetë është që S të jetë i vërtetë. Pra nëse prova pranohet, S duhet të ketë zgjidhje. Decoy nuk mund ta bëjë një S të rremë të bëhet i vërtetë.

Por për pjesën në stil zero-knowledge, pyesni çfarë do të ndodhte nëse D *do të kishte* zgjidhje. Ajo zgjidhje decoy do të vepronte si witness alternativ. Do t'i lejonte dikujt të prodhonte prova pa e ditur zgjidhjen e vërtetë të mega-Sudoku-t të Alice-s — pra një simulator. Në realitet D nuk ka zgjidhje, kështu që kjo rrugë simulimi është e mbyllur. Pika është se rulebook-u nuk mund të provojë në mënyrë efikase se është e mbyllur.

Pra D ka dy punë. Për **soundness**, D është false, kështu që një provë e vlefshme e "S ose D" e detyron S. Për **effective zero-knowledge**, D është e vështirë të refutohet, ndaj rulebook-u nuk mund të përjashtojë shpejt rrugën decoy që do ta bënte simulimin të mundur.

Pra testi i sigurisë nuk është më:

> A mund të provojmë se një simulator ekziston vërtet?

Bëhet:

> A mund rulebook-u juaj të provojë në mënyrë efikase se simulatori është i pamundur?

Nëse përgjigjja është jo, del diçka çuditërisht e fortë: çdo garanci sigurie që (a) mund të vëzhgohet duke ekzekutuar një test dhe (b) provohet — brenda atij rulebook-u — se rrjedh nga ekzistenca e

një simulatori, në fakt vlen. Një sulm i suksesshëm ndaj cilësdo prej tyre do të ishte vetë refutimi i shkurtër që mungon, dhe ai refutim i shkurtër nuk ekziston. Kjo është pjesa “effective” e effectively zero-knowledge.

Pra kontrasti në klasë është:

Zero-knowledge klasik: provat janë të sigurta sepse ekziston një simulator.

Effective zero-knowledge në stil Gödel: provat trajtohen si të sigurta për teste të vëzhgueshme sigurie sepse rulebook-u nuk mund të provojë në mënyrë efikase se simulatori është i pamundur.

Pretendimi i dytë është më i dobët. Dhe pikërisht kjo është arsyeja pse punimi mund të mbajë tri vetitë që thyen versionin klasik: një mesazh, pa setup dhe soundness të përsosur.

Testi i ri: nuk mund të provosh se simulatori mungon

Relaksimi i Ilango-s e ndryshon pyetjen.

Zero-knowledge klasik pyet:

> A ekziston një simulator?

Effectively zero-knowledge pyet diçka më të dobët:

> A mund rulebook-u i zgjedhur të provojë në mënyrë efikase se nuk ekziston simulator?

Kjo mund të tingëllojë si shmangie teknike, por është ideja qendrore. Ndërtimi jeton në një gjendje të çuditshme: simulatori nuk ekziston realisht — punimi e thotë shprehimisht — por rulebook-u që keni fiksuar nuk mund ta provojë në mënyrë efikase këtë. Nëse çdo pasojë e keqe që ju intereson do të kërkonte pikërisht një refutim të tillë, sistemi ende sillet si zero-knowledge për ato pasoja.

Këtu hyn Gödel-i. Jo si zbukurim dhe jo si “Gödel e bën kriptografinë të sigurt”. Lidhja është proof-theoretic. Një rulebook quhet **optimal** nëse është, në një kuptim të saktë, më i miri i mundshëm: sa herë që ndonjë rulebook mund të refutojë një formulë të llojit përkatës me provë të shkurtër, rulebook-u optimal mund ta bëjë gjithashtu, me një provë maksimumi polinomialisht më të gjatë. Krajíček dhe Pudlák konjekturuan në 1989 se **nuk ekziston sistem prove optimal**: cilindo rulebook të fiksoni, një rulebook tjetër provon ndonjë familje pohimesh të vërteta shumë më shkurt. Kjo është një nga konjekturat qendrore të proof complexity dhe një kushëri i fundëm, complexity-theoretic, i teoremës së paplotësisë së Gödel-it: disa pohime të vërteta nuk kanë provë të shkurtër në rulebook-un që keni fiksuar — jo sepse janë të paprovueshme në parim, por sepse çdo rulebook i fiksuar lë disa të vërteta të shkurtra pa prova të shkurtra.

Punimi e supozon këtë konjekturë (në një formë pak më të fortë “infinitely often”, standarde kur konjekturat përdoren kriptografikisht). Përfitimi, nga një teoremë e Krajíček dhe Pudlák, është konkret: për çdo rulebook ekziston një sekuencë formulash që janë vërtet unsatisfiable, të cilat rulebook-u

nuk mund t'i refutojë me prova të shkurtra — dhe, thelbësisht, një algoritëm efikas mund t'i *gjenerojë*. Kjo vetia e fundit, uniformity, e kthen gjithë idenë nga pohim ekzistence në algoritëm real që Alice mund ta ekzekutojë: decoys D dalin nga një linjë prodhimi, jo nga hiçi.

Lëvizja kriptografike është ta vërë këtë mungesë fuqie prove në punë.

Çfarë po bën ndërtimi

Ja forma e ndërtimit të punimit, pa detajet e tepërta.

Fiksoni një rulebook — le të themi ZFC. Nën supozimin e proof complexity, ekziston një sekuencë formulash që mund të gjenerohen në mënyrë efikase, janë në realitet unsatisfiable, por rulebook-u nuk ka provë të shkurtër se janë unsatisfiable.

Tani ndërtoni një provë me një mesazh të kësaj forme:

> ose pohimi real është satisfiable, ose kjo formulë speciale e vështirë është satisfiable.

Formula speciale e vështirë nuk është satisfiable. Prandaj, nëse makineria bazë e provës ka soundness të përsosur, pranimi i mesazhit ende nënkupton se pohimi real është i vërtetë. Kjo jep soundness të përsosur.

Por për sigurinë e llojit zero-knowledge, imagjinoni se formula speciale e vështirë *do të ishte* satisfiable. Atëherë witness-i i saj mund të përdorej për të simuluar prova pa njohur witness-in real. Formula nuk është satisfiable në realitet — por rulebook-u nuk mund ta provojë këtë në mënyrë efikase. Prandaj nuk mund të provojë në mënyrë efikase se simulatori është i pamundur.

Ky është mentesha e gjithë idesë. Sistemi nuk e fsheh sekretin duke prodhuar simulator klasik. E fsheh, për një klasë të madhe testesh sigurie të vëzhgueshme, pas paaftësisë së rulebook-ut për të certifikuar mungesën e simulatorit.

Çfarë pretendon punimi

Teorema kryesore vjen në disa shtresa. Rezultati bazë është ky:

Nën një supozim standard kriptografik — ekzistencën e **non-interactive witness indistinguishable proofs**, objekte të studiuara mirë që rrjedhin nga disa paketa të njohura supozimesh — dhe nën konjekturën e proof complexity se **nuk ekziston një (infinitely often) optimal proof system**, punimi ndërton, për çdo rulebook të zgjedhur, një prover dhe verifier me një mesazh për NP/SAT, me **soundness të përsosur** dhe pa setup, që është effectively zero-knowledge relativisht ndaj atij rulebook-u. (NP/SAT është "emëruesi i përbashkët më i vështirë" standard i problemeve të tipit puzzle; mega-Sudoku është një nga kostumet e tij.)

Për pretendimin më të gjerë rreth ruajtjes së vetive të sigurisë të falsifikueshme, punimi shton edhe një supozim standard, bindjen e derandomization **P = BPP** (afërsisht: randomness nuk u jep algoritmeve fuqi shtesë thelbësore).

Përkthyer jashtë gjuhës së teoremave:

- Prova është një mesazh i vetëm.
- Nuk ka setup të besuar.
- Pohimet e rreme nuk mund të provohen.
- Prover-i nuk është zero-knowledge klasik — nuk ka simulator.
- Por çdo pasojë e falsifikueshme, game-based, e zero-knowledge klasik mund të arrihet në këtë mjedis.

"Falsifikueshme" ka rëndësi. Do të thotë se një dështim sigurie mund të testohet duke ekzekutuar një adversary në një lojë. Shumë përkufizime kriptografike të sigurisë kanë këtë formë: a mund adversary të dallojë dy ciphertexts, të invertë një funksion, të rikuperojë witness-in ose të fitojë një eksperiment të specifikuar? Teorema jep një prover për çdo veti të falsifikueshme, një nga një. Një prover i vetëm që gëzon *çdo* veti të falsifikueshme njëkohësisht ka gjasa të jetë i pamundur — sulmi i vjetër i reusability ("Bob mund t'ua tregojë provën të tjerëve") është vetë veti e falsifikueshme dhe këtu dështon realisht. Propozimi i punimit është se një prover i vetëm mund, me gjasë, të mbulojë të gjitha vetitë e falsifikueshme *natyrale* — ato që shfaqen vërtet në praktikën kriptografike — por kjo pjesë është teoremë e kushtëzuar mbi një nocion joformal të "natyrales", plus një konjekturë të shprehur. Garancia synon dështime të vëzhgueshme, jo çdo kuptim filozofik ose simulation-based të sekretit.

Një korolar konkret ia vlen të përmendet: ndërtimi jep provat e para non-interactive *witness hiding* me prover uniform — "një provë e një puzzle-i nuk të ndihmon të gjesh zgjidhjen e tij", pa ndërveprim dhe pa setup — një objekt që tingëllon modest, por i kishte rezistuar ndërtimit për dekada.

Çfarë kjo nuk thotë

Kjo është pjesa që e mban artikullin të ndershëm.

Nuk thotë se teoremat e vjetra të pamundësisë ishin gabim. Ndërtimi i shmang duke ndryshuar përkufizimin.

Nuk jep zero-knowledge të zakonshëm klasik pa ndërveprim, pa setup dhe me soundness të përsosur. Punimi thotë shprehimisht se prover-i i ndërtuar nuk ka simulator.

Nuk do të thotë se prova nuk mund të ripërdoret. Një provë me një mesazh mund t'i tregohet përsëri dikujt tjetër; punimi nuk ruan vetitë e stilit deniability. (Edhe non-interactive zero-knowledge me setup të besuar ka të njëjtin kufizim.)

Nuk do të thotë se ky është protokoll praktik gati për deployment. Kjo është complexity theory dhe themel kriptografik. Rezultati varet nga supozime të mëdha nga proof complexity dhe kriptografia,

dhe ndërtimi flet për atë që është e mundur në parim.

Nuk e bën "Gödel" një primitive magjike sigurie. Lidhja me Gödel-in kalon nga proof systems, optimal proof systems dhe analogët e fundëm të paplotësisë. Intuita e dobishme nuk është "paplotësia mbron fjalëkalimin tënd". Është: nëse një rulebook nuk mund të provojë në mënyrë efikase se simulatori është i pamundur, atëherë sulmet që do të kërkonin atë provë mund të bllokohen në nivelin e përkufizimeve të sigurisë.

Pse është interesante gjithsesi

Kriptografia shpesh e kthen vështirësinë në siguri. Faktorizimi është i vështirë, ndaj supozimet e stilit RSA bëhen të dobishme. Problemet lattice janë të vështira, ndaj lattice cryptography bëhet e dobishme. Këtu vështirësia është më e çuditshme: jo "është e vështirë të llogaritësh një sekret", por "është e vështirë të provosh se një objekt prove i caktuar nuk mund të ekzistojë".

Pikërisht kjo e bën punimin të pazakontë. Ai i trajton aksiomat dhe rulebooks pothuajse si burime kriptografike. Pamundësia e zakonshme thotë se ka tension mes soundness dhe simulation. Lëvizja e Ilango-s është ta vendosë këtë tension pas një perdeje proof-theoretic: simulatori mungon, por sistemi formal nuk mund ta ekspozojë në mënyrë efikase këtë mungesë.

Për lexuesin, pjesa befasuese nuk është se kjo do të zëvendësojë sistemet e sotme zero-knowledge. Me gjasë nuk do ta bëjë, të paktën jo drejtpërdrejt. Pjesa befasuese është se një kufizim nga logjika matematike mund të përdoret në mënyrë konstruktive: jo vetëm si mur, por si një lloj mbulese.

Sa e fortë është prova?

Ky është punim teoremash, ndaj "prova" këtu do të thotë diçka tjetër nga një punim biologjie ose astronomie. Pyetja nuk është nëse një eksperiment u replikua. Pyetja është nëse përkufizimet, supozimet dhe zinxhiri i provës e mbështesin pretendimin.

Prova është formale dhe punimi i deklaron qartë supozimet. Ato nuk janë të rastësishme. Non-interactive witness indistinguishable proofs janë objekte standarde në kriptografi dhe rrjedhin nga disa paketa supozimesh të njohura. Konjektura no-optimal-proof-system është një konjekturë qendrore e proof complexity. P = BPP është një bindje standarde e derandomization, e përdorur vetëm për teoremën më të gjerë mbi vetitë e falsifikueshme.

Punimi argumenton gjithashtu se këto supozime janë çmimi i duhur, jo një skelë arbitrare: provon një converse që tregon se janë në thelb të nevojshme — nëse ndërtime të tilla ekzistojnë fare, atëherë non-interactive witness indistinguishable proofs duhet të ekzistojnë dhe (duke pranuar standard one-way functions) nuk mund të ekzistojë optimal proof system. Dhe supozimet janë "win-win": rrëzimi i cilitdo prej tyre do të ishte vetë zbulim i madh në proof complexity, kriptografi ose complexity theory.

Por sepse rezultati është i kushtëzuar, edhe besimi është i kushtëzuar. Nëse ato supozime dështojnë,

interpretimi i teoremës ndryshon. Dhe edhe nëse mbahen, garancia nuk është zero-knowledge klasik i plotë; është versioni i relaksuar, proof-theoretic, i punimit.

Prandaj niveli i duhur i besimit është i lartë se punimi vendos një rezultat koherent të kushtëzuar për mundësinë; i moderuar se supozimet përshkruajnë botën kriptografike ku jetojmë realisht; dhe i ulët për pasoja praktike të menjëhershme.

Pse ka rëndësi

Punimi hap një rrugë që supozohej e mbyllur.

Teoria klasike thotë: zero-knowledge i plotë nuk mund të jetë një mesazh pa setup dhe nuk mund të ketë soundness të përsosur. Punimi i Ilango-s thotë: nëse kërkojmë pasojat e zero-knowledge që mund të testohen në security games dhe lejojmë që përkufizimi i sigurisë të varet nga ajo që një rulebook mund ose nuk mund ta refutojë në mënyrë efikase, shumë nga sjellja e dobishme mund të rikuperohet — me një mesazh, pa setup dhe me soundness të përsosur.

Ky nuk është ndryshim i vogël terminologjik. Është mënyrë tjetër për të menduar garancitë kriptografike. Në vend që të pyesësh vetëm çfarë ekziston, pyet çfarë mund të përjashtojë rulebook-u yt. Në vend që unprovability të trajtohet si bezdi filozofike, përdore si strukturë.

Bota praktike mund të mos ndryshojë nesër. Por harta konceptuale ndryshon. Tani ka një kuptim formal në të cilin "askush nuk mund të provojë në mënyrë efikase se sekreti rrjedhi" mund të jetë mjaftueshëm e fortë për të rikuperuar shumë nga mbrojtjet game-based që kërkonim nga "sekreti nuk rrjedhi".

Prandaj Gödel-i është në titull.

Përmbledhje e shkurtër

Provat zero-knowledge i lejojnë një prover-i ta bindë verifier-in se një pohim është i vërtetë pa zbuluar witness-in. Rezultatet klasike të pamundësisë thonë se zero-knowledge nuk mund të ngjeshet në një mesazh pa setup dhe nuk mund të ketë soundness të përsosur. Punimi i Rahul Ilango-s nuk i rrëzon këto pamundësi. Ai përcakton një nocion më të dobët, effectively zero-knowledge: në vend që të kërkojë që simulatori të ekzistojë vërtet, kërkon që një proof system i zgjedhur — një rulebook formal si ZFC — të mos mund të provojë në mënyrë efikase se simulatori nuk ekziston. Nën supozime të mëdha nga kriptografia (non-interactive witness indistinguishable proofs) dhe proof complexity (nuk ekziston optimal proof system), punimi ndërton prover-a me një mesazh për NP/SAT, pa setup dhe me soundness të përsosur, që arrijnë pasojat e falsifikueshme, game-based, të zero-knowledge një veti nga një. Një prover i vetëm që mbulon të gjitha vetitë e tilla "natyrale" është një zgjerim tjetër, pjesërisht konjektural — dhe mbulimi i çdo vetie të falsifikueshme ka gjasa të jetë i pamundur, sepse provat mbeten të ripërdorshme. Rezultati është teorik dhe i kushtëzuar, jo primitive gati për përdorim, por tregon një mënyrë të re për ta përdorur paprovueshmërinë proof-theoretic si burim

kriptografik.

Kontroll pa teprime

Çfarë tregon punimi: Nën supozimet e deklaruara, mund të ndërtohen prover-a me një mesazh, pa setup dhe me soundness të përsosur për NP/SAT, që janë effectively zero-knowledge relativisht ndaj cilitdo proof system të zgjedhur dhe që arrijnë secilën pasojë të falsifikueshme, game-based, të zero-knowledge klasik.

Çfarë është e besueshme, por jo e provuar pa kushte: Se vlejnë supozimet e nevojshme nga proof complexity dhe kriptografia. Janë supozime serioze dhe të studiuara mirë — dhe punimi tregon se janë në thelb si të nevojshme, ashtu edhe të mjaftueshme — por mbeten supozime.

Çfarë nuk tregon: Zero-knowledge klasik pa ndërveprim, pa setup dhe me soundness të përsosur; një sistem praktik gati për deployment; deniability ose mos-ripërdorim të provave; ose se teorema e paplotësisë së Gödel-it, vetë, e siguron kriptografinë.

Kufizimet kryesore: Garancia është relaksim i zero-knowledge; versioni më i gjerë varet nga disa supozime; pretendimet për një prover të vetëm universal mbeten pjesërisht konjekturale; dhe rezultati është kryesisht themelor.

Sa besim duhet të ketë një lexues i përgjithshëm? Të lartë se ky është rezultat teorik i kushtëzuar me rëndësi, nëse pranojmë përkufizimet. Të moderuar se supozimet përshkruajnë realitetin. Të ulët për deployment të menjëhershëm. Përfundimi i sigurt është: punimi nuk i thyen pamundësitë e zero-knowledge; gjen një rrugë të re proof-theoretic rreth pjesëve të tyre që kanë rëndësi për shumë security games.

Burimet

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`