The Clean Paper

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

O demonstrație criptografică își poate ascunde secretul făcând simulatorul lipsă greu de dovedit

Demonstrațiile zero-knowledge permit cuiva să dovedească un enunț fără să dezvăluie martorul. Teoria clasică spune că nu poți avea asta, în sensul deplin, cu un singur mesaj, fără setup de încredere și cu soliditate perfectă. Paperul lui Rahul Ilango nu face imposibilitatea să dispară. Schimbă ținta: în loc să ceară ca un simulator să existe cu adevărat, cere ca niciun sistem de demonstrație ales să nu poată dovedi eficient că simulatorul nu există. Sub ipoteze majore din complexitatea demonstrațiilor și criptografie, asta ajunge pentru a recupera consecințele falsificabile, bazate pe jocuri, ale proprietății zero-knowledge, proprietate cu proprietate. Ideea nu este o primitivă internet plug-in. Este o cale proof-theoretic de a transforma nedemonstrabilitatea matematică în acoperire criptografică.

Written by Cinzia Vaglio · Cross-checked by Vera Rubin · 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

Trucul nu este să dovedești că secretul este ascuns

Să pornim de la cea mai simplă versiune a proprietății zero-knowledge.

Alice vrea să îl convingă pe Bob că un Sudoku are o soluție. Dacă îi trimite soluția, Bob este convins, dar puzzle-ul este stricat. Ce vrea ea este mai ciudat: o demonstrație că există o soluție, fără să dezvăluie soluția.

Aceasta este promisiunea unei demonstrații zero-knowledge. Proverul, Alice, convinge verificatorul, Bob, că un enunț este adevărat fără să dezvăluie nimic dincolo de adevărul enunțului.

Problema este că această promisiune costă ceva. O demonstrație matematică obișnuită are două trăsături comode. Este **un singur mesaj**: o scrii, o predai și pleci. Și este **perfect solidă**: un enunț fals nu are nicio demonstrație validă. Rezultatele clasice de imposibilitate spun că zero-knowledge trebuie să renunțe la ambele trăsături, și nu doar la combinația lor: fiecare este interzisă și separat.

Mai întâi, o demonstrație zero-knowledge are nevoie de conversație. Dacă Alice trimite un singur mesaj, fără un setup de încredere pregătit dinainte, garanția zero-knowledge se prăbușește, oricâtă soliditate ai fi dispus să sacrifici în schimb.

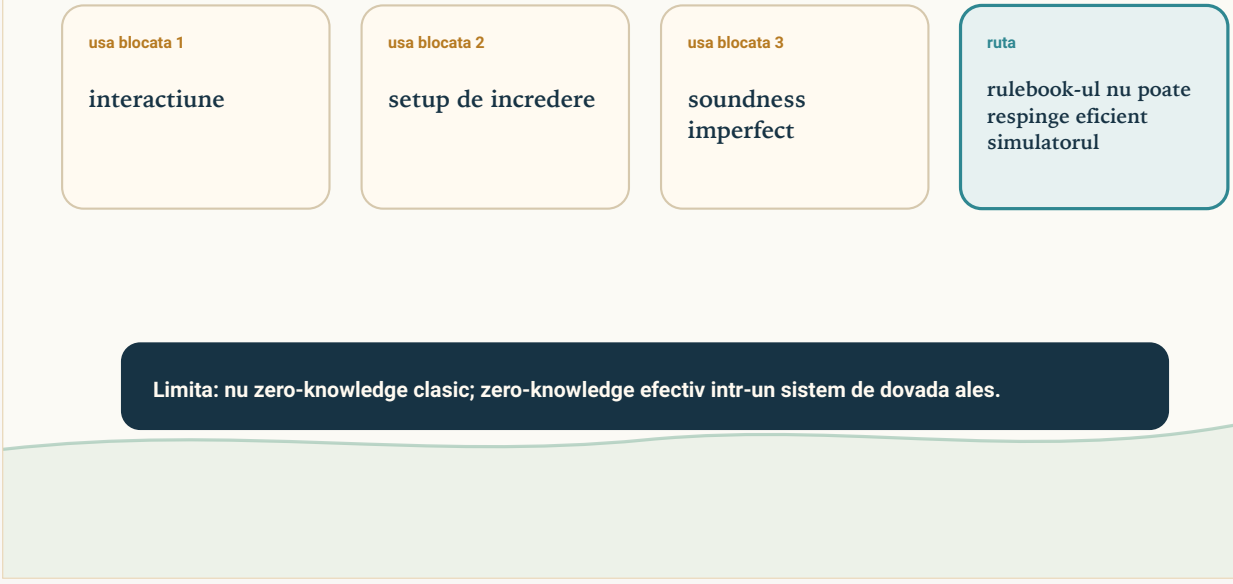
În al doilea rând, o demonstrație zero-knowledge are nevoie de o mică toleranță la eroare. Cerința de soliditate perfectă ajunge să distrugă discret și interacțiunea: un verificator care nu poate fi păcălit niciodată, indiferent ce alegeri aleatorii face, ar putea la fel de bine să fixeze acele alegeri dinainte; iar când verificatorul este previzibil, Alice poate răspunde la tot printr-un singur mesaj, adică exact cazul care s-a rupt deja.

Paperul lui Rahul Ilango este despre o cale de a ocoli acel zid dublu. Nu pretinzând că zidul nu există și nu producând zero-knowledge clasic în cadrul imposibil. Mișcarea este mai subtilă: slăbește ce înseamnă „nu dezvăluie nimic”, dar o slăbește într-un mod care păstrează proprietățile de securitate pe care criptografii le pot testa cu adevărat.

Rezultatul se numește **effectively zero-knowledge**.

O dovada fara a dezvalui martorul

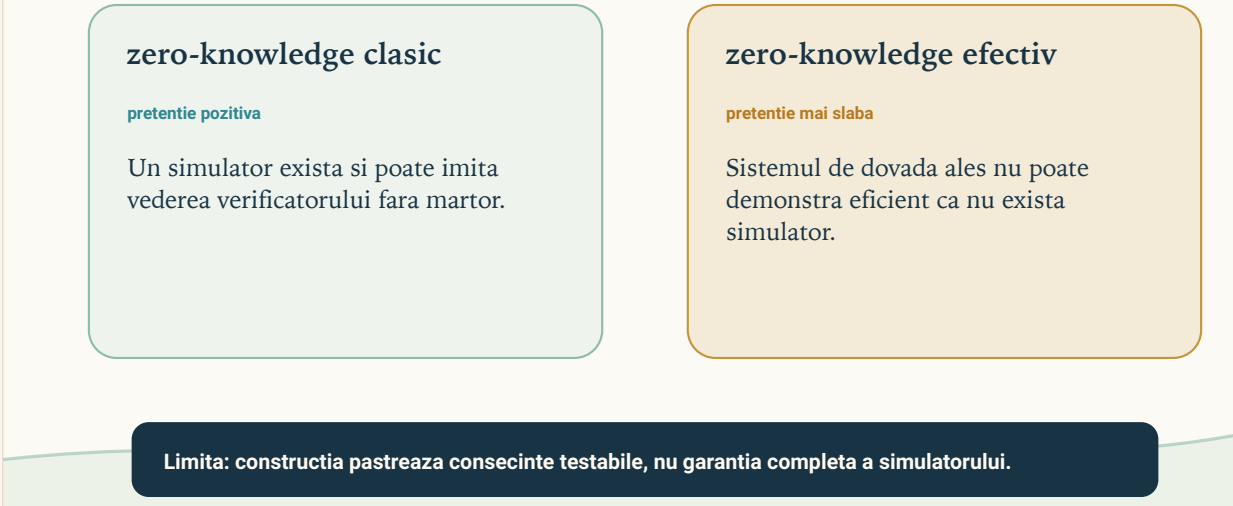
Articolul pastreaza forma unei dovezi obisnuite slabind ce trebuie sa demonstreze privacy-ul.



Zero-knowledge este blocat la trei uși: interacțiune, setup de încredere și soliditate imperfectă. Construcția lui Ilango trece printr-o altă ușă: rulebookul nu poate respinge eficient simulatorul.

Privacy clasica vs privacy efectiva

Relaxarea este punctul: articolul schimba ce trebuie sa stabileasca pretentia de privacy.



Zero-knowledge clasic întreabă dacă există un simulator; „effectively zero-knowledge” întreabă doar dacă rulebookul ales poate dovedi eficient că nu există unul. Întrebarea mai slabă este ceea ce permite construcției

să păstreze un singur mesaj, niciun setup și soliditate perfectă.

Vechiul test: există un simulator

Modul clasic de a formaliza zero-knowledge folosește un ajutor fictiv numit **simulator**.

Ideea este aceasta: imaginează-ți-o pe Jane, care **nu** cunoaște secretul lui Alice. Dacă Jane poate genera, complet singură, demonstrații care arată exact ca demonstrațiile pe care Bob le-ar fi primit de la Alice, atunci demonstrațiile lui Alice nu l-au învățat pe Bob nimic nou. Jane putea deja să falsifice experiența fără secretul lui Alice.

Așadar, zero-knowledge clasic cere un simulator real. Trebuie să existe un algoritm eficient care poate produce demonstrații cu aspect fals fără să cunoască secretul, *witnessul* în jargon; pentru Sudoku, witnessul este pur și simplu grila rezolvată.

Această definiție este puternică, dar este și exact locul unde mușcă vechea imposibilitate. Intuiția este următoarea. O demonstrație cu adevărat neinteractivă este doar un șir. Odată ce Bob are acel șir, îl poate arăta altcuiva: a câștigat capacitatea de a dovedi enunțul altora, ceea ce sună deja ca mai mult decât „nimic”. Teoremele clasice transformă acea intuiție în imposibilitățile de mai sus.

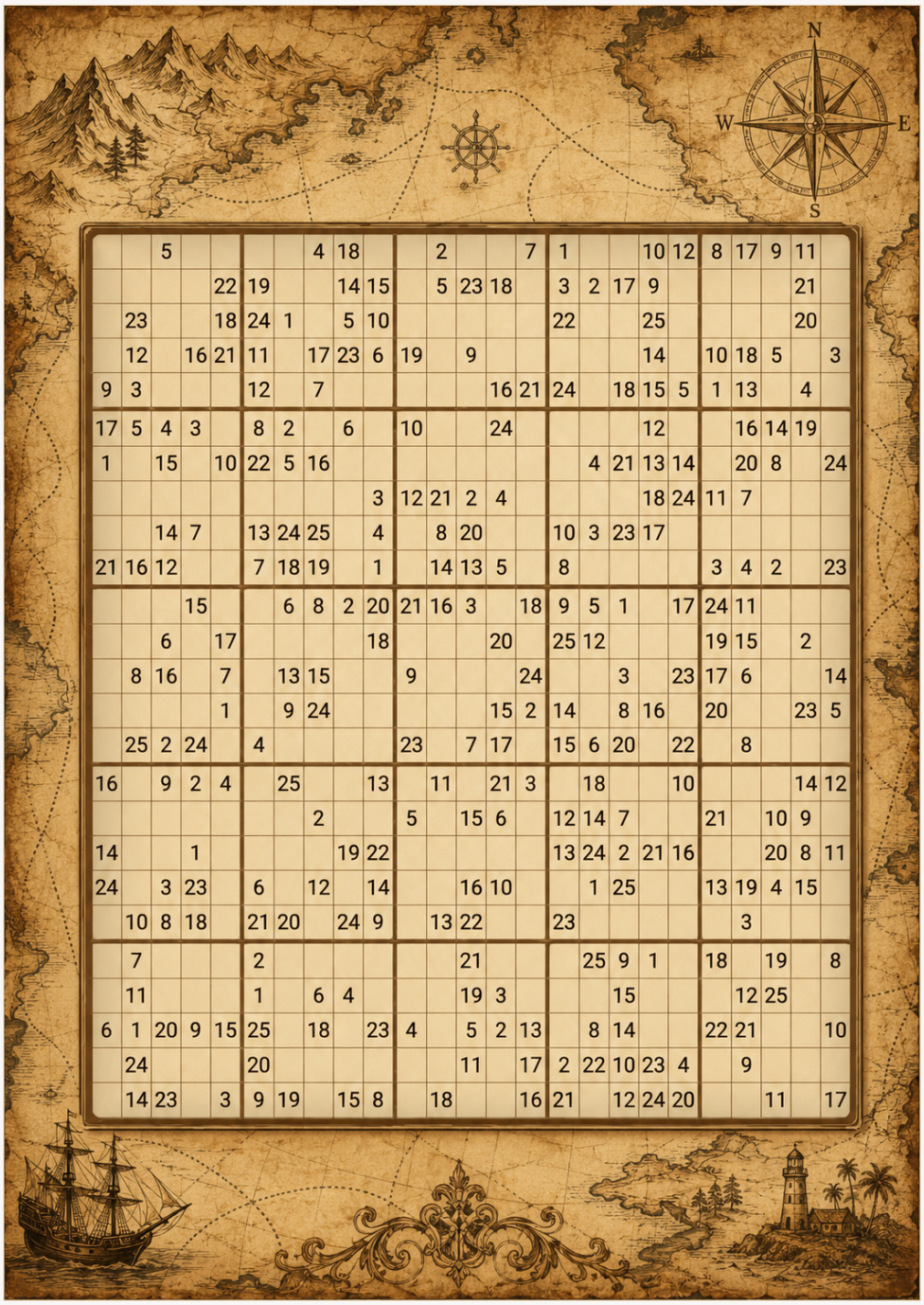
> **Cele trei proprietăți pe care insistă acest paper**
>
> Titlul paperului numește trei constrângeri:
> **Fără interacțiune:** Alice trimite un șir de demonstrație. Nu există protocol dus-întors.
> **Fără setup:** Alice și Bob nu se bazează pe un șir comun de referință de încredere sau pe altă aleatorie publică aranjată dinainte. Multe sisteme numite „zero-knowledge neinteractiv” se bazează totuși pe setup; acest paper înseamnă setup zero.
> **Soliditate perfectă:** un enunț fals nu are demonstrație validă. Nu „aproape niciodată acceptat”; nu există nicio demonstrație validă.
> Aceste trei proprietăți sunt exact ceea ce are matematica scrisă obișnuită și, cum s-a explicat mai sus, zero-knowledge clasic nu le poate păstra.

O versiune mega-Sudoku a diferenței

Iată un mod deliberat simplificat de a simți diferența.

Nu folosi un Sudoku obișnuit 9 pe 9 pentru partea serioasă a analogiei. Este prea mic și prea finit: un computer îl poate pur și simplu rezolva sau poate dovedi că nu are soluție. Imaginează-ți în schimb o familie de puzzle-uri **MegaSudoku(n)**. Scalează regula obișnuită: alege o dimensiune de bloc n, lasă $N = n^2$ și construiește o grilă N pe N împărțită în blocuri n pe n, cu N simboluri. Sudoku obișnuit este doar cazul minuscul $n = 3$, $N = 9$: o grilă 9 pe 9, blocuri 3 pe 3 și nouă simboluri. Povestea complexității

demonstrațiilor începe abia când n poate crește și când grila poate purta gadgeturi suplimentare care o fac să se comporte ca o formulă SAT îmbrăcată în Sudoku. O formulă SAT este doar o listă de constrângeri da/nu: poți atribui valori adevărat/fals variabilelor astfel încât fiecare constrângere să fie satisfăcută?



Un Sudoku 25x25: regulile sale pot fi verificate fără să dezvăluie grila finală, un substitut vizual pentru o demonstrație care verifică o soluție ascunsă, witnessul.

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

Sudoku și SAT: același puzzle în două costume

Afirmația că un Sudoku poate „să se comporte ca o formulă SAT” nu este o metaforă. Traducerea funcționează în ambele direcții, iar direcția ușoară poate fi scrisă complet.

De la Sudoku la SAT. SAT vorbește doar adevărat/fals, așa că dă-i câte o variabilă booleană pentru fiecare triplet (rând, coloană, valoare): *x(r,c,v)* înseamnă „celula din rândul *r*, coloana *c* conține valoarea *v*.” Un Sudoku 4 pe 4 (blocuri 2 pe 2, valori 1-4) are nevoie de 4·4·4 = 64 de variabile; clasicul 9 pe 9 are nevoie de 729. Fiecare regulă Sudoku devine apoi un lot de clauze. O clauză este un OR de variabile sau negațiile lor; întreaga formulă este AND-ul tuturor clauzelor.

Fiecare celulă ține cel puțin o valoare, fiecare celulă ține cel mult o valoare, fiecare rând conține fiecare valoare, iar același lucru se aplică pentru coloane și blocuri. Indiciile tipărite sunt partea cea mai simplă: un 3 tipărit în colțul din stânga sus devine clauza cu o singură variabilă *x(1,1,3)*. AND-ul tuturor acestor lucruri este satisfiabil exact când Sudoku are o soluție, iar o atribuire satisfăcătoare *este* soluția: citești care *x(r,c,v)* sunt adevărate și completezi grila. Pentru un 9 pe 9 se ajunge la 729 de variabile și câteva mii de clauze, pe care un SAT solver modern le rezolvă în milisecunde.

De la SAT la Sudoku. Paperul are nevoie de direcția opusă, mai dificilă: dată o formulă SAT *arbitrară*, să construiască un mega-Sudoku care are o soluție exact când formula are una. Regulile native ale Sudoku pot spune doar „aceste celule sunt toate diferite”, deci constrângerile logice arbitrare trebuie *construite*, iar acesta este exact rolul gadgeturilor. Un gadget este un mic cluster prefabricat de celule, unul pentru fiecare clauză a formulei, în care celule desemnate joacă rolul variabilelor, iar constrângerile interne sunt proiectate astfel încât singurele completări legale să corespundă atribuirilor care satisfac clauza. Este meșteșug standard din demonstrațiile de NP-completitudine; pentru Sudoku generalizat a fost realizat de Yato și Seta în 2003.

Împreună, cele două direcții spun că Sudoku N pe N și SAT sunt aceeași problemă în două costume diferite. Acesta este lucrul care autorizează articolul de față, și paperul, să spună o poveste despre toată NP folosind grile și simboluri.

Witnessul rămâne ușor de imaginat. Alice cunoaște o completare validă a mega-Sudoku-ului. Bob vrea să fie convins că există o astfel de completare, dar Alice nu vrea să o dezvăluie. Dacă trimite întreaga completare, Bob este convins, dar secretul s-a pierdut.

În versiunea clasică zero-knowledge, Alice și Bob interacționează. Un model mental vechi folosește plăcuțe acoperite. Alice ascunde grila rezolvată, redenumește în secret simbolurile înaintea fiecărei runde și îl lasă pe Bob să inspecteze o constrângere locală aleasă aleatoriu: un rând, o coloană, un box sau un gadget. Dacă celulele deschise arată simboluri toate diferite, Bob câștigă încredere. Apoi totul este acoperit din nou și simbolurile sunt redenumite proaspăt.

Cum gestionează de fapt protocoalele clasice celulele-indiciu

Trucul redenumirii are un punct orb. Regulile de rând, coloană și box spun toate „aceste celule sunt toate diferite”, iar *toate diferite* supraviețuiește oricărei redenumiri a simbolurilor. Dar

> un indiciu spune „această celulă conține exact 5”; după redenumire, Bob vede doar $\sigma(5)$, un simbol mascat, fără să cunoască σ. Nu poate verifica nimic. Lăsată așa, Alice ar putea dovedi că există *o anumită* grilă validă ignorând indiciile tipărite, ceea ce nu dovedește nimic despre *acest* puzzle. Literatura clasică are două reparații standard.
> **Paleta.** Adaugă un rând suplimentar de N celule la grila ascunsă: o paletă pe care Alice o umple cu simbolurile 1…N într-o ordine publică fixă, apoi o redenumește împreună cu restul, astfel încât conține $\sigma(1)$…$\sigma(N)$. Provocarea aleatorie a lui Bob are acum o opțiune în plus. Pe lângă alegerea unui rând, a unei coloane, a unui box sau a unui gadget de deschis, el poate alege **paleta plus o celulă-indiciu**. Alice le descoperă pe amândouă; paleta dezvăluie redenumirea acelei runde, iar Bob verifică dacă celula-indiciu arată exact versiunea redenumită a indiciului tipărit. Aceasta rămâne zero-knowledge pentru că Bob află doar σ, extrasă proaspăt la fiecare rundă și inutilă singură, plus valoarea unei celule pe care o știa deja din puzzle.
> **Compilarea indiciilor.** O variantă mai structurală elimină provocarea specială în loc să o adauge. În loc să *verifice* valoarea indiciului, o forțează cu constrângeri de diferență: leagă celula-indiciu de fiecare celulă a paletei cu excepția celei care poartă propria valoare — „diferită de $\sigma(1)$, diferită de $\sigma(2)$, …, diferită de tot în afară de $\sigma(5)$.” Singurul simbol pe care celula îl poate conține legal este cel al indiciului. Fiecare constrângere este din nou de tipul „aceste două diferă”, invariantă sub redenumire și verificabilă ca un rând.
> **Protocolul fizic.** Protocolul real cu cărți pentru Sudoku (Gradwohl, Naor, Pinkas și Rothblum, 2007) nu folosește redenumire și fixează indiciile înainte să înceapă ascunderea. Pentru fiecare celulă, Alice pune trei cărți identice cu valoarea celulei: cu fața în jos pentru celulele secrete, dar **cu fața în sus pentru celulele-indiciu**, astfel încât Bob vede cu ochii lui că indiciile sunt respectate înainte ca toate cărțile să fie întoarse. Apoi o carte din fiecare celulă merge în pachetul rândului, una în cel al coloanei, una în cel al boxului; fiecare pachet este amestecat și dezvăluit, iar Bob verifică dacă include toate cele N simboluri. Amestecarea distruge informația de poziție, adică zero-knowledge, dar indiciile fuseseră deja fixate la împărțire.

Acesta nu este protocolul din paper. Este modelul mental pentru zero-knowledge clasic:

- Alice și Bob merg înainte și înapoi.
- Bob alege verificări aleatorii.
- Alice dezvăluie doar consistență locală, nu întreaga soluție.
- Demonstrația de confidențialitate funcționează arătând că vederea lui Bob ar fi putut fi generată fără soluția secretă a lui Alice.

Zero-knowledge clasic este deci construit în jurul unui fapt pozitiv:

> Un simulator chiar există.

Acum elimină părțile comode. Alice trimite un șir de demonstrație și pleacă. Nu există setup de încredere, niciun șir aleatoriu comun pregătit dinainte, iar Bob nu trebuie să accepte niciodată un puzzle fals. Acesta este cadrul în care zero-knowledge clasic nu poate supraviețui.

Mai este nevoie de un personaj. Fixează un **rulebook**: un sistem formal de demonstrație, în sens logic,

adică un set fix de axiome plus reguli mecanice pentru verificarea demonstrațiilor matematice scrise. ZFC, axiomele standard ale matematicii, este exemplul canonic. De aici înainte totul este relativ la un rulebook ales dinainte, iar alegerea este flexibilă: construcția funcționează pentru orice rulebook fixat, inclusiv ZFC.

Versiunea în stil Gödel păstrează povestea mega-Sudoku, dar schimbă demonstrația.

Alege un al doilea sistem de constrângeri de aceeași dimensiune afișată, numește-l **D**. Pentru poveste, S și D sunt două puzzle-uri MegaSudoku(n) în același format. În culise, D poate să fi început ca o formulă logică dificilă de altă dimensiune; dacă e nevoie, poate fi umplută cu constrângeri dummy inofensive pentru a se potrivi aceleiași grile. D este construit dintr-o formulă logică ce este de fapt **nesatisfiabilă**: nu există nicio atribuire de valori care să facă adevărate toate constrângerile sale, exact cum un puzzle stricat nu are grilă completată legal.

Dar D nu trebuie să fie un puzzle stricat ușor de expus. Trebuie să fie fals într-un mod pe care rulebookul ales nu îl poate certifica printr-un argument scurt. Dacă rulebookul ar putea respinge D cu o demonstrație scurtă, povestea de mai jos s-ar prăbuși: ruta alternativă care ar fi putut produce demonstrații fără secretul lui Alice ar fi exclusă formal, iar odată cu ea și garanția de confidențialitate. Deci D este ales dintr-o familie pe care rulebookul fixat nu o poate respinge eficient.

Demonstrația cu un singur mesaj a lui Alice este atunci despre un enunț ori/ori:

> fie mega-Sudoku-ul real S are o soluție, fie decoy-ul D are o soluție.

Aceasta este legătura logică. D **nu** este generat în vreun mod magic care îl face pe S adevărat. Demonstrația nu spune „D nu are soluție, deci S are soluție.” Demonstrează disjuncția **S sau D**. Soliditatea perfectă spune că o disjuncție falsă nu poate avea o demonstrație validă. Cum D este fals în realitate, singurul mod în care disjuncția poate fi adevărată este ca S să fie adevărat. Așadar, dacă demonstrația este acceptată, S trebuie să aibă o soluție.

Dar pentru partea în stil zero-knowledge, întreabă ce s-ar întâmpla dacă D *ar avea* o soluție. Acea soluție-decoy ar funcționa ca witness alternativ. Ar permite cuiva să producă demonstrații fără să cunoască soluția reală mega-Sudoku a lui Alice: un simulator. În realitate D nu are soluție, deci această rută este închisă. Ideea este că rulebookul nu poate dovedi eficient că este închisă.

D are deci două sarcini. Pentru **soliditate**, D este fals, deci o demonstrație validă a lui „S sau D” îl forțează pe S. Pentru **effective zero-knowledge**, D este greu de respins, deci rulebookul nu poate exclude rapid ruta decoy care ar fi făcut posibilă simularea.

Testul de securitate nu mai este:

> Putem dovedi că un simulator există cu adevărat?

Devine:

> Rulebookul tău poate dovedi eficient că simulatorul este imposibil?

Dacă răspunsul este nu, urmează ceva surprinzător de puternic: fiecare garanție de securitate care (a) poate fi observată rulând un test și (b) urmează demonstrabil, în interiorul acelui rulebook, din existența unui simulator, chiar ține. Un atac reușit asupra oricăreia dintre ele ar echivala cu respingerea scurtă lipsă, iar acea respingere scurtă nu există. Aceasta este partea „effective" din effectively zero-knowledge.

Contrastul de clasă este:

Zero-knowledge clasic: demonstrațiile sunt sigure pentru că există un simulator.

Zero-knowledge efectiv în stil Gödel: demonstrațiile sunt tratate ca sigure pentru teste de securitate observabile deoarece rulebookul nu poate dovedi eficient că simulatorul este imposibil.

Al doilea claim este mai slab. Și tocmai de aceea paperul poate păstra cele trei trăsături care rupeau versiunea clasică: un singur mesaj, niciun setup și soliditate perfectă.

Noul test: nu poți dovedi că simulatorul lipsește

Relaxarea lui Ilango schimbă întrebarea.

Zero-knowledge clasic întreabă:

> Există un simulator?

Effectively zero-knowledge întreabă ceva mai slab:

> Rulebookul ales de tine poate dovedi eficient că nu există niciun simulator?

Sună ca o eschivă tehnică, dar este ideea centrală. Construcția trăiește într-o stare ciudată: un simulator nu există de fapt, iar paperul spune explicit acest lucru, dar rulebookul pe care l-ai fixat nu poate dovedi eficient că nu există. Dacă fiecare consecință rea care te interesează ar cere o asemenea respingere, sistemul se comportă totuși ca zero-knowledge pentru acele consecințe.

Aici intră Gödel. Nu ca decor și nu ca „Gödel face criptografia sigură". Legătura este proof-theoretic. Un rulebook se numește **optim** dacă este, într-un sens precis, cel mai bun posibil: ori de câte ori orice rulebook poate respinge o formulă de tipul relevant cu o demonstrație scurtă, și rulebookul optim poate, cu o demonstrație cel mult polinomial mai lungă. Krajíček și Pudlák au conjecturat în 1989 că **nu există niciun sistem de demonstrație optim**. Orice rulebook fixezi, un alt rulebook dovedește o familie de enunțuri adevărate mult mai succint.

Paperul asumă această conjectură într-o formă „infinitely often" puțin mai puternică, standard când conjecturile sunt folosite criptografic. Recompensa, printr-o teoremă a lui Krajíček și Pudlák, este concretă: pentru fiecare rulebook există o secvență de formule cu adevărat nesatisfiabile pe care acel rulebook nu le poate respinge cu demonstrații scurte și pe care un algoritm eficient le poate genera.

Această ultimă proprietate, uniformitatea, transformă ideea dintr-un claim abstract de existență într-un algoritm real pe care Alice îl poate rula: decoy-urile D ies de pe o linie de asamblare.

Mișcarea criptografică este să pună la lucru această lipsă de putere demonstrativă.

Ce face construcția

Iată construcția paperului, despuiată până la formă.

Fixează un rulebook, să zicem ZFC. Sub ipoteza de complexitate a demonstrațiilor, există o secvență generabilă eficient de formule care sunt cu adevărat nesatisfiabile, dar pentru care rulebookul nu are nicio demonstrație scurtă de nesatisfiabilitate.

Acum construiește o demonstrație cu un singur mesaj de forma:

> fie enunțul real este satisfiabil, fie această formulă specială dificilă este satisfiabilă.

Formula specială nu este satisfiabilă. Deci, dacă mecanismul de demonstrație de dedesubt este perfect solid, acceptarea mesajului înseamnă totuși că enunțul real este adevărat. Aceasta dă soliditate perfectă.

Dar pentru securitatea în stil zero-knowledge, imaginează-ți că formula specială *ar fi* satisfiabilă. Atunci witnessul ei ar putea fi folosit pentru a simula demonstrații fără să cunoască witnessul real. Formula nu este satisfiabilă în realitate, dar rulebookul nu poate dovedi eficient asta. Deci nu poate dovedi eficient că simulatorul este imposibil.

Acesta este pivotul. Sistemul nu ascunde secretul producând un simulator clasic. Ascunde secretul, pentru o clasă mare de teste de securitate observabile, în spatele incapacității rulebookului de a certifica faptul că simulatorul lipsește.

Ce afirmă paperul

Teorema principală vine în straturi. Rezultatul central este acesta:

Sub o ipoteză criptografică standard, existența **demonstrațiilor neinteractive witness indistinguishable**, obiecte bine studiate care urmează din mai multe pachete de ipoteze stabilite, și sub conjectura de complexitate a demonstrațiilor că **nu există niciun sistem de demonstrație optim (infinitely often)**, paperul construiește, pentru fiecare alegere de rulebook, un prover și un verifier cu un singur mesaj pentru NP/SAT, cu **soliditate perfectă** și fără setup, care este effectively zero-knowledge relativ la acel rulebook. NP/SAT este denominatorul comun „cel mai dificil” al problemelor de tip puzzle; mega-Sudoku este un costum pe care îl poartă.

Pentru claimul mai larg despre păstrarea proprietăților de securitate falsificabile, paperul adaugă

încă o ipoteză standard, credința de derandomizare **P = BPP**: pe scurt, aleatoriul nu le dă algoritmilor putere esențială în plus.

Tradus din limbajul teoremelor:

- Demonstrația este un singur mesaj.
- Nu există setup de încredere.
- Enunțurile false nu pot fi dovedite.
- Proverul nu este zero-knowledge clasic: nu are simulator.
- Dar fiecare consecință de securitate falsificabilă, bazată pe jocuri, a zero-knowledge clasic poate fi obținută în acest cadru.

„Falsificabil" contează. Înseamnă că un eșec de securitate poate fi testat rulând un adversar într-un joc. Multe definiții criptografice au această formă: poate adversarul să distingă două criptări, să inverseze o funcție, să recupereze un witness sau să câștige un experiment specificat? Teorema dă un prover pentru fiecare proprietate falsificabilă, una câte una. Un singur prover care să se bucure de *toate* proprietățile falsificabile simultan este probabil imposibil: vechiul atac de reutilizare („Bob poate arăta demonstrația altora") este el însuși o proprietate falsificabilă și chiar eșuează aici. Propunerea paperului este că un singur prover poate plauzibil acoperi toate proprietățile falsificabile *naturale*, cele care apar cu adevărat în practica criptografică, dar acea parte este o teoremă condițională sprijinită pe o noțiune informală de „natural", plus o conjectură explicită. Garanția vizează eșecurile observabile, nu fiecare sens filosofic sau simulation-based al secretului.

Merită numit un corolar concret: construcția dă primele demonstrații neinteractive *witness hiding* cu prover uniform, „o demonstrație a unui puzzle nu te ajută să îi găsești soluția", fără interacțiune și fără setup; un obiect care sună modest, dar care rezistase timp de decenii.

Ce nu spune

Aceasta este secțiunea care păstrează piesa onestă.

Nu spune că vechile teoreme de imposibilitate erau greșite. Construcția le evită schimbând definiția.

Nu dă zero-knowledge obișnuit, clasic, fără interacțiune, fără setup și cu soliditate perfectă. Paperul spune explicit că proverul construit nu are simulator.

Nu înseamnă că demonstrația nu poate fi reutilizată. O demonstrație cu un singur mesaj poate fi încă arătată altcuiva; paperul nu păstrează proprietăți de tip deniability. Zero-knowledge neinteractiv cu setup de încredere are aceeași limitare.

Nu înseamnă că acesta este un protocol practic gata de deployment. Este teorie a complexității și fundament al criptografiei. Rezultatul depinde de ipoteze majore din complexitatea demonstrațiilor și criptografie, iar construcția este despre ce este posibil în principiu.

Nu transformă „Gödel" într-o primitivă magică de securitate. Legătura cu Gödel trece prin sisteme de demonstrație, sisteme de demonstrație optime și analogi finiți ai incompletitudinii. Intuiția uti-

lizabilă nu este „incompletitudinea îți protejează parola”. Este: dacă un rulebook nu poate dovedi eficient că un simulator este imposibil, atunci atacurile care ar cere acea demonstrație pot fi blocate la nivelul definițiilor de securitate.

De ce este totuși interesant

Criptografia transformă adesea dificultatea în siguranță. Factorizarea este dificilă, deci ipotezele de tip RSA devin utile. Problemele de rețele sunt dificile, deci criptografia pe rețele devine utilă. Aici dificultatea este mai ciudată: nu „greu de calculat un secret”, ci „greu de dovedit că un anumit obiect de demonstrație nu poate exista”.

De aceea paperul pare neobișnuit. Tratează axiomele și rulebookurile aproape ca resurse criptografice. Imposibilitatea obișnuită spune că există o tensiune între soliditate și simulare. Mișcarea lui Ilango plasează acea tensiune în spatele unei cortine proof-theoretic: simulatorul lipsește, dar sistemul formal nu poate expune eficient absența lui.

Pentru un cititor, partea surprinzătoare nu este că acest lucru va înlocui sistemele zero-knowledge de azi. Probabil nu o va face, cel puțin nu direct. Partea surprinzătoare este că o limită din logica matematică poate fi folosită constructiv: nu doar ca zid, ci ca formă de acoperire.

Cât de puternică este evidența?

Este un paper de teoreme, deci „evidență” înseamnă altceva decât într-un paper de biologie sau astronomie. Întrebarea nu este dacă un experiment s-a replicat. Este dacă definițiile, ipotezele și lanțul de demonstrație susțin claimul.

Demonstrația este formală, iar paperul este explicit în privința ipotezelor. Ipotezele nu sunt întâmplătoare. Demonstrațiile neinteractive witness indistinguishable sunt obiecte standard în criptografie și urmează din mai multe pachete de ipoteze stabilite. Conjectura inexistenței sistemelor de demonstrație optime este o conjectură centrală în complexitatea demonstrațiilor. P = BPP este o credință standard de derandomizare folosită doar pentru teorema mai largă despre proprietăți falsificabile.

Paperul susține și că ipotezele sunt prețul corect, nu o schelă arbitrară: dovedește un convers potrivit căruia ele sunt în esență necesare. Dacă există construcții ca aceasta, atunci trebuie să existe demonstrații neinteractive witness indistinguishable și, acordând funcții one-way standard, nu poate exista niciun sistem de demonstrație optim. Iar ipotezele sunt „win-win”: respingerea oricăreia ar fi deja o descoperire majoră în complexitatea demonstrațiilor, criptografie sau teoria complexității.

Dar pentru că rezultatul este condițional, și încrederea este condițională. Dacă acele ipoteze eșuează, interpretarea teoremei se schimbă. Și chiar dacă se mențin, garanția nu este zero-knowledge clasic deplin; este versiunea relaxată, proof-theoretic, a paperului.

Încrederea corectă este deci ridicată că paperul stabilește un rezultat condițional coerent de posibil-

itate; moderată că ipotezele sale descriu lumea criptografică în care trăim; și scăzută pentru orice consecință practică imediată.

De ce contează

Paperul deschide o rută care trebuia să fie închisă.

Teoria clasică spune: zero-knowledge deplin nu poate fi un singur mesaj fără setup și nu poate fi perfect solid. Paperul lui Ilango spune: dacă cerem consecințele zero-knowledge care pot fi testate în jocuri de securitate și dacă permitem definiției de securitate să depindă de ceea ce un rulebook poate sau nu poate respinge eficient, atunci mare parte din comportamentul util poate fi recuperat, cu un singur mesaj, fără setup și cu soliditate perfectă.

Nu este o mică ajustare de definiție. Este un alt mod de a gândi garanțiile criptografice. În loc să întrebi doar ce există, întreabă ce poate exclude rulebookul tău. În loc să tratezi nedemonstrabilitatea ca pe o neplăcere filosofică, folosește-o ca structură.

Lumea practică poate nu se va schimba mâine. Dar harta conceptuală se schimbă. Există acum un sens formal în care „nimeni nu poate dovedi eficient că secretul s-a scurs” poate fi suficient de puternic pentru a recupera multe dintre protecțiile game-based pe care le doream de la „secretul nu s-a scurs”.

De aceea Gödel este în titlu.

Rezumat curat

Demonstrațiile zero-knowledge îi permit unui prover să convingă un verifier că un enunț este adevărat fără să dezvăluie witnessul. Rezultatele clasice de imposibilitate spun că zero-knowledge nu poate fi comprimat într-un singur mesaj fără setup și nu poate avea soliditate perfectă. Paperul lui Rahul Ilango nu respinge acele imposibilități. Definește o noțiune mai slabă, effectively zero-knowledge: în loc să ceară ca un simulator să existe cu adevărat, cere ca un sistem de demonstrație ales, un rulebook formal precum ZFC, să nu poată dovedi eficient că nu există niciun simulator. Sub ipoteze majore din criptografie (demonstrații neinteractive witness indistinguishable) și complexitatea demonstrațiilor (nu există niciun sistem de demonstrație optim), paperul construiește proveri cu un singur mesaj pentru NP/SAT, fără setup și cu soliditate perfectă, care obțin consecințele falsificabile și game-based ale zero-knowledge, proprietate cu proprietate. Un singur prover care acoperă toate aceste proprietăți „naturale” este o extensie suplimentară, parțial conjecturală; acoperirea literală a fiecărei proprietăți falsificabile este probabil imposibilă, deoarece demonstrațiile rămân reutilizabile. Rezultatul este teoretic și condițional, nu o primitivă deployată, dar arată un nou mod de a folosi nedemonstrabilitatea proof-theoretic ca resursă criptografică.

No-BS check

Ce arată paperul: sub ipoteze declarate, se pot construi proveri cu un singur mesaj, fără setup și perfect solizi pentru NP/SAT, care sunt effectively zero-knowledge relativ la orice sistem de demonstrație ales și care obțin fiecare consecință falsificabilă game-based a zero-knowledge clasic.

Ce este plauzibil, dar nu demonstrat necondiționat: că ipotezele necesare din complexitatea demonstrațiilor și criptografie se mențin. Sunt ipoteze serioase și studiate, iar paperul arată că sunt în esență necesare, nu doar suficiente, dar rămân ipoteze.

Ce nu arată: zero-knowledge clasic fără interacțiune, fără setup și cu soliditate perfectă; un sistem practic gata de deployment; deniability sau nereutilizarea demonstrațiilor; sau că teorema de incompletitudine a lui Gödel, singură, face criptografia sigură.

Limite principale: garanția este o relaxare a zero-knowledge; versiunea mai largă depinde de mai multe ipoteze; claimurile despre un singur prover universal rămân parțial conjecturale; iar rezultatul este în primul rând fundațional.

Câtă încredere ar trebui să aibă un cititor general? Ridicată că acesta este un rezultat teoretic condițional important dacă accepți definițiile. Moderată că ipotezele surprind realitatea. Scăzută pentru deployment practic imediat. Takeaway-ul sigur este: paperul nu rupe imposibilitățile zero-knowledge; găsește un nou mod proof-theoretic de a ocoli părțile care contează pentru multe jocuri de securitate.

Sources

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