The CLEAN PAPER

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

Een cryptografisch bewijs kan zijn geheim verbergen door het moeilijk te maken het ontbreken van een simulator te bewijzen

Zero-knowledgebewijzen laten iemand een uitspraak bewijzen zonder de witness te onthullen. De klassieke theorie zegt dat dit in de volledige betekenis niet kan met één bericht, zonder vertrouwde setup en met perfecte soundness. Rahul Ilango's paper laat die onmogelijkheid niet verdwijnen. Het verandert het doel: in plaats van te eisen dat een simulator werkelijk bestaat, vraagt het dat geen gekozen bewijssysteem efficiënt kan bewijzen dat de simulator niet bestaat. Onder belangrijke aannames uit bewijscomplexiteit en cryptografie is dat genoeg om falsifieerbare, spelgebaseerde gevolgen van zero-knowledge eigenschap voor eigenschap terug te winnen. Het punt is geen kant-en-klare internetprimitief. Het is een bewijstheoretische manier om wiskundige onbewijsbaarheid als cryptografische dekking te gebruiken.

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

De truc is niet bewijzen dat het geheim verborgen is

Begin met de eenvoudigste versie van zero-knowledge.

Alice wil Bob ervan overtuigen dat een Sudoku een oplossing heeft. Als ze de oplossing opstuurt, is Bob overtuigd, maar is de puzzel verpest. Wat ze wil is vreemder: een bewijs dat er een oplossing bestaat, zonder die oplossing prijs te geven.

Dat is de belofte van een zero-knowledgebewijs. De prover (Alice) overtuigt de verifier (Bob) dat een uitspraak waar is, terwijl ze niets onthult behalve het feit dat de uitspraak waar is.

Het probleem is dat die belofte iets kost. Een gewoon wiskundig bewijs heeft twee comfortabele eigenschappen. Het bestaat uit **één bericht**: je schrijft het op, geeft het af en loopt weg. En het heeft **perfecte soundness**: voor een onware uitspraak bestaat helemaal geen geldig bewijs. Klassieke onmogelijkheidsresultaten zeggen dat zero-knowledge beide eigenschappen moet opgeven — en niet alleen de combinatie; elk van de twee is op zichzelf al verboden terrein.

Ten eerste heeft een zero-knowledgebewijs interactie nodig. Als Alice één enkel bericht stuurt, zonder vooraf afgesproken vertrouwde setup, stort de zero-knowledgegarantie in — ongeacht hoeveel soundness je bereid bent daarvoor op te offeren.

Ten tweede heeft een zero-knowledgebewijs een kleine foutmarge nodig. Perfecte soundness eisen blijkt stilletjes ook de interactie kapot te maken: een verifier die nooit voor de gek kan worden gehouden, welke willekeurige keuzes hij ook maakt, kan die keuzes net zo goed vooraf vastleggen — en zodra de verifier voorspelbaar is, kan Alice alles in één bericht beantwoorden. Dat is precies het geval dat al onmogelijk bleek.

Het artikel van Rahul Ilango gaat over een route langs die dubbele muur. Niet door te doen alsof de muur er niet staat, en niet door klassieke zero-knowledge te produceren in een setting waarin dat onmogelijk is. De stap is subtieler: verzwak wat „onthult niets” betekent, maar doe dat op een manier die de beveiligingseigenschappen behoudt die cryptografen daadwerkelijk kunnen testen.

Het resultaat heet **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 wordt bij drie deuren geblokkeerd — interactie, vertrouwde setup en niet-perfecte soundness. Ilango's constructie glipt via een andere deur: het regelboek kan niet efficiënt bewijzen dat de simulator onmogelijk is.

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.

Klassieke zero-knowledge vraagt of er een simulator bestaat; „effectively zero-knowledge" vraagt alleen of het

gekozen regelboek efficiënt kan bewijzen dat er géén kan bestaan. Die zwakkere vraag maakt het mogelijk één bericht, geen setup en perfecte soundness te behouden.

Original diagram — The Clean Paper CC BY 4.0

De oude test: er bestaat een simulator

De klassieke formalisering van zero-knowledge gebruikt een fictieve helper die een **simulator** heet.

Het idee is dit: stel je Jane voor, die Alices geheim **niet** kent. Als Jane helemaal zelf bewijzen kan genereren die er precies zo uitzien als de bewijzen die Bob van Alice zou hebben ontvangen, dan hebben Alices bewijzen Bob niets nieuws geleerd. Jane kon de ervaring immers al namaken zonder Alices geheim.

Klassieke zero-knowledge vraagt dus om een echte simulator. Er moet een efficiënt algoritme bestaan dat overtuigend uitziende bewijzen kan produceren zonder het geheim te kennen — de *witness* in het jargon; bij Sudoku is die witness simpelweg het ingevulde rooster.

Die definitie is krachtig, maar precies daar grijpt de oude onmogelijkheid aan. De intuïtie is als volgt. Een echt niet-interactief bewijs is gewoon een tekenreeks. Zodra Bob die reeks bezit, kan hij haar aan iemand anders tonen: hij heeft het vermogen gekregen om de uitspraak tegenover anderen te bewijzen, en dat klinkt al als meer dan „niets”. De klassieke stellingen maken die intuïtie scherp in de onmogelijkheden hierboven.

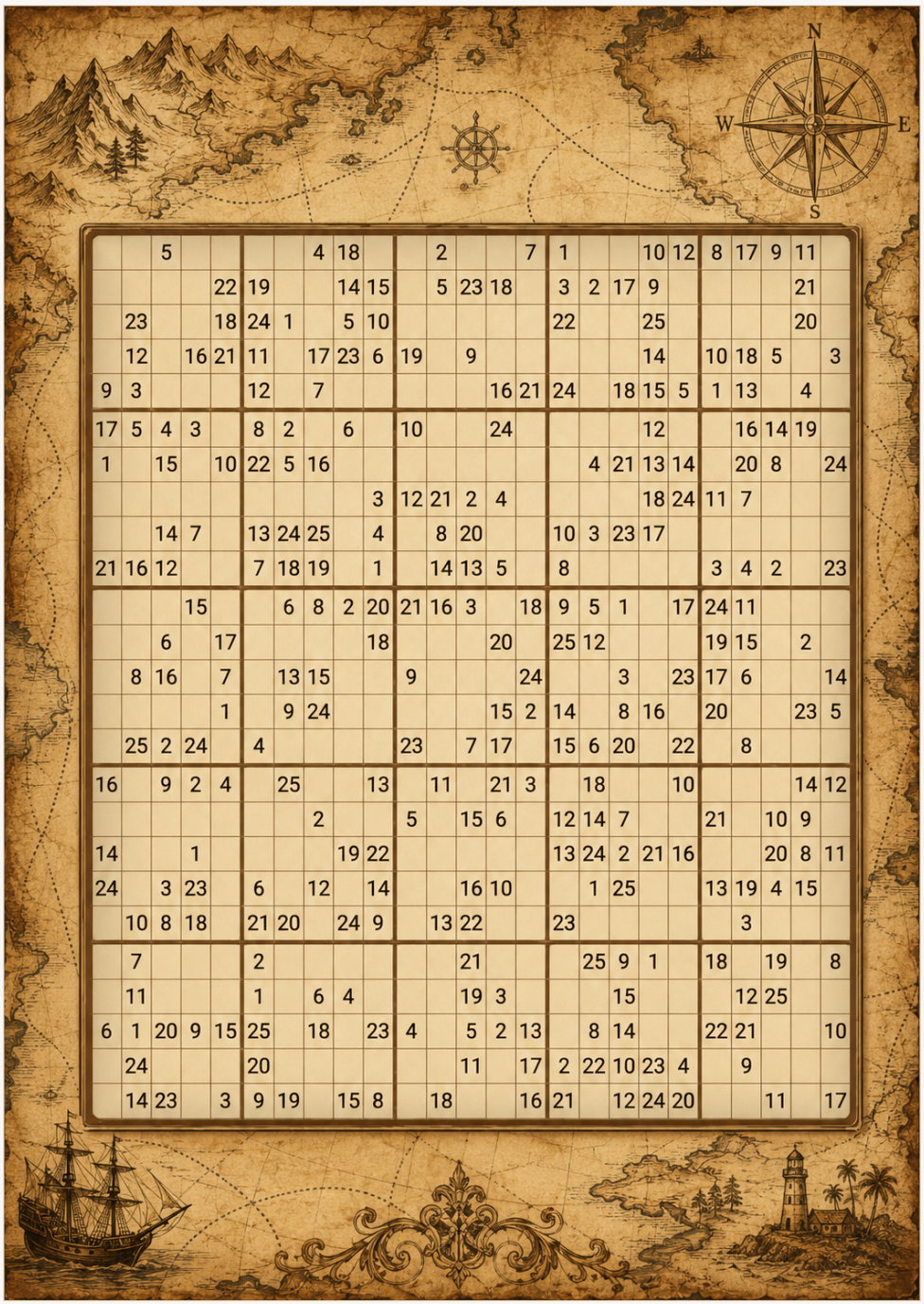
> **De drie eigenschappen waarop dit artikel staat**
>
> De titel van het artikel noemt drie beperkingen:
> **Geen interactie:** Alice stuurt één bewijsreeks. Er is geen heen-en-weerprotocol.
> **Geen setup:** Alice en Bob vertrouwen niet op een vertrouwde gemeenschappelijke referentiereeks of andere vooraf afgesproken publieke willekeur. Veel systemen die „non-interactive zero-knowledge” heten gebruiken nog steeds een setup; hier betekent het echt nul setup.
> **Perfecte soundness:** een onware uitspraak heeft geen geldig bewijs. Niet „wordt bijna nooit geaccepteerd”; er bestaat géén geldig bewijs.
> Dat zijn precies de drie eigenschappen die een gewoon geschreven wiskundig bewijs heeft — en, zoals hierboven uitgelegd, klassieke zero-knowledge kan ze niet behouden.

Een mega-Sudokuversie van het verschil

Hier is een bewust vereenvoudigde manier om het verschil aan te voelen.

Gebruik voor het serieuze deel van de analogie geen gewone Sudoku van 9 bij 9. Die is te klein en te eindig: een computer kan hem simpelweg oplossen, of bewijzen dat er geen oplossing is. Stel je in plaats daarvan een familie van puzzels **MegaSudoku(n)** voor. Schaal de gewone regel op: kies

een blokgrootte n, neem $N = n^2$ en bouw een N bij N-rooster, verdeeld in n bij n-blokken, met N symbolen. Gewone Sudoku is slechts het kleine geval $n = 3$, $N = 9$: een rooster van 9 bij 9, blokken van 3 bij 3 en negen symbolen. Het verhaal uit de bewijscomplexiteit begint pas wanneer n mag groeien en het rooster extra gadgets kan bevatten waardoor het zich gedraagt als een SAT-formule in Sudokuvermomming. Een SAT-formule is simpelweg een lijst ja/nee-beperkingen: kun je waar/onwaarwaarden aan variabelen toekennen zodat elke beperking wordt voldaan?



		5					4	18			2			7	1			10	12	8	17	9	11	
				22	19			14	15		5	23	18		3	2	17	9					21	
	23			18	24	1		5	10						22			25					20	
	12		16	21	11		17	23	6	19		9						14		10	18	5		3
9	3				12		7						16	21	24		18	15	5	1	13		4	
17	5	4	3		8	2		6		10			24					12			16	14	19	
1		15		10	22	5	16									4	21	13	14		20	8		24
									3	12	21	2	4					18	24	11	7			
		14	7		13	24	25		4		8	20			10	3	23	17						
21	16	12			7	18	19		1		14	13	5		8					3	4	2		23
			15			6	8	2	20	21	16	3		18	9	5	1		17	24	11			
		6		17					18				20		25	12				19	15		2	
	8	16		7		13	15			9				24			3		23	17	6			14
				1		9	24						15	2	14		8	16		20			23	5
	25	2	24		4					23		7	17		15	6	20		22		8			
16		9	2	4		25			13		11		21	3		18			10				14	12
							2			5		15	6		12	14	7			21		10	9	
14			1					19	22						13	24	2	21	16			20	8	11
24		3	23		6		12		14			16	10			1	25			13	19	4	15	
	10	8	18		21	20		24	9		13	22			23						3			
	7				2							21				25	9	1		18		19		8
	11				1		6	4				19	3				15				12	25		
6	1	20	9	15	25		18		23	4		5	2	13		8	14			22	21			10
	24				20							11		17	2	22	10	23	4		9			
	14	23		3	9	19		15	8		18			16	21		12	24	20			11		17

Een Sudoku van 25 bij 25: de regels kunnen worden gecontroleerd zonder het voltooide rooster te onthullen — een visuele plaatsvervanger voor een bewijs dat een verborgen oplossing, de witness, verifieert.

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

Sudoku en SAT: dezelfde puzzel in twee vermommingen

De bewering dat een Sudoku zich „als een SAT-formule kan gedragen” is geen metafoor. De omzetting werkt in beide richtingen, en de makkelijke richting kan volledig worden uitgeschreven.

Van Sudoku naar SAT. SAT spreekt alleen waar/onwaar, dus geef het één booleaanse variabele per drietal (rij, kolom, waarde): *x(r,c,v)* betekent „de cel in rij *r*, kolom *c* bevat waarde *v*”. Een Sudoku van 4 bij 4 (blokken van 2 bij 2, waarden 1–4) heeft 4·4·4 = 64 variabelen nodig; de klassieke 9 bij 9 heeft er 729. Elke Sudokuregel wordt vervolgens een verzameling clausules. (Een clausule is een OF van variabelen of hun negaties; de hele formule is de EN van alle clausules.)

Elke cel bevat minstens één waarde — één clausule per cel:

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

Elke cel bevat hoogstens één waarde — een „niet allebei”-clausule voor elk waardenpaar:

$$\neg x(1,1,1) \lor \neg x(1,1,2) \quad \neg x(1,1,1) \lor \neg x(1,1,3) \quad \text{... enzovoort voor alle zes paren.}$$

Elke rij bevat elke waarde — voor rij 1 en waarde 3: minstens één keer,

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

en hoogstens één keer: $\neg x(1,1,3) \lor \neg x(1,2,3)$, enzovoort voor elk paar cellen in de rij.

Kolommen en blokken — dezelfde verzamelingen; alleen de groep cellen verandert. Voor het blok linksboven en waarde 2:

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

plus de paarsgewijze „niet allebei”-clausules.

De voorgedrukte aanwijzingen — het eenvoudigste deel: elke aanwijzing is een clausule met één variabele. Een voorgedrukte 3 linksboven wordt de clausule

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

De EN van dit alles is precies dan satisfieerbaar wanneer de Sudoku een oplossing heeft — en een bevredigende toekenning *is* de oplossing: lees af welke x(r,c,v) waar zijn en vul het rooster in. Voor een 9-bij-9-Sudoku komt dit neer op 729 variabelen en enkele duizenden clausules, die een moderne SAT-solver in milliseconden afhandelt. Let op de aanwijzingsclausule x(1,1,3): die zegt „deze cel is exact 3”, niet „deze cellen zijn allemaal verschillend” — dezelfde asymmetrie die verderop in de protocolnotitie een extra truc voor aanwijzingscellen nodig maakt.

Van SAT naar Sudoku. Het artikel heeft de omgekeerde, moeilijkere richting nodig: neem een *willekeurige* SAT-formule en bouw een mega-Sudoku die precies dan een oplossing heeft wanneer de formule er een heeft. Sudokus eigen regels kunnen alleen zeggen „deze cellen zijn allemaal verschillend”, dus willekeurige logische beperkingen moeten worden *gebouwd* — en dat is precies wat de gadgets doen. Een gadget is een kleine vooraf ontworpen cluster cellen,

> één per clausule van de formule, waarin aangewezen cellen de rol van variabelen spelen (het symbool dat ze bevatten codeert waar of onwaar) en de interne beperkingen zo zijn ontworpen dat alleen invullingen die de clausule vervullen geldig zijn. Dit is standaardvakmanschap uit NP-volledigheidsbewijzen; voor gegeneraliseerde Sudoku werd het in 2003 uitgewerkt door Yato en Seta.
> Samen zeggen de twee richtingen dat N-bij-N-Sudoku en SAT hetzelfde probleem in verschillende kleren zijn. Daardoor kan dit artikel — en het paper — een verhaal over heel NP vertellen met roosters en symbolen.

De witness blijft gemakkelijk voor te stellen. Alice kent een volledige geldige invulling van de mega-Sudoku. Bob wil overtuigd worden dat zo'n invulling bestaat, maar Alice wil haar niet onthullen. Stuurt ze het hele rooster, dan is Bob overtuigd maar is het geheim weg.

In de klassieke zero-knowledgeversie werken Alice en Bob interactief. Een oud mentaal model gebruikt afgedekte tegels. Alice verbergt het opgeloste rooster, hernoemt de symbolen vóór elke ronde in het geheim en laat Bob één willekeurig gekozen lokale beperking controleren: een rij, een kolom, een blok of een gadget. Als de geopende cellen allemaal verschillende symbolen tonen, krijgt Bob meer vertrouwen. Daarna wordt alles weer afgedekt en worden de symbolen opnieuw willekeurig hernoemd. (Er is één complicatie: de gegeven aanwijzingen van de puzzel vereisen een extra truc, omdat het hernoemen de aanwijzingen zelf ook verbergt. De notitie hieronder legt uit hoe klassieke protocollen dit oplossen; voor wat volgt is dit speelgoedmodel voldoende.)

> **Hoe klassieke protocollen de aanwijzingscellen werkelijk behandelen**
>
> De hernoemtruc heeft een blinde vlek. De regels voor rijen, kolommen en blokken zeggen allemaal „deze cellen zijn allemaal verschillend", en *allemaal verschillend* blijft waar onder elke hernoeming van de symbolen. Maar een aanwijzing zegt „deze cel bevat exact 5", en na hernoeming ziet Bob alleen $\sigma(5)$ — een gemaskeerd symbool — zonder de hernoeming σ te kennen. Hij kan dus niets controleren. Zonder reparatie zou Alice kunnen bewijzen dat *een of ander* geldig rooster bestaat terwijl ze de voorgedrukte aanwijzingen volledig negeert, en dat bewijst niets over *deze* puzzel. De klassieke literatuur kent twee standaardoplossingen.
> **Het palet.** Voeg één extra rij van N cellen toe aan het verborgen rooster — een palet dat Alice vult met de symbolen 1…N in een vaste publieke volgorde en vervolgens samen met al het andere hernoemt, zodat het $\sigma(1)\ldots\sigma(N)$ bevat. Bobs willekeurige uitdaging krijgt nu één extra optie. Naast een rij, kolom, blok of gadget kan hij **het palet plus één aanwijzingscel** kiezen. Alice legt beide bloot; het palet onthult de hernoeming van die ronde, en Bob controleert dat de aanwijzingscel exact de hernoemde versie van de voorgedrukte aanwijzing toont. Dit blijft zero-knowledge omdat Bob alleen σ leert — elke ronde opnieuw willekeurig gekozen en op zichzelf waardeloos — plus de waarde van een cel die hij al uit de puzzel kende. Niets over de geheime cellen lekt, en een simulator kan het beeld namaken door een willekeurige σ te trekken. Het is sound omdat een frauderende Alice per ronde met een vaste kans wordt betrapt, en rondes worden herhaald totdat de resterende twijfel verwaarloosbaar is.
> **De aanwijzingen wegcompileren.** Een structurelere variant verwijdert de speciale uitdaging

> in plaats van haar toe te voegen. In plaats van de waarde van de aanwijzing te *verifiëren*, dwing je die af met ongelijkheidsbeperkingen: verbind de aanwijzingscel met elke paletcel behalve degene die zijn eigen waarde draagt — „verschillend van $\sigma(1)$, verschillend van $\sigma(2)$, …, verschillend van alles behalve $\sigma(5)$". Het enige symbool dat de cel dan legaal kan bevatten is dat van de aanwijzing. Elke beperking is weer van de vorm „deze twee verschillen" — invariant onder hernoeming en net als een rij controleerbaar. Dit is dezelfde manoeuvre als voor vooraf gekleurde hoekpunten in het klassieke graafkleuringsprotocol en precies de geest van het woord *gadgets* hierboven: in het MegaSudoku-als-SAT-beeld worden de aanwijzingen net als alle andere beperkingen gecompileerd tot ongelijkheidsgadgets.
> **Het fysieke protocol.** Het echte kaartprotocol voor Sudoku (Gradwohl, Naor, Pinkas en Rothblum, 2007) gebruikt helemaal geen hernoeming en handelt de aanwijzingen af voordat het verbergen begint. Voor elke cel legt Alice drie identieke kaarten met de waarde van de cel neer — met de afbeelding omlaag voor geheime cellen, maar **met de afbeelding omhoog voor aanwijzingscellen**, zodat Bob met eigen ogen ziet dat de aanwijzingen worden gevolgd voordat de kaarten worden omgedraaid. Daarna gaat één kaart van elke cel naar het pakket van zijn rij, één naar dat van zijn kolom en één naar dat van zijn blok; elk pakket wordt geschud en onthuld, waarna Bob controleert dat alle N symbolen aanwezig zijn. Het schudden vernietigt de positie-informatie (dat is het zero-knowledgegedeelte), maar de aanwijzingen waren al bij het neerleggen vastgezet.
> Hoe dan ook is de les dezelfde waar dit artikel telkens op terugkomt: een zero-knowledgeprotocol is zorgvuldige boekhouding van *welke* feiten het verbergen overleven. Hernoemen behoudt „allemaal verschillend" en wist „is gelijk aan 5" — dus „is gelijk aan 5" moet via een andere route weer worden ingebracht.

Dat is niet het protocol uit Ilango's paper. Het is het mentale model voor klassieke zero-knowledge:

- Alice en Bob gaan heen en weer.
- Bob kiest willekeurige controles.
- Alice onthult alleen lokale consistentie, niet de hele oplossing.
- Het privacybewijs werkt door te laten zien dat Bobs zicht op het protocol ook zonder Alices geheime oplossing gegenereerd had kunnen worden.

Klassieke zero-knowledge draait dus om een positief feit:

> Er bestaat werkelijk een simulator.

Haal nu de comfortabele onderdelen weg. Alice stuurt één bewijsreeks en loopt weg. Er is geen vertrouwde setup, geen vooraf voorbereide gedeelde willekeurige reeks, en Bob mag een onware puzzel nooit accepteren. Dit is de setting waarin klassieke zero-knowledge niet kan overleven.

Voor de truc is nog één personage nodig. Leg een **regelboek** vast: een formeel bewijssysteem in de betekenis van de logicus — een vaste verzameling axioma's plus mechanische regels om geschreven wiskundige bewijzen te controleren. ZFC, de standaardaxioma's van de wiskunde, is het canonieke voorbeeld. Alles vanaf hier wordt geformuleerd ten opzichte van een vooraf gekozen regelboek, en

die keuze is flexibel: de constructie werkt voor elk regelboek dat je vastlegt, inclusief ZFC.

(Een terminologische opmerking uit het paper zelf: „proof system" betekent hier altijd dit regelboek — het formele systeem dat wiskundige bewijzen controleert — en nooit de berichten die Alice verstuurt. De machines van Alice en Bob heten „de prover en de verifier".)

De Gödel-achtige versie behoudt het mega-Sudokuverhaal, maar verandert het bewijs.

Kies een tweede beperkingssysteem van dezelfde weergegeven grootte en noem het **D**. In het verhaal zijn S en D twee MegaSudoku(n)-puzzels in hetzelfde formaat. Achter de schermen kan D begonnen zijn als een moeilijke logische formule van een andere grootte; indien nodig kan zij met onschadelijke dummybeperkingen worden opgevuld zodat ze in hetzelfde rooster past. D is opgebouwd uit een logische formule die werkelijk **onsatisfieerbaar** is: er bestaat geen toekenning van waarden die alle beperkingen tegelijk waar maakt, net zoals een kapotte puzzel geen geldige volledige invulling heeft. Een speelgoedvoorbeeld is een formule die tegelijk eist dat „X waar is" en „X onwaar is". D heeft dus geen geldige invulling.

Maar D mag geen kapotte puzzel zijn waarvan de fout *gemakkelijk aan te tonen* is. Het speelgoedvoorbeeld hierboven faalt precies daarom: elk regelboek weerlegt „X en niet-X" in één regel. D moet onwaar zijn op een manier die het gekozen regelboek niet met een kort argument kan certificeren. Als het regelboek D met een kort bewijs kon weerleggen, stort het verhaal hieronder in: de alternatieve route die misschien bewijzen zonder Alices geheim had kunnen produceren, zou formeel uitgesloten kunnen worden, en daarmee ook de privacygarantie. D wordt daarom gekozen uit een familie die het vaste regelboek niet efficiënt kan weerleggen: binnen dat regelboek bestaat geen kort bewijs dat D geen oplossing heeft.

Alices bewijs uit één bericht gaat vervolgens over een of/of-uitspraak:

> ofwel heeft de echte mega-Sudoku S een oplossing, ofwel heeft de afleidingspuzzel D een oplossing.

Dit is de logische koppeling. D wordt **niet** op een magische manier gegenereerd waardoor S waar wordt. Het bewijs redeneert niet „D heeft geen oplossing, dus S heeft een oplossing". Het bewijst de disjunctie **S of D**. Perfecte soundness zegt dat een onware disjunctie geen geldig bewijs kan hebben. Omdat D in werkelijkheid onwaar is — er bestaat geen oplossing — kan de disjunctie alleen waar zijn als S waar is. Als het bewijs wordt geaccepteerd, moet S dus een oplossing hebben. De afleidingspuzzel kan een onware S niet waar maken.

Maar voor het zero-knowledgeachtige deel vraag je wat er zou gebeuren als D *wel* een oplossing had. Die afleidingsoplossing zou als alternatieve witness kunnen dienen. Ze zou iemand in staat stellen bewijzen te produceren zonder Alices echte mega-Sudokuoplossing te kennen — met andere woorden: als simulator. In werkelijkheid heeft D geen oplossing, dus deze simulatorroute is gesloten. Het punt is dat het regelboek niet efficiënt kan bewijzen dat ze gesloten is.

D heeft dus twee taken. Voor **soundness** is D onwaar, zodat een geldig bewijs van „S of D" S afdwingt. Voor **effectively zero-knowledge** is D moeilijk te weerleggen, zodat het regelboek niet snel de afleidingsroute kan uitsluiten die simulatie mogelijk zou hebben gemaakt.

De beveiligingstest is daarom niet langer:

> Kunnen we bewijzen dat er werkelijk een simulator bestaat?

Hij wordt:

> Kan je regelboek efficiënt bewijzen dat een simulator onmogelijk is?

Als het antwoord nee is, volgt iets verrassend sterks: elke beveiligingsgarantie die (a) kan worden waargenomen door een test uit te voeren en (b) aantoonbaar — binnen dat regelboek — volgt uit het bestaan van een simulator, geldt daadwerkelijk. Een succesvolle aanval op zo'n garantie zou zelf neerkomen op de ontbrekende korte weerlegging, en die korte weerlegging bestaat niet. Dat is het „effective"-gedeelte van effectively zero-knowledge.

Het contrast voor in de klas is dus:

Klassieke zero-knowledge: de bewijzen zijn veilig omdat er een simulator bestaat.

Gödel-achtige effectively zero-knowledge: de bewijzen worden voor waarneembare beveiligingstests als veilig behandeld omdat het regelboek niet efficiënt kan bewijzen dat de simulator onmogelijk is.

De tweede uitspraak is zwakker. En juist daardoor kan het paper de drie eigenschappen behouden waarop de klassieke versie brak: één bericht, geen setup en perfecte soundness.

De nieuwe test: je kunt niet bewijzen dat de simulator ontbreekt

Ilango's versoepeling verandert de vraag.

Klassieke zero-knowledge vraagt:

> Bestaat er een simulator?

Effectively zero-knowledge vraagt iets zwakkers:

> Kan je gekozen regelboek efficiënt bewijzen dat er geen simulator bestaat?

Dat klinkt als een technische uitvlucht, maar het is het kernidee. De constructie bevindt zich in een vreemde toestand: er bestaat in werkelijkheid geen simulator — het paper is daar expliciet over — maar het regelboek dat je hebt vastgelegd kan niet efficiënt bewijzen dat die er niet is. Als elk slecht

gevolg waar je om geeft zo'n weerlegging zou vereisen, gedraagt het systeem zich voor die gevolgen toch als zero-knowledge.

Hier komt Gödel binnen. Niet als decoratie, en niet als „Gödel maakt crypto veilig". De verbinding is bewijstheoretisch. Een regelboek heet **optimaal** als het in precieze zin het best mogelijke is: wanneer enig regelboek een formule van de relevante soort met een kort bewijs kan weerleggen, kan het optimale regelboek dat ook, met een bewijs dat hoogstens polynomiaal langer is. Krajíček en Pudlák conjectureerden in 1989 dat **geen optimaal bewijssysteem bestaat**: welk regelboek je ook vastlegt, een ander regelboek kan sommige families ware uitspraken veel bondiger bewijzen. Dit is een van de centrale open vermoedens in de bewijscomplexiteit en de eindige, complexiteitstheoretische neef van Gödels onvolledigheidsstelling: sommige ware uitspraken hebben geen kort bewijs in het regelboek dat je hebt gekozen — niet omdat ze in principe onbewijsbaar zijn, maar omdat elk vast regelboek bepaalde korte waarheden zonder korte bewijzen laat.

Het paper neemt dit vermoeden aan (in een iets sterkere „infinitely often"-vorm, gebruikelijk wanneer vermoedens cryptografisch worden ingezet). De opbrengst, via een stelling van Krajíček en Pudlák, is concreet: voor elk regelboek bestaat een reeks formules die werkelijk onsatisfieerbaar zijn en die het regelboek niet met korte bewijzen kan weerleggen — en, cruciaal, die een efficiënt algoritme kan *genereren*. Die laatste eigenschap, uniformiteit, verandert het geheel van een bestaansclaim in een werkelijk algoritme dat Alice kan uitvoeren: haar afleidingsformules D rollen van een lopende band, ze worden niet uit het niets getoverd.

De cryptografische stap is dat tekort aan bewijskracht aan het werk zetten.

Wat de constructie doet

Hier is de constructie uit het paper, teruggebracht tot haar vorm.

Leg een regelboek vast — bijvoorbeeld ZFC. Onder de aanname uit de bewijscomplexiteit bestaat er een efficiënt te genereren reeks formules die werkelijk onsatisfieerbaar zijn, maar waarvoor het regelboek geen kort bewijs van onsatisfieerbaarheid heeft.

Bouw nu een bewijs uit één bericht van de vorm:

> ofwel is de echte uitspraak satisfieerbaar, ofwel is deze speciale moeilijke formule satisfieerbaar.

De speciale moeilijke formule is niet satisfieerbaar. Als het onderliggende bewijsmechanisme perfecte soundness heeft, betekent het accepteren van het bericht dus nog steeds dat de echte uitspraak waar is. Dat geeft perfecte soundness.

Maar stel voor de zero-knowledgeachtige beveiliging dat de speciale moeilijke formule *wel* satisfieerbaar was. Dan zou haar witness kunnen worden gebruikt om bewijzen te simuleren zonder de echte witness te kennen. In werkelijkheid is de formule niet satisfieerbaar — maar het regelboek kan dat niet efficiënt bewijzen. Het kan dus ook niet efficiënt bewijzen dat de simulator onmogelijk is.

Dat is het scharnierpunt. Het systeem verbergt het geheim niet door een klassieke simulator te produceren. Voor een grote klasse waarneembare beveiligingstests verbergt het het geheim achter het onvermogen van het regelboek om te certificeren dat de simulator ontbreekt.

Wat het paper beweert

De hoofdstelling komt in lagen. Het kernresultaat is dit:

Onder een standaard cryptografische aanname — het bestaan van **non-interactive witness indistinguishable proofs**, goed bestudeerde objecten die uit verschillende gevestigde pakketten van aannames volgen — en onder het vermoeden uit de bewijscomplexiteit dat **geen (infinitely often) optimaal bewijssysteem bestaat**, construeert het paper voor elk gekozen regelboek een prover en verifier met één bericht voor NP/SAT, met **perfecte soundness** en zonder setup, die relatief aan dat regelboek effectively zero-knowledge zijn. (NP/SAT is de standaard „moeilijkste gemene deler" van puzzelachtige problemen; mega-Sudoku is één van de vermommingen ervan.)

Voor de bredere claim over het behoud van falsifieerbare beveiligingseigenschappen voegt het paper nog één standaardaanname toe, het derandomisatievermoeden **P = BPP** (grofweg: willekeur geeft algoritmen geen wezenlijk extra vermogen).

Uit stellingentaal vertaald:

- Het bewijs is één bericht.
- Er is geen vertrouwde setup.
- Onware uitspraken kunnen niet worden bewezen.
- De prover is niet klassiek zero-knowledge — hij heeft geen simulator.
- Maar elke falsifieerbare, spelgebaseerde beveiligingsconsequentie van klassieke zero-knowledge kan in deze setting worden bereikt.

„Falsifieerbaar" is belangrijk. Het betekent dat een beveiligingsfout kan worden getest door een tegenstander in een spel te laten draaien. Veel cryptografische beveiligingsdefinities hebben deze vorm: kan de tegenstander twee versleutelingen onderscheiden, een functie inverteren, een witness terugvinden of een bepaald experiment winnen? De stelling geeft voor elke falsifieerbare eigenschap afzonderlijk een prover. Eén prover die *alle* falsifieerbare eigenschappen tegelijk bezit is waarschijnlijk onmogelijk — de oude hergebruikaanval („Bob kan het bewijs aan anderen laten zien") is zelf een falsifieerbare eigenschap, en die faalt hier daadwerkelijk. Het voorstel van het paper is dat één prover plausibel alle *natuurlijke* falsifieerbare eigenschappen kan dekken — degene die werkelijk in cryptografische praktijk voorkomen — maar dat deel is een conditionele stelling die rust op een informele notie van „natuurlijk", plus een expliciet vermoeden. De garantie richt zich op waarneembare fouten, niet op elke filosofische of simulatiegebaseerde betekenis van geheimhouding.

Eén concrete gevolgtrekking verdient een naam: de constructie levert de eerste niet-interactieve *witness-hiding*-bewijzen met een uniforme prover — „een bewijs dat een puzzel een oplossing heeft

helpt je niet die oplossing te vinden", zonder interactie en zonder setup — een bescheiden klinkend object dat zich decennialang aan constructie had onttrokken.

Wat dit niet zegt

Dit is het gedeelte dat het verhaal eerlijk houdt.

Het zegt **niet** dat de oude onmogelijkheidsstellingen fout waren. De constructie ontwijkt ze door de definitie te veranderen.

Het levert **geen** gewone, klassieke zero-knowledge met geen interactie, geen setup en perfecte soundness. Het paper zegt expliciet dat de geconstrueerde prover geen simulator heeft.

Het betekent **niet** dat het bewijs niet kan worden hergebruikt. Een bewijs uit één bericht kan nog steeds aan iemand anders worden getoond; het paper behoudt geen eigenschappen in de sfeer van ontkenbaarheid. (Niet-interactieve zero-knowledge met vertrouwde setup heeft dezelfde beperking.)

Het betekent **niet** dat dit een praktisch protocol is dat klaar is voor uitrol. Dit is complexiteitstheorie en de grondslagen van cryptografie. Het resultaat hangt af van grote aannames uit bewijscomplexiteit en cryptografie, en de constructie gaat over wat in principe mogelijk is.

Het maakt „Gödel" **niet** tot een magische beveiligingsprimitief. De Gödelverbinding loopt via bewijssystemen, optimale bewijssystemen en eindige analogen van onvolledigheid. De bruikbare intuïtie is niet „onvolledigheid beschermt je wachtwoord". Ze is: als een regelboek niet efficiënt kan bewijzen dat een simulator onmogelijk is, dan kunnen aanvallen die zo'n bewijs zouden vereisen op het niveau van beveiligingsdefinities worden geblokkeerd.

Waarom het toch interessant is

Cryptografie zet moeilijkheid vaak om in veiligheid. Factoriseren is moeilijk, dus RSA-achtige aannames worden bruikbaar. Roosterproblemen zijn moeilijk, dus roostercryptografie wordt bruikbaar. Hier is de moeilijkheid vreemder: niet „het is moeilijk een geheim te berekenen", maar „het is moeilijk te bewijzen dat een bepaald bewijsobject niet kan bestaan".

Daarom voelt het paper ongewoon. Het behandelt axioma's en regelboeken bijna als cryptografische hulpbronnen. De gebruikelijke onmogelijkheid zegt dat er spanning bestaat tussen soundness en simulatie. Ilango's stap plaatst die spanning achter een bewijstheoretisch gordijn: de simulator ontbreekt, maar het formele systeem kan dat ontbreken niet efficiënt blootleggen.

Voor een lezer is het verrassende niet dat dit de huidige zero-knowledgesystemen zal vervangen. Dat zal waarschijnlijk niet gebeuren, althans niet rechtstreeks. Het verrassende is dat een beperking uit de wiskundige logica constructief kan worden gebruikt: niet alleen als muur, maar als een soort dekking.

Hoe sterk is het bewijs?

Dit is een paper met stellingen, dus „bewijs” betekent hier iets anders dan in een biologisch of astronomisch artikel. De vraag is niet of een experiment is gerepliceerd. De vraag is of definities, aannames en de bewijsredenering de claim dragen.

Het bewijs is formeel en het paper is expliciet over zijn aannames. Die aannames zijn niet terloops gekozen. Non-interactive witness indistinguishable proofs zijn standaardobjecten in de cryptografie en volgen uit verschillende gevestigde pakketten van aannames. Het vermoeden dat er geen optimaal bewijssysteem bestaat is een centraal vermoeden in de bewijscomplexiteit. P = BPP is een standaard derandomisatievermoeden dat alleen wordt gebruikt voor de bredere stelling over falsifieerbare eigenschappen.

Het paper betoogt ook dat die aannames de juiste prijs zijn en geen willekeurig steigerwerk: het bewijst een omgekeerde richting waaruit volgt dat ze in wezen noodzakelijk zijn — als zulke constructies überhaupt bestaan, moeten non-interactive witness indistinguishable proofs bestaan en kan er (gegeven standaard one-way functions) geen optimaal bewijssysteem bestaan. De aannames zijn bovendien „win-win”: het weerleggen van één ervan zou op zichzelf een baanbrekende ontdekking zijn in bewijscomplexiteit, cryptografie of complexiteitstheorie.

Maar omdat het resultaat conditioneel is, is ook het vertrouwen conditioneel. Als die aannames falen, verandert de interpretatie van de stelling. En zelfs als ze standhouden, is de garantie geen volledige klassieke zero-knowledge; het is de versoepelde, bewijstheoretische versie uit het paper.

Het juiste vertrouwensniveau is dus: hoog dat het paper een coherent conditioneel mogelijkheidsresultaat vaststelt; gematigd dat de aannames de cryptografische wereld beschrijven waarin we werkelijk leven; en laag voor onmiddellijke praktische gevolgen.

Waarom het ertoe doet

Het paper opent een route die gesloten hoorde te zijn.

De klassieke theorie zegt: volledige zero-knowledge kan zonder setup niet uit één bericht bestaan en kan geen perfecte soundness hebben. Ilango’s paper zegt: als we vragen naar de gevolgen van zero-knowledge die in beveiligingsspellen kunnen worden getest, en als de beveiligingsdefinitie mag afhangen van wat een regelboek wel of niet efficiënt kan weerleggen, dan kan veel van het nuttige gedrag worden teruggewonnen — met één bericht, zonder setup en met perfecte soundness.

Dat is geen kleine aanpassing van een definitie. Het is een andere manier om over cryptografische garanties na te denken. Vraag niet alleen wat bestaat, maar ook wat je regelboek kan uitsluiten. Behandel onbewijsbaarheid niet alleen als filosofische hinder, maar gebruik haar als structuur.

De praktische wereld verandert morgen misschien niet. Maar de conceptuele kaart verandert wel. Er bestaat nu een formele betekenis waarin „niemand kan efficiënt bewijzen dat het geheim is gelekt” sterk genoeg kan zijn om veel van de spelgebaseerde beschermingen terug te krijgen die we wilden

van „het geheim is niet gelekt”.

Daarom hoort Gödel in de titel.

Heldere samenvatting

Zero-knowledgebewijzen laten een prover een verifier overtuigen dat een uitspraak waar is zonder de witness te onthullen. Klassieke onmogelijkheidsresultaten zeggen dat zero-knowledge zonder setup niet in één bericht kan worden geperst en geen perfecte soundness kan hebben. Rahul Ilango’s paper weerlegt die onmogelijkheden niet. Het definieert een zwakker begrip, effectively zero-knowledge: in plaats van te eisen dat er werkelijk een simulator bestaat, eist het dat een gekozen bewijssysteem — een formeel regelboek zoals ZFC — niet efficiënt kan bewijzen dat er géén simulator bestaat. Onder belangrijke aannames uit de cryptografie (non-interactive witness indistinguishable proofs) en bewijscomplexiteit (er bestaat geen optimaal bewijssysteem) construeert het paper provers uit één bericht voor NP/SAT, zonder setup en met perfecte soundness, die de falsifieerbare, spelgebaseerde gevolgen van zero-knowledge eigenschap voor eigenschap realiseren. Eén prover die alle „natuurlijke” zulke eigenschappen tegelijk dekt is een verdere, deels conjecturele uitbreiding — en letterlijk elke falsifieerbare eigenschap tegelijk dekken is waarschijnlijk onmogelijk, omdat bewijzen herbruikbaar blijven. Het resultaat is theoretisch en conditioneel, geen uitgerolde primitief, maar het toont een nieuwe manier om bewijstheoretische onbewijsbaarheid als cryptografische hulpbron te gebruiken.

Zonder omwegen

Wat het paper laat zien: Onder de vermelde aannames kunnen provers voor NP/SAT worden gebouwd die één bericht gebruiken, geen setup nodig hebben en perfecte soundness bezitten, die relatief aan elk gekozen bewijssysteem effectively zero-knowledge zijn en elke afzonderlijke falsifieerbare, spelgebaseerde consequentie van klassieke zero-knowledge kunnen realiseren.

Wat plausibel is maar niet onvoorwaardelijk bewezen: Dat de benodigde aannames uit bewijscomplexiteit en cryptografie kloppen. Het zijn serieuze, veel bestudeerde aannames — en het paper laat zien dat ze in wezen zowel noodzakelijk als voldoende zijn — maar het blijven aannames.

Wat het niet laat zien: Klassieke zero-knowledge zonder interactie, zonder setup en met perfecte soundness; een praktisch systeem dat klaar is voor uitrol; ontkenbaarheid of niet-herbruikbaarheid van bewijzen; of dat Gödels onvolledigheidsstelling op zichzelf cryptografie beveiligt.

Belangrijkste beperkingen: De garantie is een versoepeling van zero-knowledge; de breedste versie hangt af van meerdere aannames; de claims over één universele prover blijven deels conjectureel; en het resultaat is hoofdzakelijk fundamenteel van aard.

Hoeveel vertrouwen moet een algemene lezer hebben? Hoog dat dit, als je de definities accepteert, een belangrijk conditioneel theorieresultaat is. Gematigd dat de aannames de werkelijkheid goed

beschrijven. Laag voor onmiddellijke praktische inzet. De veilige conclusie is: het paper breekt de onmogelijkheden van zero-knowledge niet; het vindt een nieuwe bewijstheoretische route rond de delen ervan die voor veel beveiligingsspellen relevant zijn.

Bronnen

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