The Clean Paper

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

Ett kryptografiskt bevis kan dölja sin hemlighet genom att göra den saknade simulatorn svår att bevisa

Nollkunskapsbevis låter någon bevisa ett påstående utan att avslöja vittnet. Klassisk teori säger att detta, i fullständig mening, inte går att få med ett enda meddelande, utan betrodd initiering och med perfekt sundhet. Rahul Ilangos artikel får inte denna omöjlighet att försvinna. Den ändrar målet: i stället för att kräva att en simulator verkligen existerar kräver den att inget valt bevissystem effektivt kan bevisa att simulatorn inte existerar. Under stora antaganden från beviskomplexitet och kryptografi räcker det för att återskapa falsifierbara, spelbaserade konsekvenser av nollkunskap, egenskap för egenskap. Poängen är inte en internetprimitiv som kan kopplas in direkt. Det är ett bevisteoretiskt sätt att omvandla matematisk obevisbarhet till kryptografiskt skydd.

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

Tricket är inte att bevisa att hemligheten är dold

Börja med den enklaste versionen av nollkunskap.

Alice vill övertyga Bob om att ett sudokupussel har en lösning. Om hon skickar lösningen blir Bob övertygad, men pusslet är förstört. Det hon vill ha är något märkligare: ett bevis för att det finns en lösning, utan att avslöja lösningen.

Det är löftet med ett nollkunskapsbevis. Bevisaren (Alice) övertygar verifieraren (Bob) om att ett påstående är sant, samtidigt som hon inte avslöjar något utöver att påståendet är sant.

Problemet är att detta löfte kostar något. Ett vanligt matematiskt bevis har två bekväma egenskaper. Det är **ett enda meddelande**: du skriver ned det, lämnar över det och går därifrån. Och det har **perfekt sundhet**: ett falskt påstående har över huvud taget inget giltigt bevis. Klassiska omöjlighetsresultat säger att nollkunskap måste ge upp båda egenskaperna — och inte bara de två tillsammans; var och en är otillåten på egen hand.

För det första kräver ett nollkunskapsbevis en dialog. Om Alice skickar ett enda meddelande, utan någon betrodd initiering som ordnats i förväg, faller nollkunskapsgarantin samman — och det gäller oavsett hur mycket sundhet du är villig att offra i utbyte.

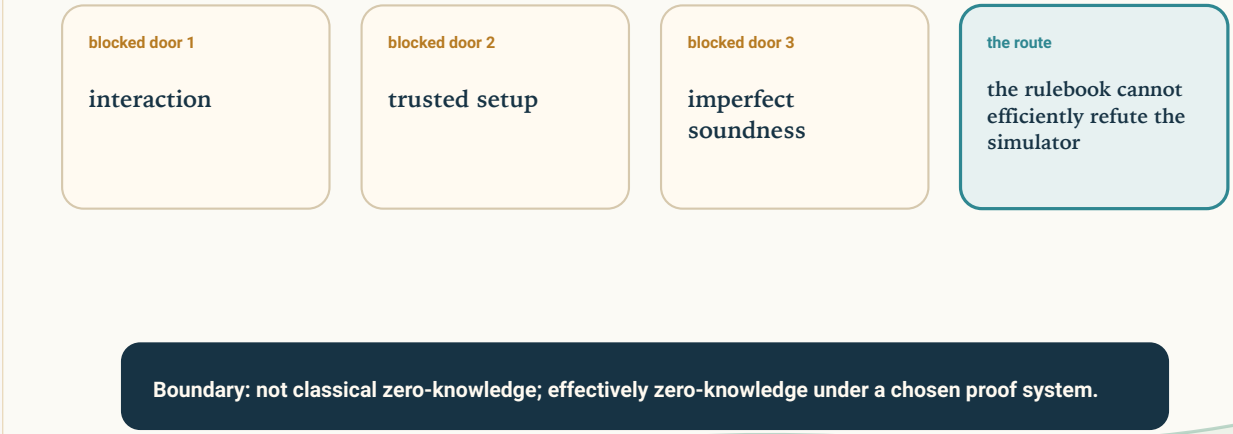
För det andra kräver ett nollkunskapsbevis en liten tolerans för fel. Det visar sig att kravet på perfekt sundhet i det tysta också förstör interaktionen: en verifierare som aldrig kan luras, oavsett vilka slumpmässiga val den gör, kan lika gärna bestämma dessa val i förväg — och när verifieraren väl är förutsägbar kan Alice besvara allt i ett enda meddelande, vilket är just det fall som redan inte fungerade.

Rahul Ilangos artikel handlar om ett sätt att ta sig runt denna dubbla mur. Inte genom att låtsas att muren inte finns och inte genom att åstadkomma klassisk nollkunskap i den omöjliga miljön. Greppet är mer subtilt: att försvaga innebörden av ”avslöjar ingenting”, men göra det på ett sätt som bevarar de säkerhetsegenskaper som kryptografer faktiskt kan testa.

Resultatet kallas **effektiv nollkunskap**.

A proof without leaking the witness

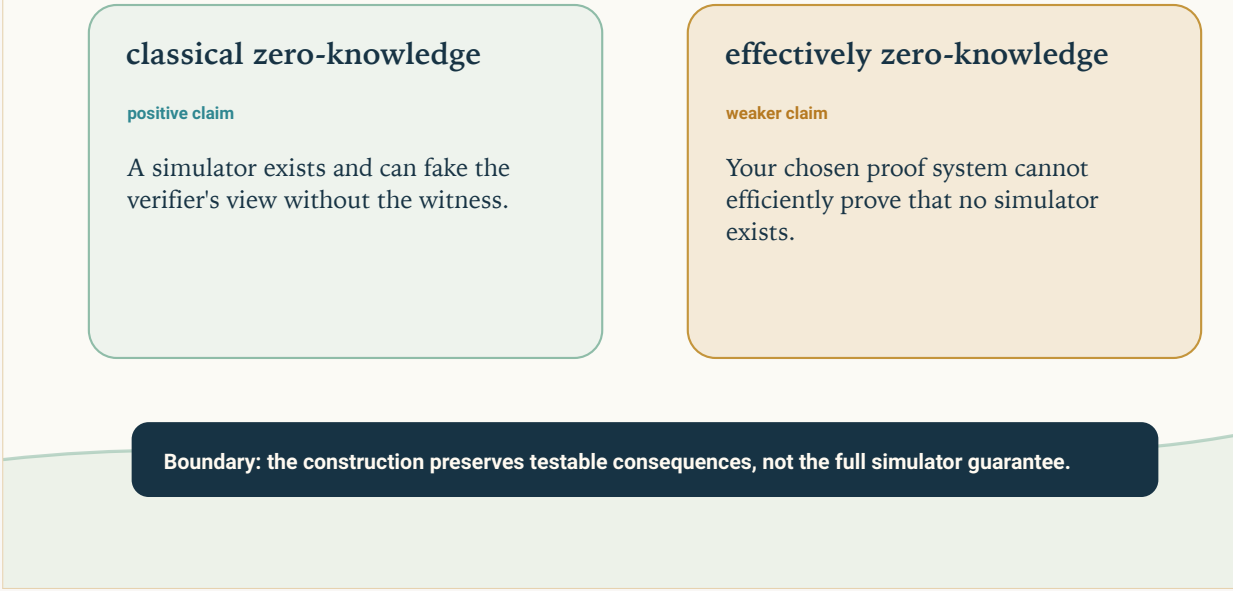
The paper keeps the ordinary-proof shape by weakening what privacy must prove.



Nollkunskap blockeras vid tre dörrar — interaktion, betrodd initiering och ofullständig sundhet. Ilangos konstruktion tar sig igenom en annan: regelverket kan inte effektivt vederlägga simulatorn.

Classical privacy vs effective privacy

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



Klassisk nollkunskap frågar om en simulator existerar; ”effektiv nollkunskap” frågar bara om det valda regelverket effektivt kan bevisa att ingen sådan existerar. Den svagare frågan är det som låter konstruktionen

behålla ett enda meddelande, ingen initiering och perfekt sundhet.

Det gamla testet: en simulator existerar

Det klassiska sättet att formalisera nollkunskap använder en fiktiv hjälpare som kallas en **simulator**.

Tanken är denna: föreställ dig Jane, som **inte** känner till Alices hemlighet. Om Jane helt på egen hand kan skapa bevis som ser precis ut som de bevis Bob skulle ha fått från Alice, då lärde Alices bevis inte Bob något nytt. Jane kunde redan efterlikna upplevelsen utan Alices hemlighet.

Klassisk nollkunskap kräver alltså en verklig simulator. Det måste finnas en effektiv algoritm som kan skapa bevis som ser äkta ut utan att känna till hemligheten — *vittnet*, på fackspråk; för sudoku är vittnet helt enkelt det lösta rutnätet.

Den definitionen är kraftfull, men det är också precis där den gamla omöjligheten biter. Så här lyder intuitionen. Ett verkligt icke-interaktivt bevis är bara en sträng. När Bob väl har strängen kan han visa den för någon annan: han har fått förmågan att bevisa påståendet för andra, vilket redan låter som mer än ”ingenting”. De klassiska satserna skärper denna intuition till omöjligheterna ovan.

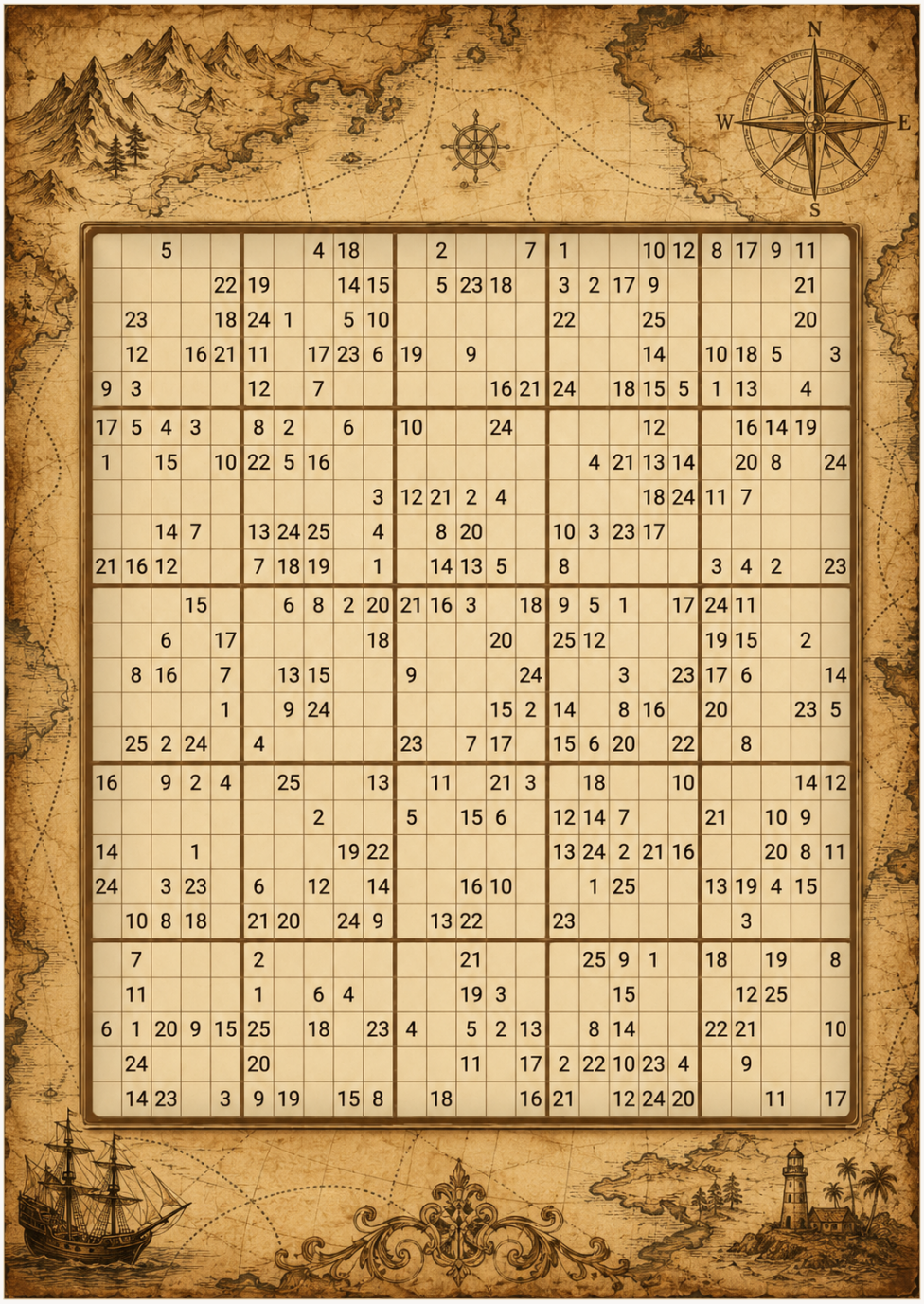
> **De tre egenskaper som denna artikel insisterar på**
>
> Artikelns titel anger tre begränsningar:
> **Ingen interaktion:** Alice skickar en enda bevissträng. Det finns inget protokoll med meddelanden fram och tillbaka.
> **Ingen initiering:** Alice och Bob förlitar sig inte på en betrodd gemensam referenssträng eller annan i förväg ordnad offentlig slumpmässighet. Många system som kallas ”icke-interaktiv nollkunskap” förlitar sig ändå på initiering; i denna artikel betyder det ingen initiering alls.
> **Perfekt sundhet:** ett falskt påstående har inget giltigt bevis. Inte ”accepteras nästan aldrig”; inget giltigt bevis existerar.
> Dessa tre egenskaper är exakt vad vanlig skriven matematik har — och som förklarats ovan kan klassisk nollkunskap inte behålla dem.

En mega-sudokuversion av skillnaden

Här är ett avsiktligt förenklat sätt att få en känsla för skillnaden.

Använd inte ett vanligt 9 gånger 9-sudoku för den seriösa delen av analogin. Det är för litet och för ändligt: en dator kan helt enkelt lösa det eller bevisa att det saknar lösning. Föreställ dig i stället en familj av **MegaSudoku(n)**-pussel. Skala upp den vanliga regeln: välj en blockstorlek n, låt $N = n^2$ och bygg ett N gånger N-rutnät indelat i n gånger n-block, med N symboler. Vanlig sudoku är

bara det lilla fallet $n = 3$, $N = 9$: ett 9 gånger 9-rutnät, 3 gånger 3-block och nio symboler. Berättelsen om beviskomplexitet börjar först när n får växa och när rutnätet kan bära extra gadgetar som får det att bete sig som en SAT-formel utklädd till ett sudokupussel. En SAT-formel är bara en lista med ja/nej-begränsningar: går det att tilldela variablerna värdena sant/falskt så att varje begränsning är uppfylld?



Ett 25x25-sudoku: dess regler kan kontrolleras utan att det färdiga rutnätet avslöjas — en visuell ersättning för ett bevis som verifierar en dold lösning, vittnet.

Sudoku och SAT: samma pussel i två förklädnader

Påståendet att en sudoku kan ”bete sig som en SAT-formel” är ingen metafor. Översättningen går åt båda hållen, och den enkla riktningen kan skrivas ut i sin helhet.

Från sudoku till SAT. SAT talar bara om sant/falskt, så ge den en boolesk variabel för varje trippel (rad, kolumn, värde): *x(r,c,v)* betyder ”cellen på rad *r*, kolumn *c* innehåller värdet *v*”. Ett 4 gånger 4-sudoku (2 gånger 2-block, värdena 1–4) behöver 4·4·4 = 64 variabler; det klassiska 9 gånger 9-spelet behöver 729. Varje sudokuregel blir sedan en samling klausuler. (En klausul är ett ELLER av variabler eller deras negationer; hela formeln är OCH-kopplingen av alla dess klausuler.)

Varje cell innehåller minst ett värde — en klausul per cell:

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

Varje cell innehåller högst ett värde — en ”inte båda”-klausul för varje värdepar:

¬x(1,1,1) ∨ ¬x(1,1,2) ¬x(1,1,1) ∨ ¬x(1,1,3) … och så vidare för alla sex paren.

Varje rad innehåller varje värde — för rad 1 och värdet 3: minst en gång,

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

och högst en gång: ¬x(1,1,3) ∨ ¬x(1,2,3), och så vidare för varje par av celler på raden.

Kolumner och block — identiska samlingar; endast cellgruppen ändras. För blocket längst upp till vänster och värdet 2:

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

plus de parvisa ”inte båda”-klausulerna.

De tryckta ledtrådarna — den enklaste delen: varje ledtråd är en klausul med en enda variabel. En tryckt 3:a i det övre vänstra hörnet blir klausulen

x(1,1,3)

OCH-kopplingen av allt detta är satisfierbar exakt när sudokun har en lösning — och en satisfierande tilldelning *är* lösningen: läs av vilka x(r,c,v) som är sanna och fyll i rutnätet. För ett 9 gånger 9-sudoku blir det 729 variabler och några tusen klausuler, vilket en modern SAT-lösare klarar av på millisekunder. Lägg märke till ledtrådsklausulen x(1,1,3): den säger ”den här cellen är exakt 3”, inte ”alla dessa celler är olika” — samma asymmetri som kommer att kräva extratricket för ledtrådsceller i protokollnoten längre ned.

Från SAT till sudoku. Artikeln behöver den motsatta, svårare riktningen: givet en *godtycklig* SAT-formel, bygg en mega-sudoku som har en lösning exakt när formeln har det. Sudokuns inbyggda regler kan bara säga ”alla dessa celler är olika”, så godtyckliga logiska begränsningar måste *byggas* — och det är precis vad gadgetarna är till för. En gadget är ett litet prefabricerat kluster av celler, ett per klausul i formeln, där särskilt utsedda celler spelar variablernas roll (symbolen de innehåller kodar sant eller falskt) och klustrets interna begränsningar är konstruerade så att dess enda tillåtna ifyllningar motsvarar tilldelningar som uppfyller

klausulen. Detta är standardhantverk från bevis för NP-fullständighet; för generaliserad sudoku genomfördes det av Yato och Seta 2003.
Tillsammans säger de två riktningarna att N gånger N-sudoku och SAT är samma problem i olika förklädnader. Det är detta som ger artikeln — och forskningsartikeln — rätt att berätta en historia om hela NP med hjälp av rutnät och symboler.

Vittnet är fortfarande lätt att föreställa sig. Alice känner till en fullständig giltig ifyllning av mega-sudokun. Bob vill bli övertygad om att en sådan ifyllning finns, men Alice vill inte avslöja den. Om hon skickar hela ifyllningen blir Bob övertygad, men hemligheten är borta.

I den klassiska nollkunskapsversionen interagerar Alice och Bob. En gammaldags tankemodell använder övertäckta brickor. Alice döljer det lösta rutnätet, byter i hemlighet namn på symbolerna före varje omgång och låter Bob granska en slumpmässigt vald lokal begränsning: en rad, en kolumn, ett block eller en gadget. Om de öppnade cellerna visar sinsemellan olika symboler stärks Bobs tilltro. Sedan täcks allt över igen och symbolerna får nya namn. (Det finns en komplikation: pusslets givna ledtrådar kräver ett extra trick, eftersom namnbytet också döljer dem. Noten nedan förklarar hur de klassiska protokollen löser detta; leksaksbilden räcker för det som följer.)

Hur de klassiska protokollen faktiskt hanterar ledtrådscellerna

Namnbytartricket har en blind fläck. Reglerna för rader, kolumner och block säger alla ”dessa celler är sinsemellan olika”, och *sinsemellan olika* överlever varje namnbyte av symbolerna. Men en ledtråd säger ”den här cellen innehåller exakt 5”, och efter namnbytet ser Bob bara $\sigma(5)$ — någon maskerad symbol — utan att känna till namnbytet σ. Han kan inte kontrollera någonting. Om detta inte åtgärdades skulle Alice kunna bevisa att *något* giltigt rutnät existerar och samtidigt helt bortse från de tryckta ledtrådarna, vilket inte bevisar något om *det här* pusslet. Den klassiska litteraturen har två standardlösningar.
Paletten. Lägg till en extra rad med N celler i det dolda rutnätet — en palett som Alice fyller med symbolerna 1…N i en fast offentlig ordning och sedan byter namn på tillsammans med allt annat, så att den innehåller $\sigma(1)$…σ(N). Bobs slumpmässiga utmaning får nu ytterligare ett alternativ. Förutom att välja en rad, kolumn, ett block eller en gadget att öppna kan han välja **paletten plus en ledtrådscell**. Alice avtäcker båda; paletten avslöjar omgångens namnbyte, och Bob kontrollerar att ledtrådscellen visar exakt den namnbytta versionen av den tryckta ledtråden. Detta förblir nollkunskap eftersom Bob bara får veta σ — som dras på nytt i varje omgång och är värdelös på egen hand — samt värdet i en cell som han redan kände till från pusslet. Ingenting om de hemliga cellerna läcker, och en simulator kan efterlikna vyn genom att dra ett slumpmässigt σ. Protokollet är sunt eftersom en fuskande Alice avslöjas med en fast sannolikhet per omgång, och omgångarna upprepas tills tvivlet är försumbart.
Kompilera bort ledtrådarna. En mer strukturell variant tar bort den särskilda utmaningen i stället för att lägga till den. I stället för att *verifiera* ledtrådsvärdet tvingar man fram det med olikhetsbegränsningar: koppla ledtrådscellen till varje palettcell utom den som bär dess eget värde — ”skild från $\sigma(1)$, skild från $\sigma(2)$, …, skild från allt utom $\sigma(5)$”. Den enda symbol som cellen lagligen kan innehålla är ledtrådens. Varje begränsning är nu återigen av typen ”dessa

två är olika” — invariant under namnbyte och kontrollerbar precis som en rad. Det är samma manöver som används för förfärgade hörn i det klassiska protokollet för graffärgning, och den fångar andan i ordet *gadgetar* ovan: i bilden där MegaSudoku är SAT kompileras ledtrådarna till olikhetsgadgetar precis som alla andra begränsningar.
Det fysiska protokollet. Det verkliga kortprotokollet för sudoku (Gradwohl, Naor, Pinkas och Rothblum, 2007) använder inga namnbyten alls och hanterar ledtrådarna innan döljandet ens börjar. För varje cell lägger Alice ut tre identiska kort med cellens värde — med framsidan nedåt för hemliga celler, men **med framsidan uppåt för ledtrådsceller**, så att Bob med egna ögon ser att ledtrådarna respekteras innan korten vänds. Sedan hamnar ett kort från varje cell i radens bunt, ett i kolumnens och ett i blockets; varje bunt blandas och visas, och Bob kontrollerar att den innehåller alla N symboler. Blandningen förstör positionsinformationen (det är nollkunskapen), men ledtrådarna hade redan slagits fast när korten lades ut.
Oavsett metod är lärdomen densamma som artikeln hela tiden återkommer till: ett nollkunskapsprotokoll är en noggrann bokföring av *vilka* fakta som överlever döljandet. Namnbytet bevarar ”sinsemellan olika” och suddar ut ”lika med 5” — därför måste ”lika med 5” smugglas tillbaka på andra sätt.

Detta är inte protokollet i forskningsartikeln. Det är tankemodellen för klassisk nollkunskap:

- Alice och Bob skickar meddelanden fram och tillbaka.
- Bob väljer slumpmässiga kontroller.
- Alice avslöjar bara lokal konsistens, inte hela lösningen.
- Integritetsbeviset fungerar genom att visa att Bobs vy kunde ha skapats utan Alices hemliga lösning.

Klassisk nollkunskap är alltså uppbyggd kring ett positivt faktum:

> En simulator existerar verkligen.

Ta nu bort de bekväma delarna. Alice skickar en enda bevissträng och går därifrån. Det finns ingen betrodd initiering, ingen gemensam slumpsträng som förberetts i förväg, och Bob får aldrig acceptera ett falskt pussel. Det är den miljö där klassisk nollkunskap inte kan överleva.

Ytterligare en gestalt behövs före tricket. Fixera ett **regelverk**: ett formellt bevissystem, i logikerns mening — en fast uppsättning axiom plus mekaniska regler för att kontrollera skrivna matematiska bevis. ZFC, matematikens standardaxiom, är det kanoniska exemplet. Allt från och med nu anges relativt ett regelverk som valts i förväg, och valet är flexibelt: konstruktionen fungerar för vilket regelverk du än fixerar, inklusive ZFC.

(En anmärkning om ord, lånad från forskningsartikeln själv: ”bevissystem” betyder här alltid detta regelverk — det formella system som kontrollerar matematiska bevis — aldrig de meddelanden Alice skickar. Alices och Bobs maskineri kallas ”bevisaren och verifieraren”.)

Versionen i Gödels anda behåller berättelsen om mega-sudoku men ändrar beviset.

Välj ett andra begränsningssystem av samma visade storlek och kalla det **D**. I berättelsen är S och D två MegaSudoku(n)-pussel i samma format. Bakom kulisserna kan D ha börjat som en svår logisk formel av en annan storlek; vid behov kan den fyllas ut med ofarliga dummybegränsningar så att den passar i samma rutnät. D byggs från en logisk formel som faktiskt är **otillfredsställbar**: det finns ingen möjlig värdetilldelning som gör alla dess begränsningar sanna, precis som ett trasigt pussel inte har något tillåtet färdigifyllt rutnät. Ett enkelt exempel vore en formel som kräver både ”X är sant” och ”X är falskt”. D har alltså ingen giltig ifyllning.

Men D får inte vara ett trasigt pussel som är *lätt att avslöja*. Det enkla exemplet ovan misslyckas med detta: vilket regelverk som helst vederlägger ”X och icke-X” på en rad. D måste vara falskt på ett sätt som det valda regelverket inte kan intyga med ett kort resonemang. Om regelverket kunde vederlägga D med ett kort bevis skulle berättelsen nedan falla samman: den alternativa väg som kunde ha skapat bevis utan Alices hemlighet skulle formellt kunna uteslutas, och integritetsgarantin skulle försvinna med den. Därför väljs D ur en familj som det fixerade regelverket inte effektivt kan vederlägga: i det regelverket finns inget kort bevis för att D saknar lösning.

Alices bevis i ett enda meddelande gäller sedan ett antingen/eller-påstående:

> antingen har den verkliga mega-sudokun S en lösning, eller så har lockbetet D en lösning.

Detta är den logiska länken. D skapas **inte** på något magiskt sätt som gör S sant. Beviset argumenterar inte ”D har ingen lösning, alltså har S en lösning”. Det bevisar disjunktionen **S eller D**. Perfekt sundhet säger att en falsk disjunktion inte kan ha ett giltigt bevis. Eftersom D är falskt i verkligheten — det saknar lösning — kan disjunktionen bara vara sann om S är sant. Om beviset godtas måste S alltså ha en lösning. Lockbetet kan inte göra ett falskt S sant.

Men för den nollkunskapsliknande delen: fråga vad som skulle hända om D *faktiskt* hade en lösning. Den lockbeteslösningen skulle fungera som ett alternativt vittne. Den skulle låta någon skapa bevis utan att känna till Alices verkliga lösning på mega-sudokun — med andra ord en simulator. I verkligheten har D ingen lösning, så denna väg till en simulator är stängd. Poängen är att regelverket inte effektivt kan bevisa att den är stängd.

D har alltså två uppgifter. För **sundheten** är D falskt, så ett giltigt bevis för ”S eller D” tvingar fram S. För **effektiv nollkunskap** är D svårt att vederlägga, så regelverket kan inte snabbt utesluta lockbetesvägen som skulle ha gjort simulering möjlig.

Säkerhetstestet är alltså inte längre:

> Kan vi bevisa att en simulator verkligen existerar?

Det blir:

> Kan ditt regelverk effektivt bevisa att simulatorn är omöjlig?

Om svaret är nej följer något förvånansvärt starkt: varje säkerhetsgaranti som (a) kan observeras genom att köra ett test och (b) bevisligen följer — inom detta regelverk — av att en simulator existerar,

gäller faktiskt. En framgångsrik attack mot någon av dem skulle i sig utgöra det saknade korta vederläggningsbeviset, och det saknade korta vederläggningsbeviset existerar inte. Det är den ”effektiva” delen av effektiv nollkunskap.

Kontrasten i klassrummet blir alltså:

Klassisk nollkunskap: bevisen är säkra eftersom en simulator existerar.

Effektiv nollkunskap i Gödels anda: bevisen behandlas som säkra för observerbara säkerhetstester eftersom regelverket inte effektivt kan bevisa att simulatorn är omöjlig.

Det andra påståendet är svagare. Det är också därför artikeln kan behålla de tre egenskaper som fick den klassiska versionen att fallera: ett enda meddelande, ingen initiering och perfekt sundhet.

Det nya testet: du kan inte bevisa att simulatorn saknas

Ilangos uppluckring ändrar frågan.

Klassisk nollkunskap frågar:

> Existerar en simulator?

Effektiv nollkunskap ställer en svagare fråga:

> Kan ditt valda regelverk effektivt bevisa att ingen simulator existerar?

Det kan låta som en teknisk undanmanöver, men det är kärnidén. Konstruktionen befinner sig i ett märkligt tillstånd: någon simulator existerar faktiskt inte — forskningsartikeln säger det uttryckligen — men regelverket du fixerade kan inte effektivt bevisa att den inte gör det. Om varje dålig konsekvens som du bryr dig om skulle kräva en sådan vederläggning beter sig systemet ändå som nollkunskap med avseende på dessa konsekvenser.

Det är här Gödel kommer in. Inte som dekoration och inte som att ”Gödel gör krypto säkert”. Kopplingen är bevisteoretisk. Ett regelverk kallas **optimalt** om det i en precis mening är det bästa möjliga: när något regelverk kan vederlägga en formel av relevant slag med ett kort bevis kan det optimala regelverket också göra det, med ett bevis som är högst polynomiellt längre. Krajíček och Pudlák antog 1989 att **inget optimalt bevissystem existerar**: vilket regelverk du än fixerar bevisar något annat regelverk någon familj av sanna påståenden mycket mer koncist. Detta är en av beviskomplexitetens centrala öppna förmodanden och den ändliga, komplexitetsteoretiska kusinen till Gödels ofullständighetssats: vissa sanna påståenden har inget kort bevis i det regelverk du fixerade — inte för att de i princip är obevisbara, utan för att varje fixerat regelverk lämnar vissa korta sanningar utan korta bevis.

Forskningsartikeln antar detta förmodande (i en något starkare ”oändligt ofta”-form, vilket är standard när förmodanden används kryptografiskt). Vinsten, genom en sats av Krajíček och Pudlák, är

konkret: för varje regelverk finns en följd av formler som verkligen är otillfredsställbara och som regelverket inte kan vederlägga med korta bevis — och som, avgörande nog, en effektiv algoritm kan *generera*. Den sista egenskapen, uniformitet, är det som förvandlar hela idén från ett existenspåstående till en verklig algoritm som Alice kan köra: hennes lockbeten D kommer från ett löpande band, inte ur tomma intet.

Det kryptografiska greppet är att sätta denna brist på beviskraft i arbete.

Vad konstruktionen gör

Här är konstruktionen i forskningsartikeln, reducerad till sin form.

Fixera ett regelverk — exempelvis ZFC. Under antagandet från beviskomplexitet finns en effektivt genererbar följd av formler som faktiskt är otillfredsställbara, men regelverket har inget kort bevis för att de är otillfredsställbara.

Bygg nu ett bevis i ett enda meddelande av denna form:

> antingen är det verkliga påståendet satisfierbart, eller så är denna särskilda svåra formel satisfierbar.

Den särskilda svåra formeln är inte satisfierbar. Om det underliggande bevismaskineriet har perfekt sundhet innebär ett godtagande av meddelandet därför fortfarande att det verkliga påståendet är sant. Det ger perfekt sundhet.

Men för den nollkunskapsliknande säkerheten, föreställ dig att den särskilda svåra formeln *vore* satisfierbar. Då skulle dess vittne kunna användas för att simulera bevis utan kännedom om det verkliga vittnet. Formeln är inte satisfierbar i verkligheten — men regelverket kan inte effektivt bevisa det. Därför kan det inte effektivt bevisa att simulatorn är omöjlig.

Det är gångjärnet. Systemet döljer inte hemligheten genom att frambringa en klassisk simulator. För en stor klass av observerbara säkerhetstester döljer det hemligheten bakom regelverkets oförmåga att intyga att simulatorn saknas.

Vad forskningsartikeln hävdar

Huvudsatsen kommer i lager. Kärnresultatet är detta:

Under ett kryptografiskt standardantagande — existensen av **icke-interaktiva vittnesoskiljbara bevis**, välstuderade objekt som följer av flera etablerade antagandepaket — och under det beviskomplexitetsteoretiska förmodandet att **inget (oändligt ofta) optimalt bevissystem existerar**, konstruerar artikeln, för varje val av regelverk, en bevisare och verifierare med ett enda meddelande för NP/SAT med **perfekt sundhet** och utan initiering som är effektivt nollkunskap relativt detta

regelverk. (NP/SAT är den vanliga ”svåraste gemensamma nämnaren” för pusselliknande problem; mega-sudoku är en av dess förklädnader.)

För det bredare påståendet om att bevara falsifierbara säkerhetsegenskaper lägger artikeln till ytterligare ett standardantagande, avrandomiseringsövertygelsen **P = BPP** (grovt uttryckt: slumpmässighet ger inte algoritmer någon väsentlig extra kraft).

Översatt från satsspråk:

- Beviset är ett enda meddelande.
- Det finns ingen betrodd initiering.
- Falska påståenden kan inte bevisas.
- Bevisaren är inte klassiskt nollkunskap — den har ingen simulator.
- Men varje falsifierbar, spelbaserad säkerhetskonsekvens av klassisk nollkunskap kan uppnås i denna miljö.

”Falsifierbar” spelar roll. Det betyder att ett säkerhetsfel kan testas genom att köra en motståndare i ett spel. Många kryptografiska säkerhetsdefinitioner har denna form: kan motståndaren skilja mellan två krypteringar, invertera en funktion, återskapa ett vittne eller vinna något angivet experiment? Satsen ger en bevisare för varje falsifierbar egenskap, en i taget. En enda bevisare som har *varje* falsifierbar egenskap samtidigt är sannolikt omöjlig — den gamla attacken genom återanvändning (”Bob kan visa beviset för andra”) är i sig en falsifierbar egenskap, och den fallerar verkligen här. Artikelns förslag är att en enda bevisare rimligen kan täcka alla *naturliga* falsifierbara egenskaper — de som faktiskt förekommer i kryptografisk praktik — men den delen är en villkorlig sats som vilar på ett informellt begrepp om ”naturlig”, plus ett uttryckligt förmodande. Garantin riktar in sig på observerbara fel, inte på varje filosofisk eller simuleringsbaserad innebörd av hemlighållande.

En konkret följdsats är värd att nämna: konstruktionen ger de första icke-interaktiva *vittnesdöljande* bevisen med en uniform bevisare — ”ett bevis för ett pussel hjälper dig inte att hitta dess lösning”, utan interaktion och utan initiering — ett blygsamt klingande objekt som hade trotsat konstruktion i årtionden.

Vad detta inte säger

Det här är avsnittet som håller texten hederlig.

Det säger **inte** att de gamla omöjlighetssatserna hade fel. Konstruktionen undviker dem genom att ändra definitionen.

Det ger **inte** vanlig, klassisk nollkunskap utan interaktion, utan initiering och med perfekt sundhet. Forskningsartikeln säger uttryckligen att den konstruerade bevisaren inte har någon simulator.

Det betyder **inte** att beviset inte kan återanvändas. Ett bevis i ett enda meddelande kan fortfarande visas för någon annan; artikeln bevarar inte egenskaper av förnekbarhetstyp. (Icke-interaktiv nollkunskap med betrodd initiering har samma begränsning.)

Det betyder **inte** att detta är ett praktiskt protokoll redo för driftsättning. Det här är komplexitetsteori och kryptografiska grunder. Resultatet beror på stora antaganden från beviskomplexitet och kryptografi, och konstruktionen handlar om vad som är möjligt i princip.

Det gör **inte** ”Gödel” till en magisk säkerhetsprimitiv. Gödelkopplingen går genom bevissystem, optimala bevissystem och ändliga analoger till ofullständighet. Den användbara intuitionen är inte ”ofullständighet skyddar ditt lösenord”. Den är: om ett regelverk inte effektivt kan bevisa att en simulator är omöjlig, kan attacker som skulle kräva ett sådant bevis blockeras på säkerhetsdefinitionernas nivå.

Varför det ändå är intressant

Kryptografi omvandlar ofta svårighet till säkerhet. Faktorisering är svårt, så antaganden av RSA-typ blir användbara. Gitterproblem är svåra, så gitterkryptografi blir användbar. Här är svårigheten märkligare: inte ”svårt att beräkna en hemlighet”, utan ”svårt att bevisa att ett visst bevisobjekt inte kan existera”.

Det är därför forskningsartikeln känns ovanlig. Den behandlar axiom och regelverk nästan som kryptografiska resurser. Den vanliga omöjligheten säger att det finns en spänning mellan sundhet och simulering. Ilangos grepp är att placera spänningen bakom en bevisteoretisk ridå: simulatorn saknas, men det formella systemet kan inte effektivt blottlägga denna frånvaro.

För en läsare är det överraskande inte att detta kommer att ersätta dagens nollkunskapssystem. Det gör det förmodligen inte, åtminstone inte direkt. Det överraskande är att en begränsning från matematisk logik kan användas konstruktivt: inte bara som en mur, utan som ett slags skydd.

Hur starka är beläggen?

Detta är en artikel som presenterar en sats, så ”belägg” betyder något annat än i en artikel om biologi eller astronomi. Frågan är inte om ett experiment har replikerats. Frågan är om definitionerna, antagandena och beviskedjan stöder påståendet.

Beviset är formellt, och artikeln redovisar sina antaganden uttryckligen. Antagandena är inte lättvindiga. Icke-interaktiva vittnesoskiljbara bevis är standardobjekt inom kryptografin och följer av flera etablerade antagandepaket. Förmodandet att inget optimalt bevissystem existerar är ett centralt förmodande inom beviskomplexitet. P = BPP är en standardövertygelse om avrandomisering som bara används för den bredare satsen om falsifierbara egenskaper.

Artikeln argumenterar också för att antagandena är rätt pris, inte en godtycklig ställning: den bevisar en omvänd sats som visar att de i allt väsentligt är nödvändiga — om sådana konstruktioner alls existerar måste icke-interaktiva vittnesoskiljbara bevis existera, och (givet vanliga envägsfunktioner) kan inget optimalt bevissystem existera. Antagandena är dessutom ”vinn-vinn”: att vederlägga något av dem skulle i sig vara en banbrytande upptäckt inom beviskomplexitet, kryptografi eller komplexitetsteori.

Men eftersom resultatet är villkorligt är tilltron till det också villkorlig. Om dessa antaganden faller förändras tolkningen av satsen. Och även om antagandena håller är garantin inte fullständig klassisk nollkunskap; det är artikelns försvagade, bevisteoretiska version.

Rimlig tilltro är därför hög till att artikeln fastställer ett sammanhängande villkorligt möjlighetsresultat, måttlig till att dess antaganden beskriver den kryptografiska värld vi faktiskt lever i och låg när det gäller omedelbara praktiska konsekvenser.

Varför det spelar roll

Artikeln öppnar en väg som skulle vara stängd.

Klassisk teori säger: fullständig nollkunskap kan inte vara ett enda meddelande utan initiering och kan inte ha perfekt sundhet. Ilangos artikel säger: om vi frågar efter de konsekvenser av nollkunskap som kan testas i säkerhetsspel, och om vi låter säkerhetsdefinitionen bero på vad ett regelverk kan eller inte effektivt kan vederlägga, då kan mycket av det användbara beteendet återskapas — med ett enda meddelande, ingen initiering och perfekt sundhet.

Det är inte en liten definitionsjustering. Det är ett annat sätt att tänka på kryptografiska garantier. Fråga inte bara vad som existerar, utan vad ditt regelverk kan utesluta. Behandla inte obevisbarhet som en filosofisk olägenhet, utan använd den som struktur.

Den praktiska världen kanske inte förändras i morgon. Men det gör den begreppsliga kartan. Det finns nu en formell mening i vilken ”ingen kan effektivt bevisa att hemligheten läckte” kan vara starkt nog för att återskapa många av de spelbaserade skydd vi ville ha från ”hemligheten läckte inte”.

Det är därför Gödel hör hemma i titeln.

Ren sammanfattning

Nollkunskapsbevis låter en bevisare övertyga en verifierare om att ett påstående är sant utan att avslöja vittnet. Klassiska omöjlighetsresultat säger att nollkunskap inte kan pressas ihop till ett enda meddelande utan initiering och inte kan ha perfekt sundhet. Rahul Ilangos artikel vederlägger inte dessa omöjligheter. Den definierar ett svagare begrepp, effektiv nollkunskap: i stället för att kräva att en simulator verkligen existerar kräver det att ett valt bevissystem — ett formellt regelverk som ZFC — inte effektivt kan bevisa att ingen simulator existerar. Under stora antaganden från kryptografi (icke-interaktiva vittnesoskiljbara bevis) och beviskomplexitet (inget optimalt bevissystem existerar) konstruerar artikeln bevisare med ett enda meddelande för NP/SAT, utan initiering och med perfekt sundhet, som uppnår de falsifierbara, spelbaserade konsekvenserna av nollkunskap egenskap för egenskap. En enda bevisare som täcker alla ”naturliga” sådana egenskaper är en ytterligare, delvis förmodandebaserad utvidgning — och att täcka bokstavligen varje falsifierbar egenskap är sannolikt omöjligt, eftersom bevis fortfarande kan återanvändas. Resultatet är teoretiskt och villkorligt, inte en driftsatt primitiv, men det visar ett nytt sätt att använda bevisteoretisk obevisbarhet som en kryp-

tografisk resurs.

Utan omsvep

Vad artikeln visar: Under angivna antaganden kan man bygga bevisare för NP/SAT med ett enda meddelande, utan initiering och med perfekt sundhet, som är effektivt nollkunskap relativt vilket valt bevissystem som helst och som uppnår varje falsifierbar, spelbaserad konsekvens av klassisk nollkunskap.

Vad som är rimligt men inte ovillkorligt bevisat: Att de nödvändiga antagandena från beviskomplexitet och kryptografi håller. De är seriösa, välstuderade antaganden — och artikeln visar att de i allt väsentligt är både nödvändiga och tillräckliga — men de är fortfarande antaganden.

Vad den inte visar: Klassisk nollkunskap utan interaktion, utan initiering och med perfekt sundhet; ett praktiskt system redo för driftsättning; förnekbarhet eller att bevis inte kan återanvändas; eller att Gödels ofullständighetssats i sig gör kryptografi säker.

Huvudsakliga begränsningar: Garantin är en uppluckring av nollkunskap; den bredaste versionen beror på flera antaganden; påståendena om en enda universell bevisare är fortfarande delvis förmodandebaserade; och resultatet är i första hand grundläggande teori.

Hur stor tilltro bör en allmän läsare ha? Hög till att detta är ett viktigt villkorligt teoriresultat om definitionerna godtas. Måttlig till att antagandena fångar verkligheten. Låg när det gäller omedelbar praktisk driftsättning. Den säkra slutsatsen är: artikeln bryter inte mot omöjlighetsresultaten för nollkunskap; den hittar en ny bevisteoretisk väg runt de delar av dem som spelar roll för många säkerhetsspel.

Källor

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