The Clean Paper

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

Kryptografinen todistus voi peittää salaisuutensa tekemällä simulaattorin puuttumisen vaikeaksi todistaa

*Nollatietotodistuksissa väitteen voi todistaa paljastamatta sen todistajaa (*witness*). Klassinen teoria sanoo, ettei tätä voida täydessä merkityksessä yhdistää yhteen viestiin, luotetun alkuasetelman puuttumiseen ja täydelliseen luotettavuuteen (*perfect soundness*). Rahul Ilangon tutkimus ei poista tätä mahdottomuutta. Se muuttaa tavoitetta: sen sijaan että vaadittaisiin simulaattorin todella olevan olemassa, valittu todistusjärjestelmä ei saa pystyä tehokkaasti todistamaan, ettei simulaattoria ole. Merkittävien todistuskompleksisuuteen ja kryptografiaan liittyvien oletusten vallitessa tämä riittää palauttamaan nollatiedon falsifioitavissa olevia, pelipohjaisia seurauksia ominaisuus kerrallaan. Kyse ei ole suoraan käyttöön otettavasta internet-primitiivistä vaan todistusteoreettisesta tavasta käyttää matemaattista todistamattomuutta kryptografisena suojana.*

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

Temppu ei ole todistaa, että salaisuus pysyy piilossa

Aloitetaan nollatiedon yksinkertaisimmasta versiosta.

Alice haluaa vakuuttaa Bobin siitä, että Sudoku-tehtävällä on ratkaisu. Jos hän lähettää ratkaisun, Bob vakuuttuu, mutta tehtävä on pilalla. Alice haluaa jotain oudompaa: todistuksen siitä, että ratkaisu on olemassa, paljastamatta itse ratkaisua.

Tämän nollatietotodistus lupaa. Todistuksen esittäjä (Alice) vakuuttaa tarkistajan (Bob) väitteen paikkansapitävyydestä paljastamatta mitään muuta kuin sen, että väite on tosi.

Ongelma on, että tämä lupaus maksaa jotakin. Tavallisella matemaattisella todistuksella on kaksi mukavaa ominaisuutta. Se on **yksi viesti**: kirjoitat sen, annat sen toiselle ja poistut. Ja sillä on **täydellinen luotettavuus (perfect soundness)**: väärällä väitteellä ei ole lainkaan pätevää todistusta. Klassiset mahdottomuustulokset sanovat, että nollatiedon on luovuttava kummastakin ominaisuudesta — eikä vain niiden yhdistelmästä; kumpikin on jo yksin ongelmallinen.

Ensinnäkin nollatietotodistus tarvitsee vuorovaikutusta. Jos Alice lähettää yhden ainoan viestin eikä ennalta ole järjestetty luotettua alkuasetelmaa, nollatietotakuu romahtaa — riippumatta siitä, kuinka paljon luotettavuudesta olisit valmis tinkimään vastineeksi.

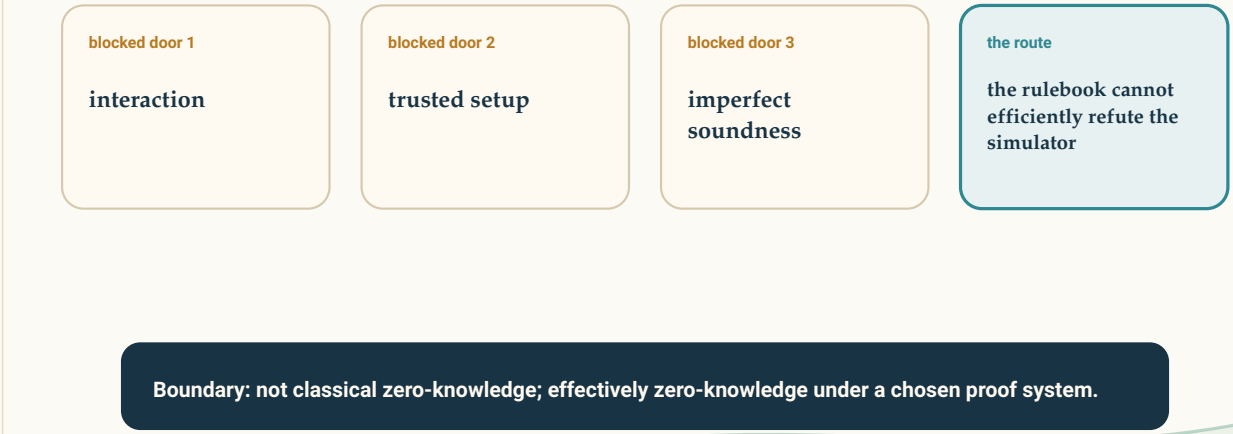
Toiseksi nollatietotodistus tarvitsee pienen virhemarginaalin. Täydellisen luotettavuuden vaatiminen tuhoaa huomaamatta myös vuorovaikutuksen: tarkistaja, jota ei voida koskaan huijata riippumatta sen satunnaisista valinnoista, voisi yhtä hyvin lukita nuo valinnat etukäteen. Kun tarkistajan toiminta on ennustettavaa, Alice voi vastata kaikkeen yhdellä viestillä — juuri sillä tavalla, jonka jo tiedämme rikkovan klassisen nollatiedon.

Rahul Ilangon tutkimus etsii kiertotien tämän kaksinkertaisen seinän ohi. Ei teeskentelemällä, ettei seinää ole, eikä tuottamalla klassista nollatietoa tilanteessa, jossa se on mahdotonta. Liike on hienovaraisempi: heikennetään sitä, mitä ”ei paljasta mitään” tarkoittaa, mutta tavalla, joka säilyttää ne turvallisuusominaisuudet, joita kryptografit voivat todella testata.

Tulosta kutsutaan **effectively zero-knowledge** -ominaisuudeksi eli tässä **tehokkaaksi nollatiedoksi**.

A proof without leaking the witness

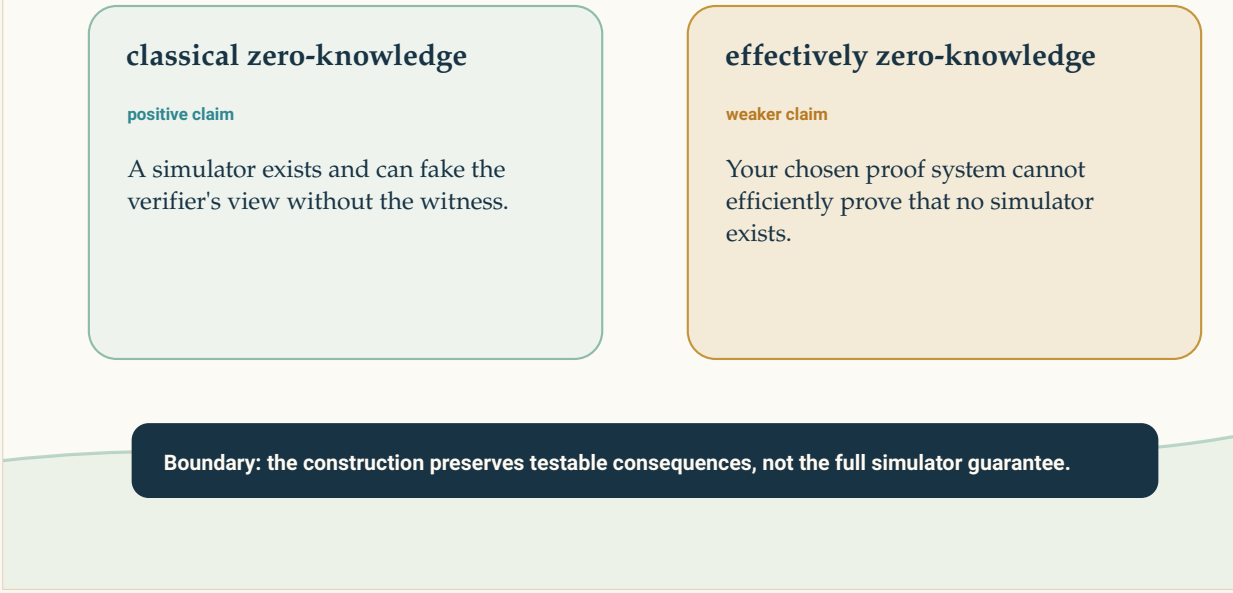
The paper keeps the ordinary-proof shape by weakening what privacy must prove.



Nollatiedon tiellä on kolme suljettua ovea — vuorovaikutus, luotettu alkuasetus ja epätäydellinen luotettavuus. Ilangon konstruktio kulkee toisesta ovesta: sääntökirja ei pysty tehokkaasti kumoamaan simulaattorin mahdollisuutta.

Classical privacy vs effective privacy

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



Klassinen nollatieto kysyy, onko simulaattori olemassa; ”effectively zero-knowledge” kysyy vain, pystyykö

valittu sääntökirja tehokkaasti todistamaan, ettei sitä voi olla. Juuri tämä heikompi kysymys antaa konstruktiolle mahdollisuuden säilyttää yhden viestin, alkuasetelman puuttumisen ja täydellisen luotettavuuden.

Vanha testi: simulaattori on olemassa

Klassinen tapa määritellä nollatieto käyttää kuvitteellista apuria, jota kutsutaan **simulaattoriksi**.

Ajatus on seuraava. Kuvittele Jane, joka **ei** tunne Alicen salaisuutta. Jos Jane pystyy täysin omin voimin tuottamaan todistuksia, jotka näyttävät aivan samalta kuin Alicelta Bobille tulleet todistukset, Alicen todistukset eivät opettaneet Bobille mitään uutta. Jane olisi voinut jäljitellä koko kokemuksen jo ilman Alicen salaisuutta.

Siksi klassinen nollatieto vaatii oikeasti olemassa olevan simulaattorin. On oltava tehokas algoritmi, joka pystyy tuottamaan aidon näköisiä todistuksia tuntematta salaisuutta — alan jargonissa *todistajaa (witness)*; Sudokussa tämä todistaja on yksinkertaisesti valmis ratkaistu ruudukko.

Määritelmä on vahva, mutta juuri siihen vanha mahdottomuustulos puree. Intuitio on tämä: aidosti ei-vuorovaikutteinen todistus on vain merkkijono. Kun Bob on saanut merkkijonon, hän voi näyttää sen jollekulle toiselle. Hän on siis saanut kyvyn todistaa väite muille, mikä kuulostaa jo enemmältä kuin ”ei mitään”. Klassiset lauseet täsmentävät tämän intuition edellä kuvatuiksi mahdottomuuksiksi.

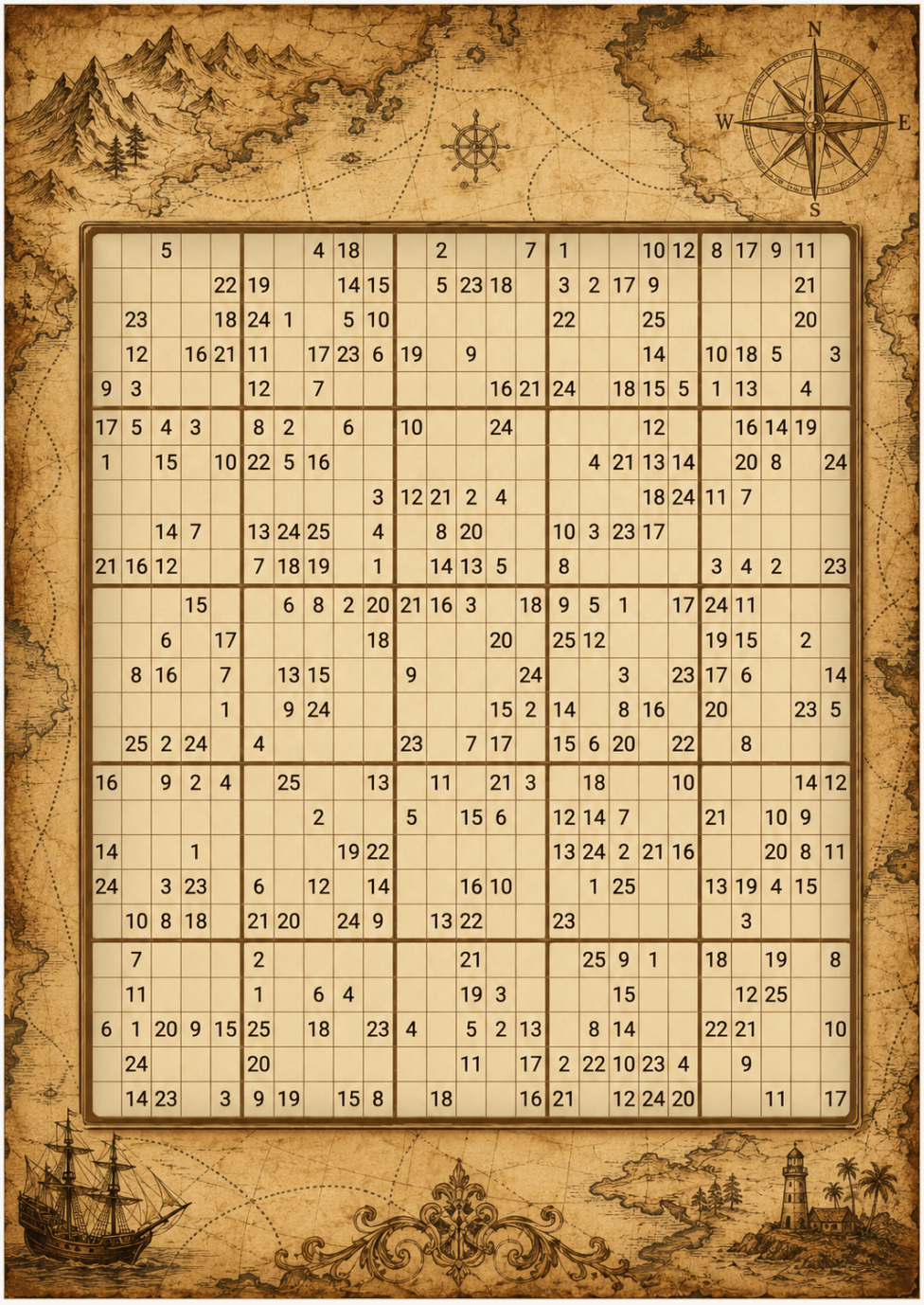
> **Kolme ominaisuutta, joista tutkimus pitää kiinni**
>
> Tutkimuksen otsikko nimeää kolme rajoitetta:
> **Ei vuorovaikutusta:** Alice lähettää yhden todistusmerkkijonon. Edestakaista protokollaa ei ole.
> **Ei alkuasetelmaa:** Alice ja Bob eivät nojaa luotettuun yhteiseen viitemerkkijonoon tai muuhun etukäteen sovittuun julkiseen satunnaisuuteen. Monet ”ei-vuorovaikutteisiksi nollatietojärjestelmiksi” kutsutut ratkaisut tarvitsevat silti alkuasetelman; tässä tutkimuksessa sitä ei ole lainkaan.
> **Täydellinen luotettavuus:** väärällä väitteellä ei ole pätevää todistusta. Ei ”hyväksytään vain äärimmäisen harvoin”, vaan pätevää todistusta ei ole olemassa.
> Tavallisella kirjoitetulla matematiikalla on juuri nämä kolme ominaisuutta — ja kuten edellä selitettiin, klassinen nollatieto ei voi säilyttää niitä.

Ero MegaSudokun avulla

Tässä on tarkoituksella yksinkertaistettu tapa hahmottaa ero.

Vakavaan analogiaan ei kannata käyttää tavallista 9×9 -Sudokua. Se on liian pieni ja rajallinen: tietokone voi yksinkertaisesti ratkaista sen tai todistaa, ettei ratkaisua ole. Kuvittele sen sijaan **MegaSu-**

doku(n)-tehtävien perhe. Skaalaa tavallinen sääntö: valitse lohkokooksi n, aseta $N = n^2$ ja rakenna N kertaa N -ruudukko, joka jakautuu n kertaa n -lohkoihin ja käyttää N:ää symbolia. Tavallinen Sudoku on vain pieni tapaus $n = 3$, $N = 9$: 9×9 -ruudukko, 3×3 -lohkot ja yhdeksän symbolia. Todistuskompleksisuuden tarina alkaa vasta, kun n saa kasvaa ja ruudukkoon voidaan lisätä apukonstruktioita, joiden avulla se käyttäytyy kuin Sudoku-asuun puettu SAT-kaava. SAT-kaava on vain luettelo kyllä/ei-rajoitteista: voidaanko muuttujille antaa tosi/epätosi-arvot niin, että kaikki rajoitteet täyttyvät?



25x25-Sudoku: sen sääntöjen täyttyminen voidaan tarkistaa paljastamatta valmista ruudukkoa — visuaalinen vastine todistukselle, joka vahvistaa piilotetun ratkaisun eli todistajan.

Sudoku ja SAT: sama pulma kahdessa asussa

Väite, että Sudoku voi ”käyttäytyä kuin SAT-kaava”, ei ole vertauskuva. Muunnos toimii kumpaankin suuntaan, ja helpompi suunta voidaan kirjoittaa kokonaan auki.

Sudokusta SAT:iin. SAT tuntee vain tosi/epätosi-arvot, joten annetaan sille yksi totuusarvo-muuttuja jokaista (rivi, sarake, arvo) -kolmikkoa kohti: *x(r,c,v)* tarkoittaa ”rivin *r*, sarakkeen *c* solussa on arvo *v*”. 4×4 -Sudoku (2×2 -lohkot, arvot 1–4) tarvitsee $4 \cdot 4 \cdot 4 = 64$ muuttujaa; klassinen 9×9 tarvitsee 729. Jokainen Sudoku-sääntö muuttuu tämän jälkeen joukoksi klausuuleja. (Klausuuli on muuttujien tai niiden negaatioiden TAI-operaatio; koko kaava on kaikkien klausuulien JA-operaatio.)

Jokaisessa solussa on vähintään yksi arvo — yksi klausuuli solua kohti:

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

Jokaisessa solussa on enintään yksi arvo — jokaiselle arvoparille yksi ”eivät molemmat” -klausuuli:

$$\neg x(1,1,1) \vee \neg x(1,1,2) \quad \neg x(1,1,1) \vee \neg x(1,1,3) \quad \text{... ja samoin kaikille kuudelle parille.}$$

Jokainen rivi sisältää jokaisen arvon — rivillä 1 arvon 3 on esiinnyttävä vähintään kerran:

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

ja enintään kerran: $\neg x(1,1,3) \vee \neg x(1,2,3)$, ja samoin jokaiselle solujen parille rivillä.

Sarakkeet ja lohkot — samanlaiset klausuulijoukot; vain soluryhmä vaihtuu. Vasemman yläkulman lohkolle ja arvolle 2:

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

sekä pareittaiset ”eivät molemmat” -klausuulit.

Painetut vihjeet — yksinkertaisin osa: jokainen vihje on yhden muuttujan klausuuli. Vasempaan yläkulmaan painettu 3 muuttuu klausuuliksi

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

Kaikkien näiden konjunktio (JA-operaatio) on toteutuva täsmälleen silloin, kun Sudokulla on ratkaisu — ja toteuttava sijoitus *on* ratkaisu: katso, mitkä x(r,c,v)-muuttujat ovat tosia, ja täytä ruudukko niiden mukaan. 9×9 -Sudokussa tämä tarkoittaa 729 muuttujaa ja muutamaa tuhatta klausuulia, jotka nykyaikainen SAT-ratkaisija käsittelee millisekunneissa. Huomaa vihjeklausuuli x(1,1,3): se sanoo ”tämä solu on täsmälleen 3”, ei ”nämä solut ovat kaikki eri” — sama epäsymmetria pakottaa käyttämään lisätemppua vihjesoluille alempana olevassa protokollahuomautuksessa.

SAT:ista Sudokuun. Tutkimus tarvitsee vastakkaisen ja vaikeamman suunnan: kun annetaan *mielivaltainen* SAT-kaava, rakennetaan MegaSudoku, jolla on ratkaisu täsmälleen silloin, kun kaava on toteutuva. Sudokun omat säännöt voivat sanoa vain ”nämä solut ovat kaikki eri”, joten mielivaltaiset loogiset rajoitteet täytyy *rakentaa* — ja juuri sitä gadgetit tekevät. Gadget on pieni ennalta rakennettu soluryhmä, yksi kaavan jokaista klausuulia kohti. Tietyt solut toimivat

muuttujina (niissä oleva symboli koodaa true- tai false-arvon), ja ryhmän sisäiset rajoitteet suunnitellaan niin, että sen ainoat lailliset täytöt vastaavat kyseisen klausuulin toteuttavia sijoituksia. Tämä on NP-täydellisyystodistusten vakiotekniikkaa; yleistetylle Sudokulle sen toteuttivat Yato ja Seta vuonna 2003.
Yhdessä nämä kaksi suuntaa sanovat, että N-by-N-Sudoku ja SAT ovat sama ongelma eri asuissa. Siksi tämä artikkeli — ja tutkimus — voi kertoa koko NP-luokan tarinan ruudukoiden ja symbolien avulla.

Todistaja on edelleen helppo kuvitella. Alice tuntee MegaSudokun täydellisen pätevän täytön. Bob haluaa vakuuttua, että sellainen täyttö on olemassa, mutta Alice ei halua paljastaa sitä. Jos hän lähettää koko täytön, Bob vakuuttuu, mutta salaisuus menetetään.

Klassisessa nollatietoversiossa Alice ja Bob ovat vuorovaikutuksessa. Vanhan koulukunnan mielikuva käyttää peitettyjä laattoja. Alice piilottaa ratkaistun ruudukon, nimeää symbolit salaa uudelleen ennen jokaista kierrosta ja antaa Bobin tarkistaa yhden satunnaisesti valitun paikallisen rajoitteen: rivin, sarakkeen, lohkon tai gadgetin. Jos avatut solut näyttävät kaikki eri symboleja, Bobin luottamus kasvaa. Sitten kaikki peitetään uudelleen ja symbolit nimetään tuoreella tavalla. (Yksi mutka: tehtävän annetut vihjeet tarvitsevat lisätempun, koska symbolien uudelleennimeäminen piilottaa myös ne. Alla oleva huomautus selittää, miten klassiset protokollat ratkaisevat tämän; tämä leikkimalli riittää seuraavaan vaiheeseen.)

Miten klassiset protokollat todella käsittelevät vihjesoluja

Uudelleennimeämistempulla on sokea piste. Rivi-, sarake- ja lohkosäännöt sanovat kaikki ”nämä solut ovat kaikki eri”, ja *kaikki eri* säilyy minkä tahansa symbolien uudelleennimeämisen läpi. Vihje taas sanoo ”tässä solussa on täsmälleen 5”, ja uudelleennimeämisen jälkeen Bob näkee vain $\sigma(5)$:n — jonkin peitetyn symbolin — tuntematta uudelleennimeämistä σ. Hän ei pysty tarkistamaan mitään. Ilman korjausta Alice voisi todistaa, että *jokin* pätevä ruudukko on olemassa, samalla kun hän sivuuttaa painetut vihjeet kokonaan. Se ei todistaisi mitään *tästä* tehtävästä. Klassinen kirjallisuus tuntee kaksi vakiokorjausta.
Paletti. Lisää piilotettuun ruudukkoon yksi ylimääräinen N solun rivi — paletti, jonka Alice täyttää symboleilla 1…N kiinteässä julkisessa järjestyksessä ja nimeää sitten uudelleen kaiken muun mukana, jolloin siinä ovat $\sigma(1)$…$\sigma(N)$. Bobin satunnaishaaste saa nyt yhden lisävaihtoehdon. Rivin, sarakkeen, lohkon tai gadgetin lisäksi hän voi valita avattavaksi **paletin ja yhden vihjesolun**. Alice paljastaa molemmat; paletti näyttää kyseisen kierroksen uudelleennimeämisen, ja Bob tarkistaa, että vihjesolussa on juuri painetun vihjeen uudelleennimetty versio. Nollatieto säilyy, koska Bob oppii vain σ:n — joka arvotaan joka kierroksella uudelleen eikä yksinään kerro mitään — sekä sellaisen solun arvon, jonka hän tiesi jo tehtävästä. Salaisista soluista ei vuoda mitään, ja simulaattori voi jäljitellä näkymän arpomalla satunnaisen σ:n. Luotettavuus säilyy, koska huijaava Alice jää kullakin kierroksella kiinni kiinteällä todennäköisyydellä, ja kierroksia toistetaan, kunnes epäily on mitätön.
Vihjeiden kääntäminen pois. Rakenteellisempi vaihtoehto poistaa erikoishaasteen sen sijaan, että lisäisi sellaisen. Vihjeen arvon *tarkistamisen* sijasta arvo pakotetaan erilaisuusrajoitteilla:

vihjesolu yhdistetään jokaiseen paletin soluun paitsi siihen, joka kantaa sen omaa arvoa — ”eri kuin $\sigma(1)$, eri kuin $\sigma(2)$, …, eri kuin kaikki paitsi $\sigma(5)$”. Ainoa symboli, jonka solu voi laillisesti sisältää, on vihjeen oma symboli. Jokainen rajoite on nyt taas muotoa ”nämä kaksi ovat eri” — muuttumaton uudelleennimeämisessä ja tarkistettavissa aivan kuten rivi. Samaa temppua käytetään valmiiksi väritetyille solmuille klassisessa graafin värityksen protokollassa, ja se kuvaa hyvin myös edellä käytettyä sanaa *gadget*: MegaSudoku-SAT-kuvassa vihjeet käännetään epätasa-arvogadgeteiksi aivan kuten muutkin rajoitteet.

Fyysinen protokolla. Todellisilla korteilla toteutettu Sudoku-protokolla (Gradwohl, Naor, Pinkas ja Rothblum, 2007) ei käytä uudelleennimeämistä lainkaan, vaan lukitsee vihjeet jo ennen piilottamisen alkua. Alice asettaa jokaista solua varten kolme samanlaista korttia, joissa on solun arvo — salaiset solut kuvapuoli alaspäin mutta **vihjesolut kuvapuoli ylöspäin**, jotta Bob näkee omin silmin vihjeiden toteutuvan ennen kuin kortit käännetään. Tämän jälkeen yksi kortti kustakin solusta menee rivinsä pinoon, yksi sarakkeensa pinoon ja yksi lohkonsa pinoon; jokainen pino sekoitetaan ja paljastetaan, ja Bob tarkistaa sen sisältävän kaikki N symbolia. Sekoittaminen tuhoaa sijaintitiedon (siinä on nollatieto), mutta vihjeet varmistettiin jo kortteja jaettaessa.

Kummassakin tapauksessa opetus on sama, johon tämä artikkeli palaa yhä uudelleen: nollatietoprotokolla pitää tarkasti kirjaa siitä, *mitkä* tosiasiat säilyvät piilottamisesta huolimatta. Uudelleennimeäminen säilyttää väitteen ”kaikki eri” ja pyyhkii pois väitteen ”on yhtä kuin 5” — joten ”on yhtä kuin 5” täytyy tuoda takaisin toisella keinolla.

Tämä ei ole tutkimuksen varsinainen protokolla. Se on mielikuva klassisesta nollatiedosta:

- Alice ja Bob käyvät vuoropuhelua.
- Bob valitsee satunnaisia tarkistuksia.
- Alice paljastaa vain paikallisen johdonmukaisuuden, ei koko ratkaisua.
- Yksityisyystodistus toimii osoittamalla, että Bobin näkymä olisi voitu tuottaa ilman Alicen salaista ratkaisua.

Klassinen nollatieto rakentuu siis myönteisen tosiasian varaan:

> Simulaattori on todella olemassa.

Poistetaan nyt mukavat osat. Alice lähettää yhden todistusmerkkijonon ja poistuu. Luotettua alkuasetelmaa ei ole, etukäteen valmisteltua yhteistä satunnaismerkkijonoa ei ole, eikä Bob saa koskaan hyväksyä väärää tehtävää. Tässä asetelmassa klassinen nollatieto ei voi selvitä.

Ennen temppua tarvitaan vielä yksi hahmo. Kiinnitetään **sääntökirja**: formaali todistusjärjestelmä loogikon tarkoittamassa mielessä — kiinteä joukko aksioomia ja mekaanisia sääntöjä kirjoitettujen matemaattisten todistusten tarkistamiseen. ZFC, matematiikan tavanomainen aksioomajärjestelmä, on kanoninen esimerkki. Kaikki tästä eteenpäin sanottu on suhteessa etukäteen valittuun sääntökirjaan, ja valinta on joustava: konstruktio toimii mille tahansa kiinnitetylle sääntökirjalle, myös ZFC:lle.

(Huomautus sanoista, tutkimuksen omaa käytäntöä seuraten: ”proof system” tarkoittaa tässä aina

tätä sääntökirjaa — formaalia järjestelmää, joka tarkistaa matemaattisia todistuksia — ei koskaan Alicen lähettämiä viestejä. Alicen ja Bobin koneistoa kutsutaan ”todistuksen esittäjäksi ja tarkistajaksi”.)

Gödel-tyylinen versio säilyttää MegaSudoku-tarinan mutta muuttaa todistusta.

Valitse toinen samankokoinen rajoitejärjestelmä ja kutsu sitä nimellä **D**. Tarinassa S ja D ovat kaksi samanmuotoista MegaSudoku(n)-tehtävää. Kulissien takana D on voinut alkaa erikokoisena vaikeana loogisena kaavana; tarvittaessa sitä voidaan täyttää harmittomilla täyterajoitteilla, jotta se sopii samaan ruudukkoon. D rakennetaan loogisesta kaavasta, joka on todellisuudessa **epätoteutuva**: ei ole olemassa arvosijoitusta, joka tekisi kaikki sen rajoitteet tosiksi, aivan kuten rikkinäisellä tehtävällä ei ole laillista valmista ruudukkoa. Leluesimerkki olisi kaava, joka vaatii yhtä aikaa ”X on tosi” ja ”X on epätosi”. D:llä ei siis ole pätevää täyttöä.

D ei kuitenkaan saa olla rikkinäinen tehtävä, jonka rikkinäisyys on *helppo osoittaa*. Edellinen leluesimerkki ei kelpaa: mikä tahansa sääntökirja kumoaa ”X ja ei-X” -väitteen yhdellä rivillä. D:n täytyy olla väärä tavalla, jota valittu sääntökirja ei pysty sertifioimaan lyhyellä argumentilla. Jos sääntökirja voisi kumota D:n lyhyellä todistuksella, alla oleva tarina romahtaisi: vaihtoehtoinen reitti, joka olisi voinut tuottaa todistuksia ilman Alicen salaisuutta, voitaisiin muodollisesti sulkea pois, ja sen mukana katoaisi yksityisyystakuu. Siksi D valitaan perheestä, jota kiinnitetty sääntökirja ei pysty tehokkaasti kumoamaan: sääntökirjan sisällä ei ole lyhyttä todistusta sille, ettei D:llä ole ratkaisua.

Alicen yhden viestin todistus koskee sitten joko–tai-väitettä:

> joko todellisella MegaSudokulla S on ratkaisu, tai harhautuksella D on ratkaisu.

Tämä on looginen kytkentä. D:tä **ei** tuoteta jollakin maagisella tavalla, joka tekisi S:n todeksi. Todistus ei väitä ”D:llä ei ole ratkaisua, siis S:llä on ratkaisu”. Se todistaa disjunktion **S tai D**. Täydellinen luotettavuus tarkoittaa, ettei väärällä disjunktiolla voi olla pätevää todistusta. Koska D on todellisuudessa väärä — sillä ei ole ratkaisua — disjunktio voi olla tosi vain, jos S on tosi. Jos todistus siis hyväksytään, S:llä täytyy olla ratkaisu. Harhautus ei voi muuttaa väärää S:ää todeksi.

Nollatietomaisen osan kannalta kysytään kuitenkin, mitä tapahtuisi, jos D:llä *olisi* ratkaisu. Tuo harhautusratkaisu toimisi vaihtoehtoisena todistajana. Sen avulla joku voisi tuottaa todistuksia tuntematta Alicen todellista MegaSudoku-ratkaisua — toisin sanoen se toimisi simulaattorina. Todellisuudessa D:llä ei ole ratkaisua, joten tämä simulaattorireitti on suljettu. Oleellista on, ettei sääntökirja pysty tehokkaasti todistamaan reitin olevan suljettu.

D:llä on siis kaksi tehtävää. **Luotettavuuden** kannalta D on väärä, joten ”S tai D” -väitteen pätevä todistus pakottaa S:n todeksi. **Tehokkaan nollatiedon** kannalta D:tä on vaikea kumota, joten sääntökirja ei pysty nopeasti sulkemaan pois sitä harhautusreittiä, joka olisi mahdollistanut simulaation.

Turvallisuustesti ei siis enää ole:

> Voimmeko todistaa, että simulaattori on todella olemassa?

Vaan:

> Pystyykö sääntökirjasi tehokkaasti todistamaan, että simulaattori on mahdoton?

Jos vastaus on ei, seuraa jotain yllättävän vahvaa: jokainen turvallisuustakuu, joka (a) voidaan havaita suorittamalla testi ja (b) voidaan sääntökirjan sisällä todistaa simulaattorin olemassaolon seuraukseksi, todella pätee. Onnistunut hyökkäys mitä tahansa niistä vastaan muodostaisi itse puuttuvan lyhyen kumoamistodistuksen, eikä tällaista lyhyttä kumoamista ole. Tässä on tehokkaan nollatiedon ”tehokas” osa.

Luokkahuonevertailu kuuluu siis näin:

Klassinen nollatieto: todistukset ovat turvallisia, koska simulaattori on olemassa.

Gödel-tyylinen tehokas nollatieto: todistuksia kohdellaan havaittavissa turvallisuustesteissä turvallisina, koska sääntökirja ei pysty tehokkaasti todistamaan simulaattorin mahdottomuutta.

Jälkimmäinen väite on heikompi. Juuri siksi tutkimus voi säilyttää kolme klassisen version rikkovaa ominaisuutta: yhden viestin, alkuasetelman puuttumisen ja täydellisen luotettavuuden.

Uusi testi: et pysty todistamaan simulaattorin puuttumista

Ilangon lievennys muuttaa kysymystä.

Klassinen nollatieto kysyy:

> Onko simulaattori olemassa?

Tehokas nollatieto (*effectively zero-knowledge*) kysyy jotakin heikompaa:

> Pystyykö valitsemasi sääntökirja tehokkaasti todistamaan, ettei simulaattoria ole?

Tämä voi kuulostaa tekniseltä kiertotieltä, mutta siinä on koko idea. Konstruktio elää oudossa tilassa: simulaattoria ei todellisuudessa ole — tutkimus sanoo tämän suoraan — mutta kiinnittämäsi sääntökirja ei pysty tehokkaasti todistamaan, ettei sitä ole. Jos jokainen sinua kiinnostava huono seuraus vaatisi tällaista kumoamista, järjestelmä käyttäytyy näiden seurausten suhteen silti nollatiedon tavoin.

Tässä Gödel tulee mukaan. Ei koristeena eikä väitteenä ”Gödel tekee kryptografiasta turvallista”. Yhteys on todistusteoreettinen. Sääntökirjaa kutsutaan **optimaaliseksi**, jos se on täsmällisessä mielessä paras mahdollinen: aina kun jokin sääntökirja pystyy kumoamaan tietynlaisen kaavan lyhyellä todistuksella, optimaalinen sääntökirja pystyy samaan todistuksella, joka on korkeintaan polynomisesti pidempi. Krajíček ja Pudlák esittivät vuonna 1989 konjektuurin, jonka mukaan **optimaalista todistusjärjestelmää ei ole olemassa**: minkä tahansa sääntökirjan kiinnitätkin, jokin toinen sääntökirja todistaa jonkin tosien väitteiden perheen paljon tiiviimmin. Tämä on yksi todistuskompleksisuuden keskeisistä avoimista konjektuureista ja Gödelin epätäydellisyyslauseen

äärellinen, kompleksisuusteoreettinen serkku: joillakin tosilla väitteillä ei ole lyhyttä todistusta valitsemassasi sääntökirjassa — ei siksi, etteikö niitä voisi periaatteessa todistaa, vaan siksi, että jokainen kiinteä sääntökirja jättää jotkin lyhyesti ilmaistavat totuudet ilman lyhyttä todistusta.

Tutkimus olettaa tämän konjektuurin (hieman vahvemmassa, kryptografiassa tavanomaisessa ”äärettömän usein” (*infinitely often*) -muodossa). Krajíčekin ja Pudlákin lause antaa vastineeksi konkreettisen tuloksen: jokaiselle sääntökirjalle löytyy kaavajono, jonka kaavat ovat aidosti epätoteutuvia mutta joita sääntökirja ei pysty kumoamaan lyhyillä todistuksilla — ja ratkaisevaa on, että tehokas algoritmi voi *tuottaa* ne. Tämä viimeinen ominaisuus, uniformisuus eli algoritminen yhtenäisyys, muuttaa idean pelkästä olemassaoloväitteestä todelliseksi algoritmiksi, jonka Alice voi suorittaa: hänen harhautuksensa D tulevat tuotantolinjalta, eivät tyhjästä.

Kryptografinen siirto on ottaa tämä todistusvoiman puute käyttöön.

Mitä konstruktio tekee

Tässä tutkimuksen konstruktio pelkistettynä muotoonsa.

Kiinnitä sääntökirja — esimerkiksi ZFC. Todistuskompleksisuusoletuksen mukaan on olemassa tehokkaasti tuotettava jono kaavoja, jotka ovat todellisuudessa epätoteutuvia mutta joiden epätoteutuvuudesta sääntökirjalla ei ole lyhyttä todistusta.

Rakenna sitten yhden viestin todistus muotoon:

> joko todellinen väite on toteutuva, tai tämä erityinen vaikea kaava on toteutuva.

Erityinen vaikea kaava ei ole toteutuva. Jos taustalla oleva todistuskoneisto on täysin luotettava, viestin hyväksyminen tarkoittaa siis edelleen, että todellinen väite on tosi. Tästä saadaan täydellinen luotettavuus.

Nollatiedon kaltaista turvallisuutta varten kuvitellaan kuitenkin, että erityinen vaikea kaava *olisi* toteutuva. Silloin sen todistajalla voitaisiin simuloida todistuksia ilman todellista todistajaa. Kaava ei todellisuudessa ole toteutuva — mutta sääntökirja ei pysty tehokkaasti todistamaan tätä. Siksi se ei pysty tehokkaasti todistamaan simulaattorin mahdottomuutta.

Tämä on nivelkohta. Järjestelmä ei piilota salaisuutta tuottamalla klassista simulaattoria. Se piilottaa salaisuuden suurelta joukolta havaittavissa olevia turvallisuustestejä sääntökirjan sen kyvyttömyyden taakse, ettei se pysty sertifioimaan simulaattorin puuttumista.

Mitä tutkimus väittää

Päätulos rakentuu kerroksittain. Ydintulos on tämä:

Vakiintuneen kryptografisen oletuksen — **ei-vuorovaikutteisten, todistajan suhteen erottamattomien todistusten (*non-interactive witness indistinguishable proofs*)**, jotka ovat hyvin tutkittuja ja seuraavat useista tunnetuista oletuspaketeista — sekä todistuskompleksisuuden konjektuurin, jonka mukaan **optimaalista todistusjärjestelmää ei ole olemassa edes ”äärettömän usein” (*infinitely often*) -mielessä**, vallitessa tutkimus rakentaa jokaista sääntökirjavalintaa varten NP/SAT:lle yhden viestin todistuksen esittäjän ja tarkistajan, joilla on **täydellinen luotettavuus (perfect soundness)**, ei alkuasetelmaa ja jotka toteuttavat suhteessa kyseiseen sääntökirjaan tehokkaan nollatiedon (*effectively zero-knowledge*). (NP/SAT on pulmatyyppisten ongelmien vakiintunut ”vaikein yhteinen nimittäjä”; MegaSudoku on yksi sen asuista.)

Laajempaa, falsifioitavissa olevien turvallisuusominaisuuksien säilymistä koskevaa väitettä varten tutkimus lisää vielä yhden tavanomaisen oletuksen, derandomisaatiota koskevan uskomuksen **P = BPP** (karkeasti: satunnaisuus ei anna algoritmeille olennaista lisävoimaa).

Lausekielestä tavalliseksi kieleksi käännettynä:

- Todistus on yksi viesti.
- Luotettua alkuasetelmaa ei ole.
- Vääriä väitteitä ei voida todistaa.
- Todistuksen esittäjä ei ole klassisen nollatiedon mukainen — sillä ei ole simulaattoria.
- Mutta jokainen klassisen nollatiedon falsifioitavissa oleva, pelipohjainen turvallisuusseuraus voidaan saavuttaa tässä asetelmassa.

”Falsifioitavissa oleva” on tärkeä ilmaus. Se tarkoittaa, että turvallisuuden pettäminen voidaan testata ajamalla vastustajaa pelissä. Monet kryptografiset turvallisuusmääritelmät ovat tätä muotoa: pystyykö vastustaja erottamaan kaksi salausta, kääntämään funktion, palauttamaan todistajan tai voittamaan jonkin määritellyn kokeen? Lause antaa yhden todistuksen esittäjän kutakin falsifioitavaa ominaisuutta varten, yksi ominaisuus kerrallaan. Yksi todistuksen esittäjä, jolla olisi *kaikki* falsifioitavat ominaisuudet yhtä aikaa, on luultavasti mahdoton — vanha uudelleenkäyttöhyökkäys (”Bob voi näyttää todistuksen muille”) on itse falsifioitava ominaisuus, ja se todella epäonnistuu tässä. Tutkimus ehdottaa, että yksi todistuksen esittäjä voisi uskottavasti kattaa kaikki *luonnolliset* falsifioitavat ominaisuudet — ne, joita kryptografisessa käytännössä oikeasti esiintyy — mutta tämä osa on ehdollinen lause, joka nojaa epämuodolliseen ”luonnollisen” käsitteeseen sekä eksplisiittiseen konjektuuriin. Takuu tähtää havaittaviin epäonnistumisiin, ei kaikkiin filosofisiin tai simulaatiopohjaisiin salaisuuden merkityksiin.

Yksi konkreettinen seuraus kannattaa nimetä: konstruktio antaa ensimmäiset ei-vuorovaikutteiset *todistajan piilottavat (witness-hiding)* todistukset uniformilla eli algoritmisesti yhtenäisellä todistuksen esittäjällä — ”pulman todistus ei auta sinua löytämään sen ratkaisua”, ilman vuorovaikutusta ja ilman alkuasetelmaa. Vaatimattomalta kuulostava kohde oli vastustanut konstruktiota vuosikymmeniä.

Mitä tämä ei sano

Tämä osio pitää jutun rehellisenä.

Se **ei** sano, että vanhat mahdottomuuslauseet olivat väärässä. Konstruktio kiertää ne muuttamalla määritelmää.

Se **ei** anna tavallista klassista nollatietoa ilman vuorovaikutusta, ilman alkuasetelmaa ja täydellisellä luotettavuudella. Tutkimus sanoo suoraan, ettei rakennetulla todistuksen esittäjällä ole simulaattoria.

Se **ei** tarkoita, ettei todistusta voisi käyttää uudelleen. Yhden viestin todistus voidaan edelleen näyttää jollekulle toiselle; tutkimus ei säilytä kiistettävyyden kaltaisia ominaisuuksia. (Luotettuun alkuasetelmaan perustuvalla ei-vuorovaikutteisella nollatiedolla on sama rajoitus.)

Se **ei** tarkoita, että kyseessä olisi käyttöönottovalmis käytännön protokolla. Tämä on kompleksisuusteoriaa ja kryptografian perusteita. Tulos riippuu merkittävistä todistuskompleksisuuden ja kryptografian oletuksista, ja konstruktio käsittelee sitä, mikä on periaatteessa mahdollista.

Se **ei** tee ”Gödelistä” maagista turvallisuusprimitiiviä. Gödel-yhteys kulkee todistusjärjestelmien, optimaalisten todistusjärjestelmien ja epätäydellisyyden äärellisten analogioiden kautta. Hyödyllinen intuitio ei ole ”epätäydellisyys suojaa salasanaasi”. Se on: jos sääntökirja ei pysty tehokkaasti todistamaan simulaattorin olevan mahdoton, tällaisen todistuksen vaativat hyökkäykset voidaan estää turvallisuusmääritelmien tasolla.

Miksi tämä on silti kiinnostavaa

Kryptografiassa vaikeus muutetaan usein turvallisuudeksi. Tekijöihin jakaminen on vaikeaa, joten RSA-tyyppisistä oletuksista tulee hyödyllisiä. Hilaongelmat ovat vaikeita, joten hilakryptografia on hyödyllistä. Tässä vaikeus on oudompaa: ei ”salaisuus on vaikea laskea”, vaan ”on vaikea todistaa, ettei tiettyä todistusoliota voi olla olemassa”.

Siksi tutkimus tuntuu epätavalliselta. Se kohtelee aksioomia ja sääntökirjoja melkein kryptografisina resursseina. Tavallinen mahdottomuustulos sanoo, että luotettavuuden ja simulaation välillä on jännite. Ilangon liike asettaa tämän jännitteen todistusteoreettisen verhon taakse: simulaattori puuttuu, mutta formaali järjestelmä ei pysty tehokkaasti paljastamaan tätä puutetta.

Lukijan kannalta yllättävä asia ei ole, että tämä korvaisi nykyiset nollatietojärjestelmät. Todennäköisesti ei korvaa, ainakaan suoraan. Yllättävää on, että matemaattisen logiikan rajoitusta voidaan käyttää rakentavasti: ei vain seinänä vaan eräänlaisena suojana.

Kuinka vahvaa näyttö on?

Tämä on lauseisiin perustuva teoreettinen tutkimus, joten ”näyttö” tarkoittaa tässä eri asiaa kuin biologian tai tähtitieteen tutkimuksessa. Kysymys ei ole siitä, toistuiko koe. Kysymys on siitä, tukevatko määritelmät, oletukset ja todistusketju väitettä.

Todistus on formaali, ja tutkimus ilmaisee oletuksensa selvästi. Oletukset eivät ole kevyitä. Ei-vuorovaikutteiset, todistajan suhteen erottamattomat todistukset (*non-interactive witness indistinguishable proofs*) ovat kryptografian vakiintuneita olioita ja seuraavat useista tunnetuista oletuspaketeista. Konjektuuri optimaalisen todistusjärjestelmän puuttumisesta on todistuskompleksisuuden keskeinen konjektuuri. P = BPP on tavanomainen derandomisaatio-oletus, jota tarvitaan vain laajempaan falsifioitavia ominaisuuksia koskevaan lauseeseen.

Tutkimus myös perustelee, että oletukset ovat oikea hinta eivätkä mielivaltainen rakennusteline: se todistaa käänteisen tuloksen, jonka mukaan ne ovat olennaisilta osin välttämättömiä — jos tällaisia konstruktioita ylipäätään on olemassa, ei-vuorovaikutteisia, todistajan suhteen erottamattomia todistuksia täytyy olla olemassa ja (olettaen tavanomaiset yksisuuntaiset eli *one-way*-funktiot) optimaalista todistusjärjestelmää ei voi olla. Lisäksi asetelma on kummassakin tapauksessa kiinnostava: minkä tahansa niistä kumoaminen olisi itsessään merkkipaalu todistuskompleksisuudessa, kryptografiassa tai kompleksisuusteoriassa.

Koska tulos on kuitenkin ehdollinen, myös luottamus siihen on ehdollista. Jos oletukset eivät pidä paikkaansa, lauseen tulkinta muuttuu. Ja vaikka oletukset pitäisivät, takuu ei ole täysi klassinen nollatieto vaan tutkimuksen lievennetty, todistusteoreettinen versio.

Siksi oikea luottamustaso on korkea sille, että tutkimus osoittaa johdonmukaisen ehdollisen mahdollisuustuloksen; kohtalainen sille, että oletukset kuvaavat sitä kryptografista maailmaa, jossa todella elämme; ja matala välittömille käytännön seurauksille.

Miksi sillä on merkitystä

Tutkimus avaa reitin, jonka piti olla suljettu.

Klassinen teoria sanoo: täyttä nollatietoa ei voi saada yhteen viestiin ilman alkuasetelmaa, eikä sillä voi olla täydellistä luotettavuutta. Ilangon tutkimus sanoo: jos kysymme nollatiedon seurauksia, joita voidaan testata turvallisuuspeleissä, ja annamme turvallisuusmääritelmän riippua siitä, mitä sääntökirja pystyy tai ei pysty tehokkaasti kumoamaan, suuri osa hyödyllisestä käyttäytymisestä voidaan palauttaa — yhdellä viestillä, ilman alkuasetelmaa ja täydellisellä luotettavuudella.

Tämä ei ole pieni määritelmän hienosäätö. Se on erilainen tapa ajatella kryptografisia takuita. Sen sijaan että kysyttäisiin vain, mitä on olemassa, kysytään myös, minkä sääntökirja pystyy sulkemaan pois. Sen sijaan että todistamattomuutta pidettäisiin filosofisena riesana, sitä käytetään rakenteena.

Käytännön maailma ei ehkä muutu huomenna. Käsitteellinen kartta kuitenkin muuttuu. Nyt on olemassa formaali merkitys, jossa ”kukaan ei pysty tehokkaasti todistamaan, että salaisuus vuoti” voi

olla riittävän vahva palauttamaan monia niistä pelipohjaisista suojista, joita halusimme väitteeltä ”salaisuus ei vuotanut”.

Siksi Gödel kuuluu otsikkoon.

Tiivis yhteenveto

Nollatietotodistuksissa todistuksen esittäjä voi vakuuttaa tarkistajan väitteen paikkansapitävyydestä paljastamatta todistajaa. Klassiset mahdottomuustulokset sanovat, ettei nollatietoa voida puristaa yhteen viestiin ilman alkuasetelmaa eikä sille voida saada täydellistä luotettavuutta. Rahul Ilangon tutkimus ei kumoa näitä mahdottomuuksia. Se määrittelee heikomman käsitteen, tehokkaan nollatiedon (*effectively zero-knowledge*): sen sijaan että vaadittaisiin simulaattorin todella olevan olemassa, vaaditaan, ettei valittu todistusjärjestelmä — esimerkiksi ZFC:n kaltainen formaali sääntökirja — pysty tehokkaasti todistamaan, ettei simulaattoria ole. Kryptografian merkittävien oletusten (ei-vuorovaikutteiset, todistajan suhteen erottamattomat todistukset) ja todistuskompleksisuuden oletuksen (optimaalista todistusjärjestelmää ei ole) vallitessa tutkimus rakentaa NP/SAT:lle yhden viestin todistuksen esittäjiä ilman alkuasetelmaa ja täydellisellä luotettavuudella; ne saavuttavat nollatiedon falsifioitavissa olevat, pelipohjaiset seuraukset ominaisuus kerrallaan. Yksi todistuksen esittäjä, joka kattaisi kaikki ”luonnolliset” tällaiset ominaisuudet, on pidemmälle menevä ja osittain konjekturaalinen laajennus — ja kirjaimellisesti kaikkien falsifioitavien ominaisuuksien kattaminen on todennäköisesti mahdotonta, koska todistuksia voi edelleen käyttää uudelleen. Tulos on teoreettinen ja ehdollinen, ei käyttöön otettu primitiivi, mutta se osoittaa uuden tavan käyttää todistusteoreettista todistamattomuutta kryptografisena resurssina.

Realistinen arvio

Mitä tutkimus osoittaa: Ilmoitettujen oletusten vallitessa voidaan rakentaa NP/SAT:lle yhden viestin, alkuasetelmattomia ja täydellisen luotettavuuden omaavia todistuksen esittäjiä, jotka toteuttavat tehokkaan nollatiedon (*effectively zero-knowledge*) minkä tahansa valitun todistusjärjestelmän suhteen ja saavuttavat klassisen nollatiedon falsifioitavissa olevat pelipohjaiset seuraukset yksi kerrallaan.

Mikä on uskottavaa mutta ei todistettu ilman ehtoja: Että tarvittavat todistuskompleksisuuden ja kryptografian oletukset pitävät paikkansa. Ne ovat vakavasti otettuja ja hyvin tutkittuja oletuksia — ja tutkimus osoittaa niiden olevan olennaisilta osin sekä välttämättömiä että riittäviä — mutta ne ovat silti oletuksia.

Mitä se ei osoita: Klassista nollatietoa ilman vuorovaikutusta, ilman alkuasetelmaa ja täydellisellä luotettavuudella; käyttöönottovalmista käytännön järjestelmää; kiistettävyyttä tai todistusten uudelleenkäytön estoa; eikä sitä, että Gödelin epätäydellisyyslause yksin tekisi kryptografiasta turvallista.

Tärkeimmät rajoitukset: Takuu on nollatiedon lievennys; laajin versio riippuu useista oletuksista; yhtä universaalia todistuksen esittäjää koskevat väitteet ovat edelleen osittain konjekturaalisia; ja tulos on ensisijaisesti perusteoreettinen.

Kuinka paljon yleislukijan kannattaa luottaa tulokseen? Vahvasti siihen, että tämä on tärkeä ehdollinen teoriatulos, jos määritelmät hyväksytään. Kohtalaisesti siihen, että oletukset kuvaavat todellisuutta. Heikosti välittömään käytännön käyttöönottoon. Turvallinen johtopäätös on: tutkimus ei riko nollatiedon mahdottomuustuloksia, vaan löytää uuden todistusteoreettisen tavan kiertää niitä osia, joilla on merkitystä monissa turvallisuuspeleissä.

Lähteet

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`