The Clean Paper

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

Криптографічний доказ може приховувати секрет, роблячи відсутність симулятора важкою для доведення

Докази з нульовим розголошенням дозволяють довести твердження, не розкриваючи witness. Класична теорія каже, що в повному сенсі це неможливо одночасно з одним повідомленням, відсутністю довіреного setup і perfect soundness. Стаття Рахула Іланго не скасовує цю неможливість. Вона змінює ціль: замість вимагати, щоб симулятор справді існував, достатньо, щоб жодна вибрана система доказів не могла ефективно довести, що симулятора не існує. За великих припущень із теорії складності доказів і криптографії цього вистачає, щоб по одній властивості за раз відновити фальсифіковані, ігрові наслідки zero-knowledge. Це не готовий інтернет-примітив, а теоретико-доказовий спосіб перетворити математичну недовідність на криптографічне прикриття.

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

Фокус не в тому, щоб довести, що секрет прихований

Почнімо з найпростішої версії нульового розголошення.

Аліса хоче переконати Боба, що судоку має розв'язок. Якщо вона надішле сам розв'язок, Боб переконається — але головоломка буде зіпсована. Їй потрібно щось дивніше: доказ того, що розв'язок існує, без розкриття самого розв'язку.

Саме це обіцяє доказ із **нульовим розголошенням (zero-knowledge proof)**. Доведувач (Аліса) переконує перевіряча (Боба), що твердження істинне, не розкриваючи нічого, крім самого факту його істинності.

Проблема в тому, що ця обіцянка має ціну. Звичайний математичний доказ має дві зручні властивості. Він складається з **одного повідомлення**: ви записуєте його, передаєте й ідете. І він має **досконалу надійність щодо хибних тверджень (perfect soundness)**: для хибного твердження взагалі не існує коректного доказу. Класичні теореми неможливості кажуть, що zero-knowledge мусить відмовитися від обох властивостей — і не лише від їх поєднання: кожна окремо теж недоступна.

По-перше, доказ із нульовим розголошенням потребує взаємодії. Якщо Аліса надсилає одне повідомлення й немає заздалегідь організованого довіреного налаштування, гарантія zero-knowledge руйнується — незалежно від того, наскільки ви готові послабити soundness в обмін.

По-друге, zero-knowledge потребує невеликої допустимої ймовірності помилки. Вимога perfect soundness непомітно знищує й взаємодію: якщо перевіряча неможливо обдурити за жодного набору його випадкових виборів, він міг би просто зафіксувати ці вибори наперед. А щойно перевіряч стає передбачуваним, Аліса може відповісти на все одним повідомленням — тобто ми повертаємося саме до випадку, який уже неможливий.

Стаття Рахула Іланго пропонує шлях навколо цієї подвійної стіни. Не заперечуючи її існування й не створюючи класичне zero-knowledge там, де воно неможливе. Хід тонший: послабити значення фрази «нічого не розкриває», але так, щоб зберегти ті властивості безпеки, які криптографи справді можуть перевіряти.

Результат називається **ефективним нульовим розголошенням (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 заблоковане трьома дверима — взаємодією, довіреним налаштуванням і недосконалою soundness. Конструкція Іланго проходить іншим шляхом: вибрана формальна система не може ефективно спростувати існування симулятора.

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.

Класичне zero-knowledge запитує, чи існує симулятор; «effectively zero-knowledge» запитує лише, чи

може вибрана формальна система ефективно довести, що симулятор не існує. Саме слабше запитання дозволяє конструкції зберегти одне повідомлення, відсутність налаштування та perfect soundness.

Original diagram — The Clean Paper CC BY 4.0

Старий критерій: симулятор існує

Класичне формальне визначення zero-knowledge використовує уявного помічника — **симулятор**.

Ідея така. Уявімо Джейн, яка **не** знає секрету Аліси. Якщо Джейн може самостійно генерувати докази, що виглядають так само, як докази, які Боб отримав би від Аліси, то докази Аліси не навчили Боба нічого нового. Джейн уже могла підробити весь досвід без секрету Аліси.

Тому класичне zero-knowledge вимагає реального симулятора. Має існувати ефективний алгоритм, здатний генерувати правдоподібні «фальшиві» докази без знання секрету — **свідка (witness)** у термінології галузі; для судоку свідком є просто заповнена правильна сітка.

Це сильне визначення, але саме тут спрацьовує стара неможливість. Інтуїція така. Справді невзаємодійний доказ — це просто рядок. Щойно Боб отримав цей рядок, він може показати його комусь іншому: він здобув здатність переконувати інших у твердженні, а це вже звучить як щось більше за «нічого». Класичні теореми перетворюють цю інтуїцію на наведені вище результати неможливості.

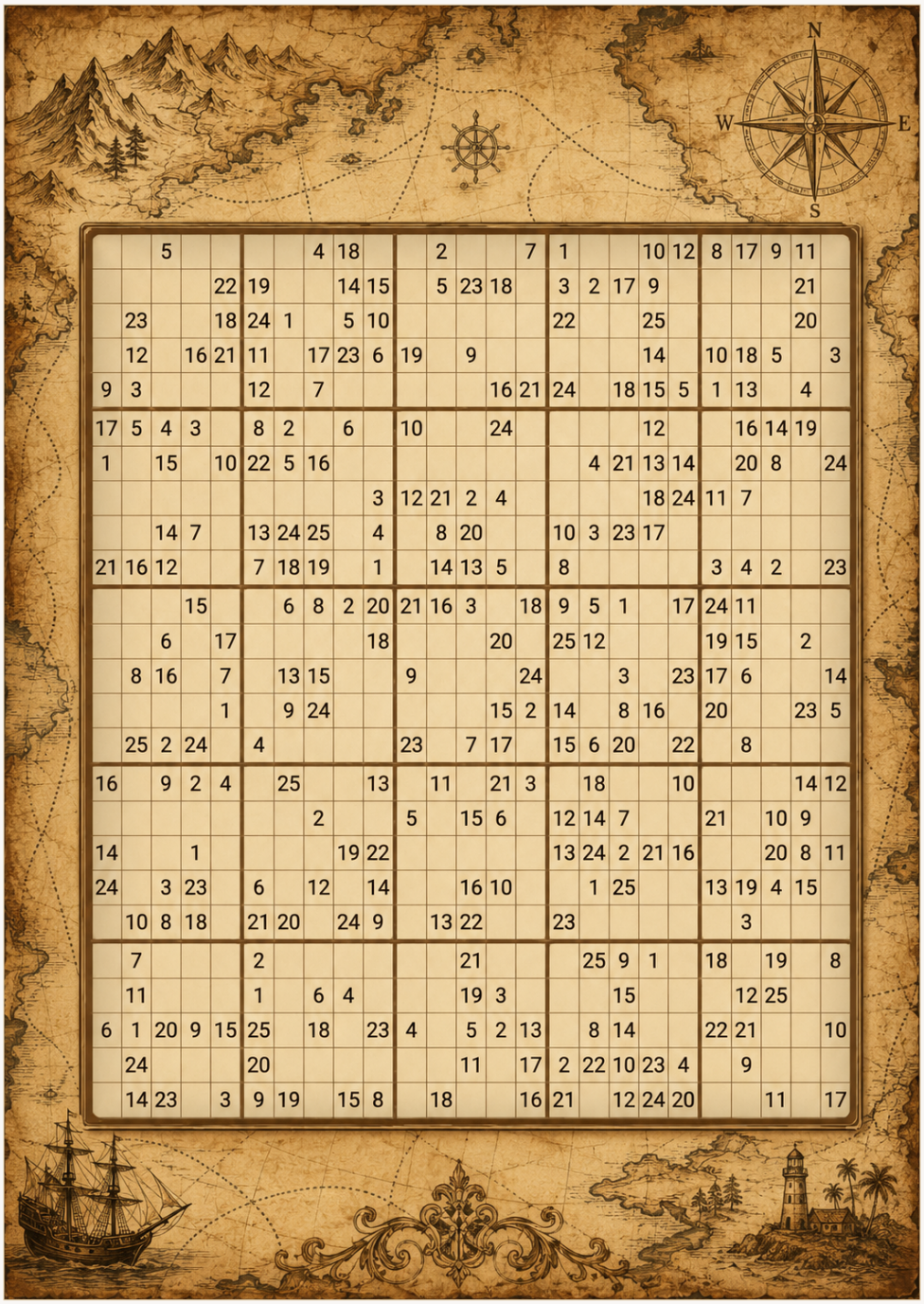
Три властивості, від яких ця стаття не відмовляється

Назва роботи прямо вказує на три обмеження:
Без взаємодії: Аліса надсилає один рядок-доказ. Немає протоколу з обміном повідомленнями туди й назад.
Без налаштування: Аліса й Боб не покладаються на довірений спільний референсний рядок або іншу заздалегідь підготовлену публічну випадковість. Багато систем, які називають «невзаємодійним zero-knowledge», все одно потребують setup; тут setup немає взагалі.
Perfect soundness: хибне твердження не має жодного коректного доказу. Не «майже ніколи не приймається», а «коректного доказу не існує».
Саме ці три властивості має звичайна письмова математика — і, як пояснено вище, класичне zero-knowledge не може зберегти їх разом.

MegaSudoku як спосіб відчути різницю

Ось навмисно спрощена аналогія.

Для серйозної частини аналогії не варто брати звичайне судоку 9×9. Воно надто мале й скінченне: комп'ютер може просто розв'язати його або довести, що розв'язку немає. Натомість уявімо сімейство головоломок **MegaSudoku(n)**. Масштабуємо звичайне правило: вибираємо розмір блока n, задаємо $N = n^2$ і будуємо сітку $N \times N$, поділену на блоки $n \times n$, з N символами. Звичайне судоку — лише крихітний випадок $n = 3$, $N = 9$: сітка 9×9, блоки 3×3 і дев'ять символів. Історія про складність доказів починається лише тоді, коли n може зростати, а сітка отримує додаткові ґаджети, які змушують її поводитися як формула SAT, перевдягнена в судоку. Формула SAT — це просто набір обмежень «так/ні»: чи можна присвоїти змінним значення істина/хиба так, щоб усі обмеження виконувалися?



Судоку 25×25: його правила можна перевіряти, не розкриваючи завершеної сітки — візуальна метафора доказу, який підтверджує існування прихованого розв’язку, тобто witness.

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

Судоку й SAT: одна головоломка у двох костюмах

Твердження, що судоку може «поводитися як формула SAT», — не метафора. Переклад працює в обох напрямках, і простіший напрямок можна записати повністю.

Від судоку до SAT. SAT оперує лише істинністю/хибністю, тому введемо одну

булеву змінну для кожної трійки (рядок, стовпець, значення): *x(r,c,v)* означає «клітинка в рядку *r*, стовпці *c* містить значення *v*». Судоку 4×4 (блоки 2×2, значення 1–4) потребує 4·4·4 = 64 змінних; класичне 9×9 — 729. Кожне правило судоку перетворюється на набір клауз. (Клауза — це OR змінних або їх заперечень; уся формула є AND усіх клауз.)
Кожна клітинка містить принаймні одне значення — одна клауза на клітинку:

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

Кожна клітинка містить не більше одного значення — клауза «не обидва» для кожної пари значень:

¬x(1,1,1) ∨ ¬x(1,1,2) ¬x(1,1,1) ∨ ¬x(1,1,3) … і так далі для всіх шести пар.

Кожен рядок містить кожне значення — для рядка 1 і значення 3: принаймні один раз,

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

і не більше одного разу: ¬x(1,1,3) ∨ ¬x(1,2,3), і так далі для кожної пари клітинок у рядку.
Стовпці та блоки — ті самі набори; змінюється лише група клітинок. Для верхнього лівого блока й значення 2:

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

плюс попарні клаузи «не обидва».
Надруковані підказки — найпростіша частина: кожна підказка є клаузою з однією змінною. Надрукована 3 у верхньому лівому куті стає клаузою

x(1,1,3)

AND усього цього є здійсненним тоді й лише тоді, коли судоку має розв'язок, а задовільне присвоєння *і є* розв'язком: дивимося, які x(r,c,v) істинні, і заповнюємо сітку. Для 9×9 це 729 змінних і кілька тисяч клауз — сучасний SAT-solver розбирається з ними за мілісекунди. Зверніть увагу на клаузу підказки x(1,1,3): вона означає «ця клітинка дорівнює саме 3», а не «всі ці клітинки різні». Та сама асиметрія нижче змусить нас використати додатковий трюк для клітинок-підказок.
Від SAT до судоку. Статті потрібен протилежний і складніший напрямок: за *довільною* формулою SAT побудувати mega-Sudoku, яке має розв'язок тоді й лише тоді, коли формула здійсненна. Власні правила судоку вміють сказати лише «ці клітинки всі різні», тому довільні логічні обмеження треба *сконструювати* — саме для цього потрібні ґаджети. Ґаджет — невеликий заздалегідь спроєктований кластер клітинок, по одному на кожну клаузу формули, в якому виділені клітинки відіграють роль змінних (символ у них кодує істину або хибу), а внутрішні обмеження спроєктовані так, щоб єдині

допустимі заповнення відповідали присвоєнням, які задовольняють цю клаузу. Це стандартна інженерія з доказів NP-повноти; для узагальненого судоку її реалізували Ято й Сета у 2003 році.
Разом обидва напрямки кажуть, що судоку N×N і SAT — одна й та сама задача у двох костюмах. Саме це дозволяє цій статті — і самій роботі — розповідати про весь NP мовою сіток і символів.

Свідок усе ще легко уявити. Аліса знає повне коректне заповнення mega-Sudoku. Боб хоче переконатися, що таке заповнення існує, але Аліса не хоче його розкривати. Якщо вона надішле всю сітку, Боб переконається, але секрет зникне.

У класичній zero-knowledge версії Аліса й Боб взаємодіють. Старий наочний образ використовує закриті плитки. Аліса ховає розв'язану сітку, перед кожним раундом таємно перейменовує символи й дозволяє Бобу перевірити одне випадково вибране локальне обмеження: рядок, стовпець, блок або ґаджет. Якщо відкриті клітинки містять усі різні символи, довіра Боба зростає. Потім усе знову закривають, а символи заново перейменовують. (Є одна тонкість: задані підказки головоломки потребують окремого трюку, бо перейменування приховує і їх. Примітка нижче пояснює, як класичні протоколи це вирішують.)

Як класичні протоколи насправді працюють із клітинками-підказками

Трюк із перейменуванням має сліпу пляму. Правила рядка, стовпця й блока всі кажуть «ці клітинки попарно різні», а властивість *усі різні* зберігається за будь-якого перейменування символів. Але підказка каже: «ця клітинка містить саме 5». Після перейменування Боб бачить лише $\sigma(5)$ — якийсь замаскований символ — і не знає перестановки σ. Перевірити нічого не можна. Якщо це не виправити, Аліса могла б довести існування *якоїсь* правильної сітки, повністю ігноруючи надруковані підказки, а це нічого не доводить про *цю* головоломку. У класичній літературі є два стандартні способи виправлення.
Палітра. До прихованої сітки додають один додатковий рядок із N клітинок — палітру, яку Аліса заповнює символами 1…N у фіксованому публічному порядку, а потім перейменовує разом з усім іншим, тож там містяться $\sigma(1)\ldots\sigma(N)$. Тепер у випадкового виклику Боба з'являється ще один варіант. Крім рядка, стовпця, блока або ґаджета він може вибрати **палітру плюс одну клітинку-підказку**. Аліса відкриває обидва; палітра показує перейменування цього раунду, а Боб перевіряє, що клітинка-підказка містить саме перейменовану версію надрукованої підказки. Zero-knowledge зберігається, бо Боб дізнається лише σ — нову випадкову перестановку для цього раунду, яка сама по собі нічого не дає, — і значення клітинки, яке він уже знав із головоломки. Нічого про секретні клітинки не витікає, а симулятор може підробити таку картину, просто вибравши випадкову σ. Soundness зберігається, бо шахрайку Алісу в кожному раунді ловлять із фіксованою ймовірністю, а раунди повторюють, доки ймовірність помилки не стане мізерною.

Компіляція підказок у обмеження. Більш структурний варіант не додає спеціальну перевірку, а усуває її потребу. Замість того щоб *перевіряти* значення підказки, його примушують виконуватися через обмеження нерівності: клітинку-підказку з'єднують з усіма клітинками палітри, крім тієї, що несе її власне значення — «відмінна від $\sigma(1)$, відмінна від $\sigma(2)$, …, відмінна від усього, крім $\sigma(5)$». Єдиним символом, який ця клітинка може законно містити, залишається потрібний. Усі обмеження знову мають форму «ці дві клітинки різні» — вони інваріантні до перейменування й перевіряються так само, як рядок. Саме такий прийом використовують для наперед пофарбованих вершин у класичному протоколі розфарбування графа; він передає й сенс слова *ґаджети* вище: у картині MegaSudoku-як-SAT підказки компілюються в ґаджети нерівності так само, як усі інші обмеження.
Фізичний протокол. Реальний картковий протокол для судоку (Gradwohl, Naor, Pinkas і Rothblum, 2007) взагалі не використовує перейменування й фіксує підказки ще до початку приховування. Для кожної клітинки Аліса викладає три однакові карти зі значенням клітинки — сорочкою догори для секретних клітинок, але **лицем догори для клітинок-підказок**, щоб Боб на власні очі бачив дотримання підказок до того, як карти перевернуть. Потім одна карта з кожної клітинки йде в пакет її рядка, одна — у пакет стовпця, одна — у пакет блока; кожен пакет перемішують і відкривають, а Боб перевіряє, що він містить усі N символів. Перемішування руйнує інформацію про позиції — це й дає zero-knowledge, — але підказки вже були зафіксовані під час розкладання.
У будь-якому варіанті урок той самий, до якого ця стаття постійно повертається: протокол zero-knowledge — це ретельний облік того, *які саме* факти переживають приховування. Перейменування зберігає «усі різні» й стирає «дорівнює 5», тому «дорівнює 5» треба повернути іншим способом.

Це не протокол із самої статті. Це лише ментальна модель класичного zero-knowledge:

- Аліса й Боб обмінюються повідомленнями.
- Боб вибирає випадкові перевірки.
- Аліса розкриває лише локальну узгодженість, а не весь розв'язок.
- Доказ приватності полягає в тому, що картину, яку бачить Боб, можна було б згенерувати без знання секретного розв'язку Аліси.

Отже, класичне zero-knowledge побудоване навколо позитивного факту:

> Симулятор справді існує.

Тепер приберімо зручні частини. Аліса надсилає один рядок-доказ і йде. Немає довіреного setup, немає спільного випадкового рядка, підготовленого наперед, а Боб ніколи не повинен приймати хибну головоломку. Саме в такому середовищі класичне zero-knowledge не виживає.

Перед самим трюком потрібен ще один персонаж. Зафіксуймо **книгу правил**: формальну систему доведення в логічному сенсі — фіксований набір аксіом плюс механічні правила перевірки письмових математичних доказів. Канонічний приклад — ZFC, стандартна система аксіом математики. Усе далі формулюється відносно наперед вибраної книги правил, але вибір гнучкий: конструкція працює для будь-якої фіксованої формальної системи, включно із ZFC.

(Термінологічна примітка, запозичена із самої статті: «proof system» тут завжди означає саме таку книгу правил — формальну систему, що перевіряє математичні докази, — а не повідомлення, які надсилає Аліса. Механізми Аліси й Боба називаються «доведувачем» і «перевірячем».)

Версія в стилі Ґеделя зберігає історію про mega-Sudoku, але змінює сам доказ.

Виберімо другу систему обмежень такого самого відображуваного розміру й назвімо її **D**. У нашій історії S і D — дві головоломки MegaSudoku(n) того самого формату. За лаштунками D могла починатися як складна логічна формула іншого розміру; за потреби її можна доповнити нешкідливими фіктивними обмеженнями, щоб вона помістилася в таку саму сітку. D будується з логічної формули, яка насправді **нездійсненна**: не існує присвоєння значень, яке задовольняє всі її обмеження, так само як зламана головоломка не має законно заповненої сітки. Іграшковий приклад — формула, що одночасно вимагає «X істинне» і «X хибне». Отже, D не має правильного заповнення.

Але D не повинна бути зламаною головоломкою, яку *легко викрити*. Іграшковий приклад не годиться: будь-яка формальна система спростує «X і не-X» одним рядком. D має бути хибною так, щоб вибрана книга правил не могла сертифікувати це коротким аргументом. Якби вона могла коротко спростувати D, вся конструкція нижче розвалилася б: альтернативний шлях, який міг би дозволити генерувати докази без секрету Аліси, формально виключався б, а разом із ним зникла б гарантія приватності. Тому D беруть із сімейства, яке фіксована книга правил не може ефективно спростувати: всередині цієї системи немає короткого доказу того, що D не має розв’язку.

Одноповідомленнєвий доказ Аліси тепер стосується диз’юнкції:

> або справжнє mega-Sudoku S має розв’язок, або приманка D має розв’язок.

Ось логічний зв’язок. D **не** генерується якимось магічним способом, що робить S істинною. Доказ не стверджує «D не має розв’язку, отже S має розв’язок». Він доводить диз’юнкцію **S або D**. Perfect soundness каже, що хибна диз’юнкція не може мати коректного доказу. Оскільки в реальності D хибна — розв’язку немає, — єдиний спосіб, щоб диз’юнкція була істинною, полягає в істинності S. Отже, якщо доказ прийнятий, S мусить мати розв’язок. Приманка не може перетворити хибну S на істинну.

Але для частини в стилі zero-knowledge запитаймо, що сталося б, якби D *мала* розв’язок. Такий розв’язок-приманка був би альтернативним witness. Він дозволив би генерувати докази, не знаючи справжнього розв’язку mega-Sudoku Аліси — тобто дав би симулятор.

У реальності D розв'язку не має, тому цей шлях симуляції закритий. Суть у тому, що книга правил не може ефективно довести, що він закритий.

Отже, D виконує дві роботи. Для **soundness** D хибна, тому коректний доказ «S або D» змушує S бути істинною. Для **effective zero-knowledge** D важко спростувати, тому книга правил не може швидко виключити шлях через приманку, який зробив би симуляцію можливою.

Тож тест безпеки вже не такий:

> Чи можемо ми довести, що симулятор справді існує?

Натомість він стає таким:

> Чи може ваша книга правил ефективно довести, що симулятор неможливий?

Якщо відповідь «ні», випливає щось несподівано сильне: кожна гарантія безпеки, яка (а) спостерігається шляхом виконання тесту й (б) усередині цієї книги правил доведено випливає з існування симулятора, насправді виконується. Успішна атака на будь-яку з них сама стала б тим відсутнім коротким спростуванням — а такого короткого спростування немає. Саме це означає «effective» в effectively zero-knowledge.

Отже, навчальне протиставлення виглядає так:

Класичне zero-knowledge: докази безпечні, бо симулятор існує.

Effective zero-knowledge у стилі Ґеделя: для спостережуваних тестів безпеки докази поводяться як безпечні, бо книга правил не може ефективно довести, що симулятор неможливий.

Друге твердження слабше. Але саме тому стаття може зберегти три властивості, які руйнували класичну версію: одне повідомлення, відсутність setup і perfect soundness.

Новий критерій: ви не можете довести відсутність симулятора

Послаблення Іланго змінює саме запитання.

Класичне zero-knowledge питає:

> Чи існує симулятор?

Effectively zero-knowledge питає щось слабше:

> Чи може вибрана вами книга правил ефективно довести, що симулятора не існує?

Це може звучати як технічна лазівка, але саме тут центральна ідея. Конструкція перебуває в дивному стані: симулятора насправді **не існує** — стаття говорить про це прямо, — але зафіксована вами книга правил не може ефективно довести його відсутність. Якщо кожен небажаний наслідок, який вас цікавить, вимагав би такого спростування, система все одно поводиться як zero-knowledge щодо цих наслідків.

Саме тут з'являється Ґедель. Не як прикраса й не в сенсі «Ґедель робить криптографію безпечною». Зв'язок теоретико-доказовий. Книгу правил називають **оптимальною**, якщо вона в точному сенсі є найкращою можливою: коли будь-яка формальна система може спростувати формулу потрібного типу коротким доказом, оптимальна теж може це зробити, причому її доказ буде щонайбільше поліноміально довшим. Крайчічек і Пудлак у 1989 році висунули гіпотезу, що **оптимальної системи доказів не існує**: яку книгу правил не зафіксуй, знайдеться інша, що доводить певне сімейство істинних тверджень набагато коротше. Це одна з центральних відкритих гіпотез теорії складності доказів і скінченний, теоретико-складнісний родич теореми Ґеделя про неповноту: деякі істинні твердження не мають короткого доказу у фіксованій вами системі — не тому, що їх у принципі неможливо довести, а тому, що кожна фіксована книга правил залишає деякі коротко сформульовані істини без коротких доказів.

Стаття припускає цю гіпотезу — у трохи сильнішій формі «нескінченно часто», стандартній для криптографічного використання гіпотез. За теоремою Крайчічека й Пудлака виграш конкретний: для кожної книги правил існує послідовність формул, які справді нездійсненні, але які ця система не може спростувати короткими доказами — і, що критично, ефективний алгоритм може їх *генерувати*. Саме ця остання властивість, однорідність (uniformity), перетворює всю ідею з твердження про існування на справжній алгоритм, який може виконувати Аліса: її приманки D сходять із конвеєра, а не виникають із повітря.

Криптографічний хід полягає в тому, щоб використати цю нестачу доказової сили.

Що робить конструкція

Ось конструкція статті в оголеному вигляді.

Зафіксуйте книгу правил — скажімо, ZFC. За припущенням із теорії складності доказів існує ефективно генерована послідовність формул, які насправді нездійсненні, але книга правил не має короткого доказу їхньої нездійсненності.

Тепер побудуйте одноповідомленнєвий доказ такого типу:

> або справжнє твердження здійсненне, або ця спеціальна складна формула здійсненна.

Спеціальна складна формула нездійсненна. Отже, якщо базовий механізм доказу має perfect soundness, прийняття повідомлення все одно означає, що справжнє твердження

істинне. Це дає perfect soundness.

Але для безпеки в стилі zero-knowledge уявімо, що спеціальна складна формула *була б* здійсненною. Тоді її witness можна було б використати для симуляції доказів без знання справжнього witness. Насправді формула нездійсненна — але книга правил не може ефективно це довести. Отже, вона не може ефективно довести, що симулятор неможливий.

Саме це є шарніром усієї конструкції. Система не приховує секрет створенням класичного симулятора. Для великого класу спостережуваних тестів безпеки вона ховає секрет за нездатністю книги правил сертифікувати відсутність симулятора.

Що стверджує стаття

Головна теорема має кілька шарів. Ядро результату таке.

За стандартного криптографічного припущення — існування **невзаємодійних доказів, у яких свідки нерозрізненні (non-interactive witness indistinguishable proofs)**, добре вивчених об'єктів, що випливають із кількох усталених наборів припущень, — і за гіпотези теорії складності доказів, що **не існує (нескінченно часто) оптимальної системи доказів**, стаття для кожної вибраної книги правил будує одноповідомленнєвий доведувач і перевіряч для NP/SAT із **perfect soundness**, без setup, який є effectively zero-knowledge відносно цієї книги правил. (NP/SAT — стандартний «найскладніший спільний знаменник» задач-головоломок; mega-Sudoku — лише один із його костюмів.)

Для ширшого твердження про збереження фальсифікованих властивостей безпеки стаття додає ще одне стандартне припущення — дерендомізаційну гіпотезу **P = BPP** (грубо кажучи: випадковість не дає алгоритмам принципової додаткової сили).

Якщо перекласти з мови теорем:

- Доказ — одне повідомлення.
- Довіреного setup немає.
- Хибні твердження неможливо довести.
- Доведувач не є класичним zero-knowledge — симулятора в нього немає.
- Але в цьому середовищі можна отримати кожен фальсифікований, ігровий наслідок безпеки класичного zero-knowledge.

Слово «фальсифікований» важливе. Воно означає, що провал безпеки можна перевірити, запустивши противника в певній грі. Багато криптографічних визначень безпеки мають саме таку форму: чи може противник розрізнити два шифротексти, обернути функцію, відновити witness або виграти конкретний експеримент? Теорема дає доведувач для кожної такої властивості окремо, по одній за раз. Один доведувач, який одночасно має *всі* фальсифіковані властивості, найімовірніше неможливий: стара атака через повторне використання («Боб може показати доказ іншим») сама

є фальсифікованою властивістю, і тут вона справді не виконується. Пропозиція статті полягає в тому, що один доведувач може правдоподібно покрити всі *природні* фальсифіковані властивості — ті, що реально зустрічаються в криптографічній практиці, — але ця частина є умовною теоремою, яка спирається на неформальне поняття «природності» плюс явну гіпотезу. Гарантія націлена на спостережувані провали, а не на кожне філософське чи симуляційне значення секретності.

Один конкретний наслідок варто назвати окремо: конструкція дає перші невзаємодійні докази з **приховуванням witness (witness hiding)** і однорідним доведувачем — «доказ існування розв'язку головоломки не допомагає вам знайти сам розв'язок», без взаємодії й без setup. Звучить скромно, але такий об'єкт не вдавалося побудувати десятиліттями.

Чого це не означає

Саме цей розділ не дає матеріалу перетворитися на гіперболу.

Це **не** означає, що старі теореми неможливості були неправильними. Конструкція обходить їх, змінивши визначення.

Це **не** дає звичайного класичного zero-knowledge без взаємодії, без setup і з perfect soundness. Стаття прямо каже, що побудований доведувач не має симулятора.

Це **не** означає, що доказ неможливо повторно використати. Одноповідомленнєвий доказ усе ще можна показати комусь іншому; стаття не зберігає властивостей на кшталт заперечуваності (deniability). (Невзаємодійне zero-knowledge з довіреним setup має ту саму межу.)

Це **не** практичний протокол, готовий до розгортання. Це теорія складності та фундаментальна криптографія. Результат залежить від великих припущень із теорії складності доказів і криптографії, а конструкція відповідає на питання про те, що принципово можливо.

І це **не** перетворює «Ґеделя» на магічний примітив безпеки. Зв'язок із Ґеделем проходить через системи доказів, оптимальні системи доказів і скінченні аналоги неповноти. Корисна інтуїція тут не «неповнота захищає ваш пароль», а така: якщо книга правил не може ефективно довести, що симулятор неможливий, то атаки, які вимагали б такого доказу, можна блокувати на рівні визначень безпеки.

Чому це все одно цікаво

Криптографія часто перетворює складність на безпеку. Факторизація складна — отже, припущення в стилі RSA стають корисними. Ґраткові задачі складні — отже, корисною стає ґраткова криптографія. Тут складність дивніша: не «важко обчислити секрет», а «важко довести, що певний об'єкт доказу не може існувати».

Саме тому робота відчувається незвично. Вона поводиться з аксіомами й книгами правил майже як із криптографічними ресурсами. Звичайна неможливість говорить про напруження між soundness і симуляцією. Хід Іланго ставить це напруження за теоретико-доказову завісу: симулятора немає, але формальна система не може ефективно викрити цю відсутність.

Для читача несподіване не те, що це замінить сучасні zero-knowledge системи. Найімовірніше, принаймні безпосередньо, не замінить. Несподіване те, що обмеження з математичної логіки можна використати конструктивно: не лише як стіну, а як своєрідне укриття.

Наскільки переконливі підстави?

Це стаття з теоремою, тому «докази» тут означають інше, ніж у біології чи астрономії. Питання не в тому, чи повторився експеримент. Питання в тому, чи підтримують визначення, припущення й ланцюжок математичного доведення заявлений результат.

Доказ формальний, а стаття чітко перераховує припущення. І вони не випадкові. Невзаємодійні witness-indistinguishable proofs — стандартні об’єкти криптографії, що випливають із кількох усталених наборів припущень. Гіпотеза про відсутність оптимальної системи доказів є центральною гіпотезою теорії складності доказів. **P = BPP** — стандартне припущення про дерендомізацію, потрібне лише для ширшої теореми про фальсифіковані властивості.

Стаття також аргументує, що ці припущення — не довільні риштування, а фактично потрібна ціна. Вона доводить обернений результат: якщо такі конструкції взагалі існують, то мають існувати невзаємодійні witness-indistinguishable proofs, а за стандартного припущення про існування односторонніх функцій оптимальної системи доказів не може існувати. І припущення мають «win-win» характер: спростування будь-якого з них саме стало б знаковим відкриттям у теорії складності доказів, криптографії або теорії складності.

Але оскільки результат умовний, упевненість у його інтерпретації теж умовна. Якщо припущення хибні, зміст теореми змінюється. І навіть якщо вони правильні, гарантія не є повним класичним zero-knowledge; це послаблена, теоретико-доказова версія статті.

Тому правильна оцінка така: висока впевненість, що стаття встановлює цілісний умовний результат про можливість; помірна — що припущення описують криптографічний світ, у якому ми насправді живемо; і низька — щодо будь-якого негайного практичного наслідку.

Чому це важливо

Стаття відкриває маршрут, який вважався закритим.

Класична теорія каже: повне zero-knowledge не може бути одним повідомленням без setup і не може мати perfect soundness. Стаття Іланго каже: якщо запитувати про ті наслідки zero-knowledge, які можна перевіряти в іграх безпеки, і дозволити визначенню безпеки залежати від того, що книга правил може або не може ефективно спростувати, то значну частину корисної поведінки можна повернути — з одним повідомленням, без setup і з perfect soundness.

Це не маленьке підкручування визначення. Це інший спосіб думати про криптографічні гарантії. Замість того щоб питати лише, що існує, запитайте, що ваша книга правил здатна виключити. Замість того щоб трактувати недовідність як філософську незручність, використайте її як структуру.

Практичний світ, можливо, не зміниться завтра. Але концептуальна карта вже змінилася. Тепер існує формальний сенс, у якому «ніхто не може ефективно довести, що секрет витік» може бути достатньо сильним, щоб відновити багато ігрових захистів, яких ми хотіли від твердження «секрет не витік».

Саме тому Ґедель — у назві.

Короткий підсумок

Докази з нульовим розголошенням дозволяють доведувачу переконати перевіряча в істинності твердження, не розкриваючи witness. Класичні результати неможливості кажуть, що zero-knowledge не можна втиснути в одне повідомлення без setup і що воно не може мати perfect soundness. Стаття Рахула Іланго не спростовує цих теорем. Вона визначає слабше поняття — effectively zero-knowledge: замість вимоги реального існування симулятора потрібно, щоб вибрана система доказів — формальна книга правил на кшталт ZFC — не могла ефективно довести, що симулятора не існує. За великих припущень із криптографії (невзаємодійні witness-indistinguishable proofs) та теорії складності доказів (відсутність оптимальної системи доказів) стаття будує одноповідомленнєві доведувачі для NP/SAT без setup і з perfect soundness, які по одній властивості за раз відтворюють фальсифіковані, ігрові наслідки zero-knowledge. Один доведувач для всіх «природних» таких властивостей — подальше, частково гіпотетичне розширення, а буквально всі фальсифіковані властивості одночасно, ймовірно, неможливі, бо докази залишаються повторно використовуваними. Результат теоретичний і умовний, а не готовий криптографічний примітив, але він показує новий спосіб перетворити теоретико-доказову недовідність на криптографічний ресурс.

Твереза оцінка

Що показує стаття: За явно сформульованих припущень можна побудувати одноповідомленнєві без-setup і perfectly sound доведувачі для NP/SAT, які є effectively zero-knowledge відносно будь-якої вибраної системи доказів і реалізують кожен фальсифікований, ігровий наслідок класичного zero-knowledge.

Що правдоподібно, але не доведено безумовно: Що потрібні припущення з теорії складності доказів і криптографії правильні. Це серйозні, добре вивчені припущення — і стаття показує, що вони по суті не лише достатні, а й необхідні, — але вони все одно лишаються припущеннями.

Чого вона не показує: Класичне zero-knowledge без взаємодії, без setup і з perfect soundness; практичну систему, готову до розгортання; заперечуваність або неможливість повторного використання доказів; або те, що сама по собі теорема Ґеделя про неповноту забезпечує криптографічну безпеку.

Головні обмеження: Гарантія є послабленням zero-knowledge; найширша версія залежить від кількох припущень; твердження про один універсальний доведувач частково лишаються гіпотетичними; а результат передусім фундаментальний.

Якої впевненості заслуговує висновок для загального читача? Високої, що це важливий умовний теоретичний результат, якщо прийняти його визначення. Помірної, що припущення відповідають реальності. Низької щодо негайного практичного застосування. Безпечний висновок: стаття не ламає теореми неможливості для zero-knowledge; вона знаходить новий теоретико-доказовий спосіб обійти ті їхні частини, які мають значення для багатьох ігор безпеки.

Джерела

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