The Clean Paper

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

Dowód kryptograficzny może ukryć sekret, utrudniając wykazanie braku symulatora

Dowody wiedzy zerowej pozwalają udowodnić zdanie bez ujawniania świadka. Klasyczna teoria mówi, że w pełnym znaczeniu nie da się tego osiągnąć jednocześnie za pomocą jednej wiadomości, bez zaufanej konfiguracji wstępnej i z doskonałą poprawnością. Praca Rahula Ilango nie usuwa tej niemożliwości. Zmienia cel: zamiast wymagać rzeczywistego istnienia symulatora, wymaga, by żaden wybrany system dowodowy nie potrafił efektywnie udowodnić, że symulator nie istnieje. Przy ważnych założeniach z teorii złożoności dowodów i kryptografii wystarcza to, by odzyskać — właściwość po właściwości — falsyfikowalne, definiowane przez gry konsekwencje wiedzy zerowej. Nie chodzi o gotowy element infrastruktury internetowej. To wywodzący się z teorii dowodów sposób przekształcenia matematycznej nieudowadnialności w kryptograficzną osłonę.

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

Sztuczka nie polega na udowodnieniu, że sekret jest ukryty

Zacznijmy od najprostszej wersji wiedzy zerowej.

Alice chce przekonać Boba, że łamigłówka Sudoku ma rozwiązanie. Jeśli je wyśle, Bob zostanie przekonany, ale zagadka straci sens. Alice chce czegoś dziwniejszego: dowodu, że rozwiązanie istnieje, lecz bez jego ujawniania.

Na tym polega obietnica dowodu wiedzy zerowej. Dowodzący (Alice) przekonuje weryfikatora (Boba), że dane zdanie jest prawdziwe, nie ujawniając niczego poza samą jego prawdziwością.

Problem w tym, że ta obietnica ma swoją cenę. Zwykły dowód matematyczny ma dwie wygodne cechy. Jest **jedną wiadomością**: zapisujesz go, przekazujesz i odchodzisz. Jest też **doskonale poprawny**: fałszywe zdanie nie ma żadnego prawidłowego dowodu. Klasyczne wyniki o niemożliwości mówią, że wiedza zerowa musi zrezygnować z obu tych cech — i nie chodzi tylko o ich połączenie; każda z osobna jest niedostępna.

Po pierwsze, dowód wiedzy zerowej wymaga rozmowy. Jeśli Alice wyśle jedną wiadomość bez zaufanej konfiguracji przygotowanej z góry, gwarancja wiedzy zerowej upada — niezależnie od tego, z jak dużej poprawności jesteśmy gotowi w zamian zrezygnować.

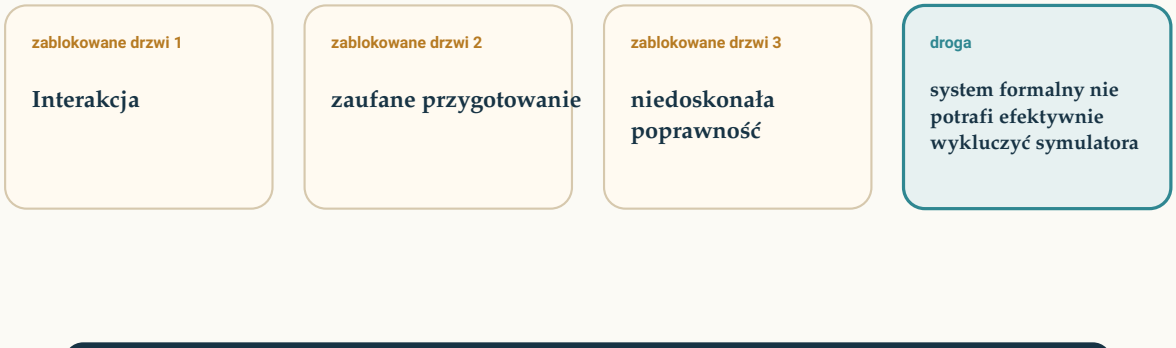
Po drugie, dowód wiedzy zerowej wymaga niewielkiej tolerancji błędu. Wymóg doskonałej poprawności po cichu niszczy także interakcję: weryfikator, którego nie da się oszukać bez względu na jego losowe wybory, równie dobrze może ustalić te wybory z góry. A gdy staje się przewidywalny, Alice może odpowiedzieć na wszystko w jednej wiadomości — czyli wracamy dokładnie do przypadku, który już okazał się niemożliwy.

Praca Rahula Ilango proponuje sposób obejścia tej podwójnej ściany. Nie udaje, że ściany nie ma, ani nie tworzy klasycznej wiedzy zerowej w warunkach, w których jest ona niemożliwa. Posunięcie jest subtelniejsze: osłabić znaczenie słów „niczego nie ujawnia”, ale tak, by zachować właściwości bezpieczeństwa, które kryptografowie potrafią rzeczywiście testować.

Rezultat nazywa się **efektywną wiedzą zerową**.

Dowód bez ujawniania świadka

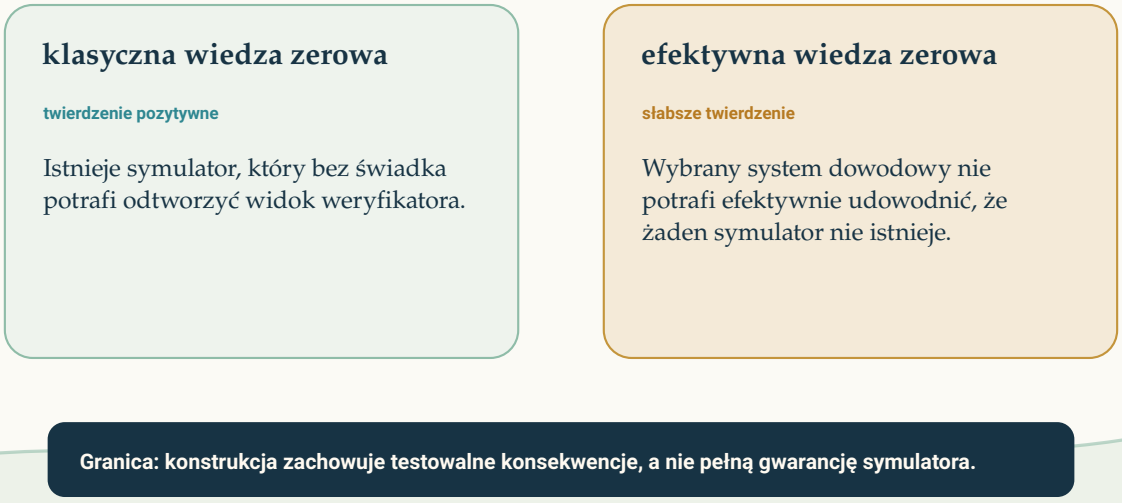
Praca zachowuje formę zwykłego dowodu, osłabiając to, co prywatność musi wykazać.



Wiedzę zerową blokują troje drzwi — interakcja, zaufana konfiguracja i niedoskonała poprawność. Konstrukcja Ilango przechodzi inną drogą: system formalny nie potrafi efektywnie wykluczyć symulatora.

Prywatność klasyczna a prywatność efektywna

Sednem jest osłabienie warunku: praca zmienia to, co musi wykazać gwarancja prywatności.



Klasyczna wiedza zerowa pyta, czy symulator istnieje; „efektywna wiedza zerowa” pyta jedynie, czy wybrany system formalny potrafi efektywnie udowodnić, że symulator istnieć nie może. To słabsze pytanie pozwala

konstrukcji zachować jedną wiadomość, brak konfiguracji wstępnej i doskonałą poprawność.

Stary test: istnieje symulator

Klasyczny sposób formalizacji wiedzy zerowej wykorzystuje fikcyjnego pomocnika zwanego **symulatorem**.

Pomysł jest taki: wyobraźmy sobie Jane, która **nie** zna sekretu Alice. Jeśli Jane potrafi samodzielnie wygenerować dowody wyglądające dokładnie jak te, które Bob otrzymałby od Alice, dowody Alice nie przekazały Bobowi niczego nowego. Jane mogła odtworzyć całe doświadczenie bez sekretu Alice.

Klasyczna wiedza zerowa wymaga rzeczywistego symulatora. Musi istnieć wydajny algorytm, który potrafi tworzyć pozornie prawdziwe dowody bez znajomości sekretu — w żargonie *świadka*; w Sudoku świadkiem jest po prostu wypełniona plansza.

Ta definicja jest mocna, ale właśnie tutaj uderza dawna niemożliwość. Intuicja jest następująca. Prawdziwie nieinteraktywny dowód to po prostu ciąg znaków. Gdy Bob go otrzyma, może pokazać go komuś innemu: zyskuje możliwość dowodzenia tego zdania innym, co już brzmi jak coś więcej niż „nic”. Klasyczne twierdzenia przekuwają tę intuicję w opisane wyżej wyniki o niemożliwości.

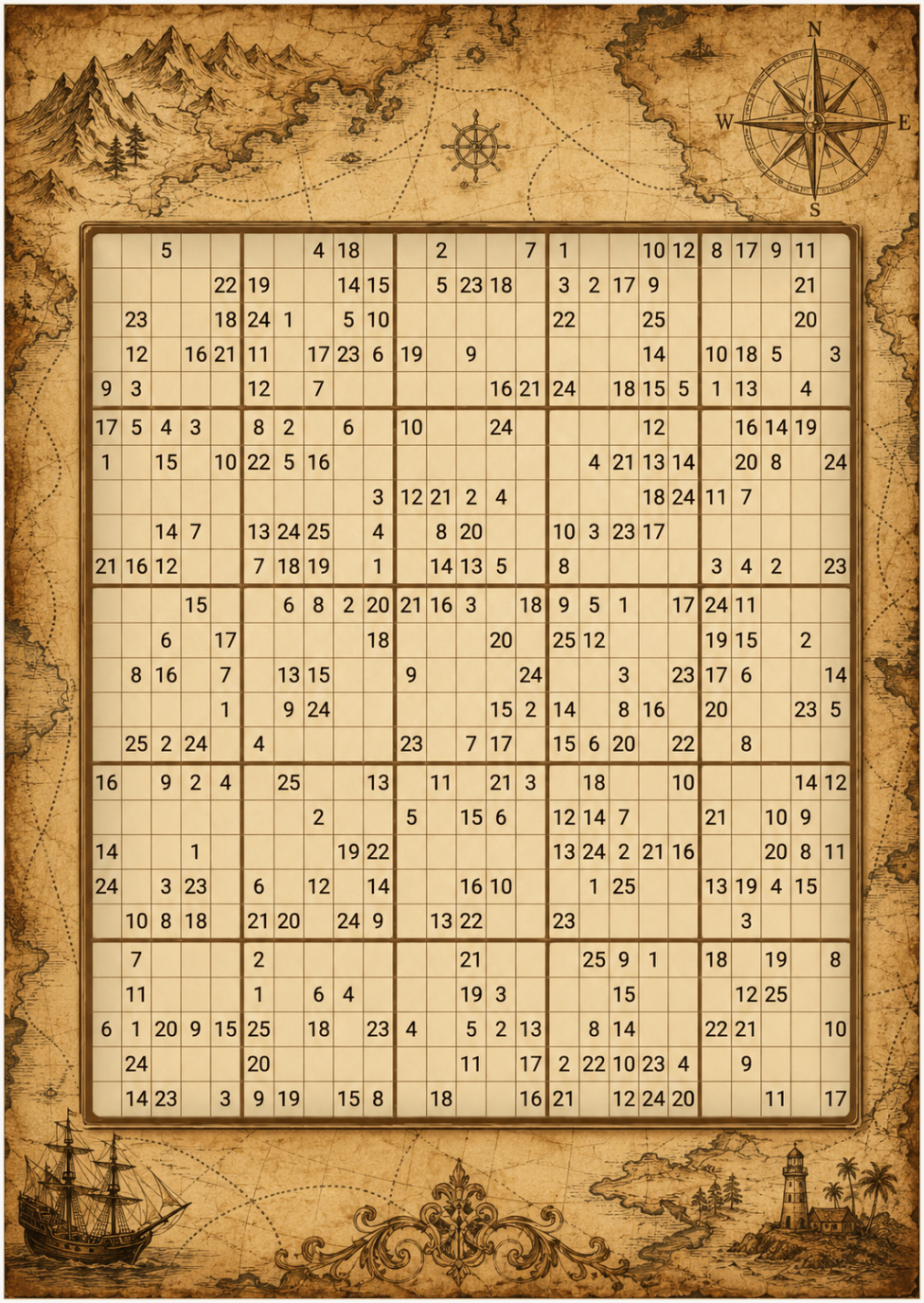
> **Trzy właściwości, na których opiera się ten artykuł**
>
> Tytuł artykułu wymienia trzy ograniczenia:
> **Brak interakcji:** Alice wysyła jeden ciąg stanowiący dowód. Nie ma protokołu wymiany wiadomości.
> **Brak konfiguracji wstępnej:** Alice i Bob nie korzystają z zaufanego wspólnego ciągu referencyjnego ani z innego publicznego źródła losowości przygotowanego z góry. Wiele systemów nazywanych „nieinteraktywną wiedzą zerową” nadal wymaga takiej konfiguracji; ta praca nie wymaga żadnej.
> **Doskonała poprawność:** fałszywe zdanie nie ma żadnego prawidłowego dowodu. Nie chodzi o to, że „niemal nigdy nie zostanie przyjęte”; prawidłowy dowód po prostu nie istnieje.
> Zwykłe zapisane dowody matematyczne mają dokładnie te trzy właściwości — a klasyczna wiedza zerowa, jak wyjaśniono wyżej, nie może ich zachować.

Różnica na przykładzie mega-Sudoku

Oto celowo uproszczony sposób, by poczuć różnicę.

W poważnej części analogii nie używajmy zwykłego Sudoku 9 na 9. Jest zbyt małe i zbyt skończone: komputer może je po prostu rozwiązać albo wykazać, że rozwiązania nie ma. Wyobraźmy sobie zamiast tego rodzinę łamigłówek **MegaSudoku(n)**. Skalujemy zwykłą regułę: wybieramy rozmiar bloku

n, przyjmujemy $N = n^2$ i budujemy planszę N na N, podzieloną na bloki n na n i wypełnianą N symbolami. Zwykłe Sudoku to tylko maleńki przypadek $n = 3$, $N = 9$: plansza 9 na 9, bloki 3 na 3 i dziewięć symboli. Opowieść o złożoności dowodów zaczyna się dopiero wtedy, gdy n może rosnąć, a plansza może zawierać dodatkowe gadżety, dzięki którym zachowuje się jak formuła SAT przebrana za Sudoku. Formuła SAT jest po prostu listą ograniczeń typu tak/nie: czy zmiennym da się przypisać wartości prawda/fałsz tak, by każde ograniczenie było spełnione?



Sudoku 25 × 25: przestrzeganie reguł można sprawdzić bez ujawniania ukończonej planszy — to wizualny odpowiednik dowodu weryfikującego ukryte rozwiązanie, czyli świadka.

Sudoku i SAT: ta sama zagadka w dwóch kostiumach

Stwierdzenie, że Sudoku może „zachowywać się jak formuła SAT", nie jest metaforą. Przekład działa w obie strony, a łatwiejszy kierunek można rozpisać w całości.

Od Sudoku do SAT. SAT operuje wyłącznie wartościami prawda/fałsz, więc wprowadzamy po jednej zmiennej logicznej dla każdej trójki (wiersz, kolumna, wartość): *x(r,c,v)* znaczy „komórka w wierszu *r* i kolumnie *c* zawiera wartość *v*". Sudoku 4 na 4 (bloki 2 na 2, wartości 1–4) wymaga 4·4·4 = 64 zmiennych; klasyczne 9 na 9 — 729. Każda reguła Sudoku staje się następnie zestawem klauzul. (Klauzula jest alternatywą zmiennych lub ich negacji; cała formuła jest koniunkcją wszystkich klauzul.)

Każda komórka zawiera co najmniej jedną wartość — jedno zdanie na komórkę:

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

Każda komórka zawiera co najwyżej jedną wartość — po jednej klauzuli „nie obie naraz" dla każdej pary wartości:

$$\neg x(1,1,1) \lor \neg x(1,1,2) \quad \neg x(1,1,1) \lor \neg x(1,1,3)$$ … i tak dalej dla wszystkich sześciu par.

Każdy wiersz zawiera każdą wartość — dla wiersza 1 i wartości 3: przynajmniej raz,

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

i co najwyżej raz: $\neg x(1,1,3) \lor \neg x(1,2,3)$, i tak dalej dla każdej pary komórek w wierszu.

Kolumny i bloki — identyczne partie; zmienia się tylko grupa komórek. Dla bloku w lewym górnym rogu i wartości 2:

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

oraz klauzule „nie obie naraz" dla każdej pary.

Wydrukowane cyfry — najprostsza część: każda podana cyfra staje się klauzulą z jedną zmienną. Wydrukowana 3 w lewym górnym rogu daje klauzulę

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

Koniunkcja tych wszystkich klauzul jest spełnialna dokładnie wtedy, gdy Sudoku ma rozwiązanie — a spełniające ją przypisanie *jest* rozwiązaniem: wystarczy odczytać, które zmienne x(r,c,v) są prawdziwe, i wypełnić planszę. Dla Sudoku 9 na 9 otrzymujemy 729 zmiennych i kilka tysięcy klauzul, z którymi współczesny solver SAT radzi sobie w ciągu milisekund. Zwróćmy uwagę na klauzulę x(1,1,3): mówi „ta komórka ma dokładnie wartość 3", a nie „te komórki mają różne wartości". Ta sama asymetria wymusi dodatkowy zabieg dla podanych cyfr w opisanym niżej protokole.

Od SAT do Sudoku. Praca potrzebuje odwrotnego, trudniejszego kierunku: dla *dowolnej* formuły SAT należy zbudować mega-Sudoku, które ma rozwiązanie dokładnie wtedy, gdy formuła jest spełnialna. Własne reguły Sudoku potrafią powiedzieć jedynie „te komórki mają różne wartości", więc dowolne ograniczenia logiczne trzeba *zbudować* — i temu służą gadżety. Gadżet jest małym, przygotowanym z góry układem komórek, po jednym

na każdą klauzulę formuły. Wybrane komórki pełnią w nim rolę zmiennych (umieszczony symbol koduje prawdę lub fałsz), a wewnętrzne ograniczenia zaprojektowano tak, by jedyne legalne wypełnienia odpowiadały przypisaniom spełniającym daną klauzulę. To standardowe rzemiosło z dowodów NP-zupełności; dla uogólnionego Sudoku konstrukcję przedstawili Yato i Seta w 2003 roku.
Oba kierunki razem pokazują, że Sudoku N na N i SAT są tym samym problemem w dwóch różnych kostiumach. To pozwala temu artykułowi — i pracy naukowej — opowiadać o całej klasie NP za pomocą plansz i symboli.

Świadka nadal łatwo sobie wyobrazić. Alice zna kompletne, prawidłowe wypełnienie mega-Sudoku. Bob chce się przekonać, że takie wypełnienie istnieje, lecz Alice nie chce go ujawniać. Jeśli wyśle całą planszę, Bob zostanie przekonany, ale sekret przepadnie.

W klasycznej wersji wiedzy zerowej Alice i Bob prowadzą interakcję. Jeden ze starych modeli myślowych wykorzystuje zakryte płytki. Alice ukrywa rozwiązaną planszę, przed każdą rundą potajemnie permutuje nazwy symboli i pozwala Bobowi sprawdzić jedno losowo wybrane ograniczenie lokalne: wiersz, kolumnę, blok albo gadżet. Jeśli odsłonięte komórki zawierają różne symbole, Bob nabiera pewności. Potem wszystko zostaje ponownie zakryte, a symbole na nowo przemianowane. (Jest jeden haczyk: podane cyfry wymagają dodatkowego zabiegu, ponieważ zmiana nazw symboli ukrywa również je. Poniższa nota wyjaśnia rozwiązania z klasycznych protokołów; do dalszej opowieści wystarczy ten uproszczony obraz.)

Jak klasyczne protokoły naprawdę radzą sobie z podanymi cyframi

Sztuczka ze zmianą nazw ma ślepy punkt. Reguły wierszy, kolumn i bloków mówią „te komórki mają różne wartości", a właściwość *wszystkie różne* przeżywa każdą zmianę nazw symboli. Podana cyfra mówi jednak „ta komórka zawiera dokładnie 5". Po przemianowaniu Bob widzi tylko $\sigma(5)$ — zamaskowany symbol — nie znając permutacji σ, więc niczego nie może sprawdzić. Bez poprawki Alice mogłaby udowodnić istnienie *jakiejś* prawidłowej planszy, całkowicie ignorując wydrukowane cyfry, a to nie dowodzi niczego o *tej* łamigłówce. Klasyczna literatura zna dwa standardowe rozwiązania.
Paleta. Do ukrytej planszy dodajemy dodatkowy wiersz N komórek — paletę. Alice wpisuje w nim symbole 1…N w stałej, publicznie znanej kolejności, po czym przemianowuje je razem z całą resztą, tak że paleta zawiera $\sigma(1)…\sigma(N)$. Losowe wyzwanie Boba zyskuje jedną możliwość. Oprócz wybrania wiersza, kolumny, bloku albo gadżetu do odsłonięcia może wybrać **paletę i jedną komórkę z podaną cyfrą**. Alice odsłania oba elementy; paleta ujawnia permutację z danej rundy, a Bob sprawdza, czy komórka zawiera dokładnie przemianowaną wersję wydrukowanej cyfry. Protokół nadal ma wiedzę zerową, ponieważ Bob poznaje tylko σ — losowane od nowa w każdej rundzie i samo w sobie bezużyteczne — oraz wartość komórki, którą znał już z łamigłówki. Żadna informacja o tajnych komórkach nie wycieka, a symulator może odtworzyć ten widok, losując σ. Poprawność wynika z tego, że oszukująca Alice zostaje przyłapana ze stałym prawdopodobieństwem w każdej rundzie, a rundy powtarza się, aż wątpliwość stanie się pomijalna.

Wbudowanie podanych cyfr w ograniczenia. Wariant bardziej strukturalny usuwa specjalne wyzwanie, zamiast je dodawać. Zamiast *sprawdzać* podaną wartość, wymusza ją ograniczeniami nierówności: łączy daną komórkę ze wszystkimi komórkami palety oprócz tej niosącej właściwą wartość — „różna od $\sigma(1)$, różna od $\sigma(2)$, …, różna od wszystkiego poza $\sigma(5)$". Jedynym symbolem, który można legalnie umieścić w komórce, pozostaje symbol wskazany przez łamigłówkę. Każde ograniczenie znów ma postać „te dwa symbole są różne" — niezmienną przy przemianowaniu i sprawdzalną tak samo jak wiersz. Ten sam zabieg stosuje się do wstępnie pokolorowanych wierzchołków w klasycznym protokole kolorowania grafu; oddaje on też sens opisanych wyżej *gadżetów*. W obrazie MegaSudoku jako SAT podane cyfry, podobnie jak każde inne ograniczenie, zostają wbudowane w gadżety nierówności.
Protokół fizyczny. Rzeczywisty protokół karciany dla Sudoku (Gradwohl, Naor, Pinkas i Rothblum, 2007) w ogóle nie zmienia nazw i rozstrzyga kwestię podanych cyfr, zanim rozpocznie się ukrywanie. Dla każdej komórki Alice wykłada trzy jednakowe karty z jej wartością — zakryte dla tajnych komórek, lecz **odkryte dla komórek z podanymi cyframi**. Bob widzi więc na własne oczy, że cyfry są przestrzegane, zanim karty zostaną odwrócone. Następnie po jednej karcie z każdej komórki trafia do pakietu odpowiadającego jej wierszowi, kolumnie i blokowi; każdy pakiet zostaje potasowany i odsłonięty, a Bob sprawdza, czy zawiera wszystkie N symboli. Tasowanie usuwa informację o położeniu — to właśnie zapewnia wiedzę zerową — lecz poprawność podanych cyfr ustalono już podczas wykładania kart.
W obu przypadkach lekcja jest ta sama i artykuł stale do niej wraca: protokół wiedzy zerowej wymaga starannego rozliczenia, *które* fakty przetrwają ukrywanie. Przemianowanie zachowuje „wszystkie różne", ale wymazuje „równa się 5" — dlatego informację „równa się 5" trzeba przywrócić innymi środkami.

Nie jest to protokół z omawianej pracy. To model myślowy klasycznej wiedzy zerowej:

- Alice i Bob wymieniają wiadomości.
- Bob losowo wybiera sprawdzane ograniczenia.
- Alice ujawnia tylko lokalną spójność, nie całe rozwiązanie.
- Dowód prywatności pokazuje, że widok Boba można było wygenerować bez tajnego rozwiązania Alice.

Klasyczna wiedza zerowa opiera się zatem na pozytywnym fakcie:

> Symulator naprawdę istnieje.

Usuńmy teraz wygodne elementy. Alice wysyła jeden ciąg stanowiący dowód i odchodzi. Nie ma zaufanej konfiguracji, nie ma przygotowanego wcześniej wspólnego ciągu losowego, a Bob nigdy nie może przyjąć fałszywego twierdzenia. W takich warunkach klasyczna wiedza zerowa nie może przetrwać.

Przed właściwą sztuczką potrzebujemy jeszcze jednego bohatera. Ustalmy **zbiór reguł**: formalny system dowodowy w znaczeniu logicznym — stały zestaw aksjomatów i mechanicznych zasad sprawdzania zapisanych dowodów matematycznych. Kanonicznym przykładem jest ZFC, standardowy system

aksjomatów matematyki. Wszystko dalej określa się względem wybranego z góry systemu, przy czym wybór jest dowolny: konstrukcja działa dla każdego ustalonego systemu, w tym ZFC.

(Uwaga terminologiczna, zaczerpnięta z samej pracy: „system dowodowy" oznacza tu zawsze ten zbiór reguł — formalny system sprawdzający dowody matematyczne — nigdy wiadomości wysyłane przez Alice. Mechanizmy używane przez Alice i Boba nazywamy „dowodzącym i weryfikatorem".)

Wersja w stylu Gödela zachowuje historię mega-Sudoku, ale zmienia dowód.

Wybierzmy drugi układ ograniczeń o takim samym przedstawionym rozmiarze i nazwijmy go **D**. W naszej opowieści S i D są dwiema łamigłówkami MegaSudoku(n) w tym samym formacie. W rzeczywistości D mogło zacząć jako trudna formuła logiczna innego rozmiaru; w razie potrzeby można ją uzupełnić nieszkodliwymi, sztucznymi ograniczeniami, by pasowała do tej samej planszy. D powstaje z formuły logicznej, która jest naprawdę **niespełnialna**: nie istnieje przypisanie wartości czyniące wszystkie jej ograniczenia prawdziwymi, tak jak wadliwa łamigłówka nie ma prawidłowo ukończonej planszy. Prostym przykładem byłaby formuła wymagająca jednocześnie, by „X było prawdziwe" i „X było fałszywe". D nie ma więc prawidłowego wypełnienia.

D nie może jednak być wadliwą łamigłówką, której wadę *łatwo wykazać*. Prosty przykład powyżej się nie nadaje: każdy system formalny obala „X i nie-X" w jednym kroku. D musi być fałszywe w sposób, którego wybrany system nie potrafi poświadczyć krótkim argumentem. Gdyby system mógł obalić D krótkim dowodem, dalsza opowieść by się rozpadła: można byłoby formalnie wykluczyć alternatywną drogę tworzenia dowodów bez sekretu Alice, a wraz z nią gwarancję prywatności. D wybiera się więc z rodziny formuł, których ustalony system nie potrafi efektywnie obalić: w ramach tego systemu nie ma krótkiego dowodu, że D nie ma rozwiązania.

Jednowiadomościowy dowód Alice dotyczy zatem alternatywy:

> albo prawdziwy mega-Sudoku S ma rozwiązanie, albo przynęta D je ma.

To właśnie jest logiczne połączenie. D **nie** powstaje w magiczny sposób, który czyni S prawdziwym. Argument nie brzmi: „D nie ma rozwiązania, więc S ma rozwiązanie". Dowód dotyczy alternatywy **S lub D**. Doskonała poprawność oznacza, że fałszywa alternatywa nie może mieć prawidłowego dowodu. Ponieważ w rzeczywistości D jest fałszywe — nie ma rozwiązania — alternatywa może być prawdziwa tylko wtedy, gdy prawdziwe jest S. Jeżeli dowód zostanie przyjęty, S musi więc mieć rozwiązanie. Formuła pozorna nie może uczynić fałszywego S prawdziwym.

W części przypominającej wiedzę zerową pytamy jednak, co stałoby się, gdyby D *miało* rozwiązanie. Rozwiązanie formuły pozornej byłoby alternatywnym świadkiem. Pozwalałoby tworzyć dowody bez znajomości prawdziwego rozwiązania mega-Sudoku Alice — innymi słowy, dawałoby symulator. W rzeczywistości D nie ma rozwiązania, więc ta droga do symulatora jest zamknięta. Sedno polega na tym, że system formalny nie potrafi efektywnie wykazać jej zamknięcia.

D pełni zatem dwie funkcje. Dla **poprawności** jest fałszywe, więc prawidłowy dowód „S lub D" wymusza prawdziwość S. Dla **efektywnej wiedzy zerowej** jest trudne do obalenia, dlatego system formalny nie może szybko wykluczyć drogi przez formułę pozorną, która umożliwiłaby symulację.

Więc test bezpieczeństwa już nie jest:

> Czy możemy udowodnić, że symulator naprawdę istnieje?

Zamiast niego pojawia się pytanie:

> Czy twój system formalny potrafi efektywnie udowodnić, że symulator nie może istnieć?

Jeśli odpowiedź brzmi „nie”, wynika z tego coś zaskakująco mocnego: obowiązuje każda gwarancja bezpieczeństwa, którą (a) można zaobserwować za pomocą testu i (b) w ramach tego systemu formalnego da się wyprowadzić z istnienia symulatora. Skuteczny atak na którąkolwiek z tych gwarancji sam stanowiłby brakujące krótkie obalenie — a takiego obalenia nie ma. To właśnie „efektywna” część efektywnej wiedzy zerowej.

Kontrast w klasie jest więc następujący:

Klasyczna wiedza zerowa: dowody są bezpieczne, ponieważ istnieje symulator.

Efektywna wiedza zerowa w stylu Gödla: dowody uznaje się za bezpieczne względem obserwowalnych testów bezpieczeństwa, ponieważ system formalny nie potrafi efektywnie udowodnić, że symulator jest niemożliwy.

Drugie twierdzenie jest słabsze. Właśnie dlatego praca może zachować trzy cechy nieosiągalne w wersji klasycznej: jedną wiadomość, brak konfiguracji wstępnej i doskonałą poprawność.

Nowy test: nie można udowodnić braku symulatora

Osłabienie zaproponowane przez Ilango zmienia pytanie.

Klasyczna wiedza zerowa pyta:

> Czy istnieje symulator?

Efektywna wiedza zerowa pyta o coś słabszego:

> Czy wybrany system formalny potrafi efektywnie udowodnić, że żaden symulator nie istnieje?

Może to brzmieć jak techniczny wybieg, lecz właśnie w tym tkwi sedno. Konstrukcja znajduje się w osobliwym stanie: symulator rzeczywiście nie istnieje — praca mówi o tym wprost — ale ustalony system formalny nie potrafi efektywnie udowodnić jego nieistnienia. Jeśli każda niepożądana konsekwencja, na której nam zależy, wymagałaby takiego obalenia, system nadal zachowuje się wobec tych konsekwencji jak wiedza zerowa.

Tutaj pojawia się Gödel. Nie jako ozdoba ani jako hasło „Gödel zabezpiecza kryptografię”. Związek dotyczy teorii dowodów. System dowodowy nazywa się **optymalnym**, jeśli w precyzyjnym sensie

jest najlepszy z możliwych: gdy dowolny inny system potrafi krótko obalić formułę odpowiedniego rodzaju, system optymalny potrafi zrobić to samo dowodem co najwyżej wielomianowo dłuższym. Krajíček i Pudlák postawili w 1989 roku hipotezę, że **optymalny system dowodowy nie istnieje**: niezależnie od ustalonego systemu znajdzie się inny, który pewną rodzinę prawdziwych zdań dowodzi znacznie zwięźlej. Jest to jedna z głównych otwartych hipotez teorii złożoności dowodów i skończony, teoriokompleksyjny kuzyn twierdzenia Gödla o niezupełności. Niektóre prawdziwe zdania nie mają krótkiego dowodu w ustalonym systemie — nie dlatego, że zasadniczo nie da się ich udowodnić, lecz dlatego, że każdy stały system pozostawia pewne krótkie prawdy bez krótkich dowodów.

Praca zakłada tę hipotezę w nieco mocniejszej postaci „nieskończenie często", standardowej w kryptograficznych zastosowaniach hipotez. Dzięki twierdzeniu Krajíčeka i Pudláka otrzymujemy konkretny skutek: dla każdego systemu dowodowego istnieje ciąg naprawdę niespełnialnych formuł, których system ten nie potrafi obalić krótkimi dowodami — i, co kluczowe, potrafi je *generować* wydajny algorytm. Ta ostatnia właściwość, jednostajność, zamienia twierdzenie o istnieniu w rzeczywisty algorytm dostępny Alice: jej formuły D schodzą z taśmy produkcyjnej, nie pojawiają się znikąd.

Kryptograficzne posunięcie polega na wykorzystaniu tego niedostatku mocy dowodowej.

Co robi konstrukcja

Oto konstrukcja z pracy, sprowadzona do zasadniczego kształtu.

Ustalmy system formalny — na przykład ZFC. Przy założeniu z teorii złożoności dowodów istnieje wydajnie generowany ciąg formuł, które rzeczywiście są niespełnialne, lecz wybrany system nie ma krótkiego dowodu ich niespełnialności.

Teraz zbuduj dowód w jednej wiadomości tej formy:

> albo rzeczywiste zdanie jest spełnialne, albo ta specjalna, twarda formuła jest spełnialna.

Specjalna trudna formuła nie jest spełnialna. Jeśli zatem mechanizm dowodowy ma doskonałą poprawność, przyjęcie wiadomości nadal oznacza, że prawdziwe jest właściwe zdanie. To daje doskonałą poprawność.

Dla gwarancji przypominającej wiedzę zerową wyobraźmy sobie jednak, że ta szczególna trudna formuła *byłaby* spełnialna. Jej świadek pozwalałby symulować dowody bez znajomości właściwego świadka. W rzeczywistości formuła jest niespełnialna — lecz system formalny nie potrafi efektywnie tego udowodnić. Nie potrafi więc efektywnie wykazać, że symulator jest niemożliwy.

To punkt podparcia całej konstrukcji. System nie ukrywa sekretu przez stworzenie klasycznego symulatora. Dla szerokiej klasy obserwowalnych testów bezpieczeństwa chowa go za niezdolnością systemu formalnego do wykazania braku symulatora.

Co twierdzi artykuł

Główne twierdzenie ma kilka warstw. Jego rdzeń brzmi następująco:

Przy standardowym założeniu kryptograficznym — istnieniu **nieinteraktywnych dowodów nierozróżnialnych ze względu na świadka**, dobrze zbadanych obiektów wynikających z kilku uznanych zestawów założeń — oraz przy hipotezie z teorii złożoności dowodów, że **nie istnieje (nieskończenie często) optymalny system dowodowy**, praca konstruuje dla każdego wybranego systemu formalnego jednowiadomościowego dowodzącego i weryfikator dla NP/SAT. Konstrukcja nie wymaga konfiguracji wstępnej, ma **doskonałą poprawność** i jest efektywnie wiedzy zerowej względem tego systemu. (NP/SAT to standardowy „najtrudniejszy wspólny mianownik" problemów przypominających łamigłówki; mega-Sudoku jest jednym z jego kostiumów.)

Dla szerszego twierdzenia o zachowaniu falsyfikowalnych właściwości bezpieczeństwa praca dodaje jeszcze jedno standardowe założenie: derandomizacyjną hipotezę **P = BPP** (w przybliżeniu: losowość nie daje algorytmom istotnej dodatkowej mocy).

Przetłumaczone z języka twierdzeń:

- Dowód jest jedną wiadomością.
- Nie ma zaufanej konfiguracji wstępnej.
- Fałszywych zdań nie da się udowodnić.
- Dowodzący nie ma klasycznej wiedzy zerowej — nie istnieje dla niego symulator.
- Można jednak uzyskać każdą falsyfikowalną, definiowaną przez grę konsekwencję bezpieczeństwa klasycznej wiedzy zerowej.

Słowo „falsyfikowalne" ma znaczenie. Oznacza, że naruszenie bezpieczeństwa można sprawdzić, uruchamiając przeciwnika w grze. Wiele kryptograficznych definicji bezpieczeństwa ma właśnie taką postać: czy przeciwnik potrafi odróżnić dwa szyfrogramy, odwrócić funkcję, odzyskać świadka albo wygrać określony eksperyment? Twierdzenie daje osobnego dowodzącego dla każdej falsyfikowalnej właściwości. Jeden dowodzący mający *wszystkie* takie właściwości naraz jest prawdopodobnie niemożliwy — dawny atak przez ponowne użycie („Bob może pokazać dowód innym") sam jest właściwością falsyfikowalną i rzeczywiście tutaj się udaje. Autor proponuje, że jeden dowodzący może prawdopodobnie obejmować wszystkie *naturalne* właściwości falsyfikowalne, czyli te rzeczywiście spotykane w praktyce kryptograficznej. Ta część jest jednak twierdzeniem warunkowym, opartym na nieformalnym pojęciu „naturalności" i na jawnej hipotezie. Gwarancja dotyczy obserwowalnych naruszeń, nie każdego filozoficznego lub symulacyjnego znaczenia tajemnicy.

Warto wymienić jeden konkretny wniosek: konstrukcja daje pierwsze nieinteraktywne dowody *ukrywające świadka* z jednostajnym dowodzącym — „dowód rozwiązania łamigłówki nie pomaga tego rozwiązania znaleźć", bez interakcji i bez konfiguracji wstępnej. Ten skromnie brzmiący obiekt przez dziesięciolecia opierał się konstrukcji.

Czego praca nie mówi

Ta część pozwala zachować uczciwe proporcje.

Praca **nie** mówi, że dawne twierdzenia o niemożliwości były błędne. Konstrukcja omija je, zmieniając definicję.

Nie daje zwykłej, klasycznej wiedzy zerowej bez interakcji i konfiguracji wstępnej, a zarazem z doskonałą poprawnością. Autor wyraźnie pisze, że skonstruowany dowodzący nie ma symulatora.

Nie oznacza, że dowodu nie da się ponownie użyć. Jednowiadomościowy dowód nadal można pokazać komuś innemu; praca nie zachowuje właściwości takich jak zaprzeczalność. (Nieinteraktywna wiedza zerowa z zaufaną konfiguracją ma to samo ograniczenie.)

Nie oznacza, że jest to praktyczny protokół gotowy do wdrożenia. To teoria złożoności i podstawy kryptografii. Wynik zależy od ważnych założeń z teorii złożoności dowodów i kryptografii, a konstrukcja dotyczy tego, co jest możliwe co do zasady.

Nie czyni „Gödla" magicznym prymitywem bezpieczeństwa. Związek z Gödlem prowadzi przez systemy dowodowe, ich optymalność i skończone odpowiedniki niezupełności. Użyteczna intuicja nie brzmi „niezupełność chroni twoje hasło", lecz: jeśli system formalny nie potrafi efektywnie udowodnić, że symulator jest niemożliwy, ataki wymagające takiego dowodu mogą zostać zablokowane na poziomie definicji bezpieczeństwa.

Dlaczego to w ogóle jest interesujące

Kryptografia często zamienia trudność w bezpieczeństwo. Rozkład na czynniki jest trudny, więc użyteczne stają się założenia w rodzaju RSA. Trudne są problemy kratowe, więc użyteczna staje się kryptografia oparta na kratach. Tutaj trudność ma dziwniejszą postać: nie „trudno obliczyć sekret", lecz „trudno udowodnić, że pewien obiekt dowodowy nie może istnieć".

Dlatego praca wydaje się tak nietypowa. Traktuje aksjomaty i systemy formalne niemal jak zasoby kryptograficzne. Zwykły wynik o niemożliwości wskazuje napięcie między poprawnością a symulacją. Ilango przesuwa je za kurtynę teorii dowodów: symulatora nie ma, lecz system formalny nie potrafi efektywnie ujawnić jego braku.

Dla czytelnika zaskoczeniem nie powinno być to, że konstrukcja zastąpi dzisiejsze systemy wiedzy zerowej. Prawdopodobnie tego nie zrobi, przynajmniej nie bezpośrednio. Zaskakuje raczej możliwość konstruktywnego wykorzystania ograniczenia logiki matematycznej: nie tylko jako ściany, lecz także jako rodzaju osłony.

Jak mocne są dowody?

To praca teoretyczna, więc „dowody” znaczą tu coś innego niż w biologii czy astronomii. Nie pytamy, czy eksperyment został powtórzony. Pytamy, czy definicje, założenia i łańcuch rozumowania wspierają twierdzenie.

Dowód jest formalny, a praca jasno przedstawia założenia. Nie są one przypadkowe. Nieinteraktywne dowody nierozróżnialne ze względu na świadka są standardowymi obiektami kryptograficznymi i wynikają z kilku uznanych zestawów założeń. Nieistnienie optymalnego systemu dowodowego jest centralną hipotezą teorii złożoności dowodów. P = BPP to standardowa hipoteza derandomizacyjna, użyta tylko w szerszym twierdzeniu o właściwościach falsyfikowalnych.

Autor argumentuje również, że założenia są właściwą ceną, a nie arbitralnym rusztowaniem: dowodzi twierdzenia odwrotnego, pokazując ich zasadniczą konieczność. Jeśli takie konstrukcje w ogóle istnieją, muszą istnieć nieinteraktywne dowody nierozróżnialne ze względu na świadka, a przy standardowym założeniu o funkcjach jednokierunkowych nie może istnieć optymalny system dowodowy. Założenia mają też charakter „win-win”: obalenie któregokolwiek byłoby samo w sobie przełomem w teorii złożoności dowodów, kryptografii albo teorii złożoności.

Ponieważ wynik jest warunkowy, warunkowa jest również związana z nim pewność. Jeśli założenia okażą się fałszywe, zmieni się interpretacja twierdzenia. Nawet jeśli są prawdziwe, gwarancja nie jest pełną klasyczną wiedzą zerową; pozostaje osłabionym pojęciem zdefiniowanym w pracy.

Właściwa ocena to zatem: wysoka pewność, że praca ustanawia spójny warunkowy wynik o możliwości; umiarkowana, że jej założenia opisują świat kryptograficzny, w którym rzeczywiście żyjemy; oraz niska w odniesieniu do bezpośrednich konsekwencji praktycznych.

Dlaczego to ma znaczenie

Praca otwiera drogę, która miała być zamknięta.

Klasyczna teoria mówi: pełnej wiedzy zerowej nie da się zawrzeć w jednej wiadomości bez konfiguracji wstępnej ani połączyć z doskonałą poprawnością. Praca Ilango odpowiada: jeśli poprosimy o te konsekwencje wiedzy zerowej, które można testować w grach bezpieczeństwa, i pozwolimy, by definicja bezpieczeństwa zależała od tego, co system formalny potrafi efektywnie obalić, można odzyskać znaczną część użytecznego zachowania — z jedną wiadomością, bez konfiguracji wstępnej i z doskonałą poprawnością.

Nie jest to drobna korekta definicji. To inny sposób myślenia o gwarancjach kryptograficznych. Zamiast pytać wyłącznie, co istnieje, pytamy, co system formalny potrafi wykluczyć. Zamiast traktować nieudowadnialność jako filozoficzną niedogodność, wykorzystujemy ją jako strukturę.

Praktyczny świat nie zmieni się zapewne jutro. Zmienia się jednak mapa pojęciowa. Mamy teraz formalne znaczenie, w którym „nikt nie potrafi efektywnie udowodnić, że sekret wyciekł” może wystarczać do odzyskania wielu zabezpieczeń definiowanych przez gry, których oczekiwaliśmy od

stwierdzenia „sekret nie wyciekł”.

Dlatego Gödel powinien znaleźć się w tytule.

Zwięzłe podsumowanie

Dowody wiedzy zerowej pozwalają dowodzącemu przekonać weryfikatora o prawdziwości zdania bez ujawniania świadka. Klasyczne wyniki o niemożliwości mówią, że wiedzy zerowej nie da się zawrzeć w jednej wiadomości bez konfiguracji wstępnej ani nadać jej doskonałej poprawności. Praca Rahula Ilango nie obala tych wyników. Definiuje słabsze pojęcie — efektywną wiedzę zerową: zamiast wymagać rzeczywistego istnienia symulatora, wymaga, by wybrany system dowodowy, formalny zbiór reguł taki jak ZFC, nie potrafił efektywnie udowodnić, że symulator nie istnieje. Przy ważnych założeniach z kryptografii (nieinteraktywne dowody nierozróżnialne ze względu na świadka) i teorii złożoności dowodów (nieistnienie optymalnego systemu dowodowego) praca konstruuje jednowiadomościowych dowodzących dla NP/SAT, bez konfiguracji wstępnej i z doskonałą poprawnością, którzy uzyskują — właściwość po właściwości — falsyfikowalne, definiowane przez gry konsekwencje wiedzy zerowej. Jeden dowodzący obejmujący wszystkie takie „naturalne” właściwości to dalsze, częściowo hipotetyczne rozszerzenie; objęcie dosłownie każdej właściwości falsyfikowalnej jest prawdopodobnie niemożliwe, bo dowody nadal można przekazywać dalej. Wynik jest teoretyczny i warunkowy, nie jest gotowym prymitywem, lecz pokazuje nowy sposób wykorzystania nieudowadnialności w systemach formalnych jako zasobu kryptograficznego.

Kontrola bez upiększeń

Co pokazuje praca: Przy podanych założeniach można zbudować dla NP/SAT jednowiadomościowych dowodzących, bez konfiguracji wstępnej i z doskonałą poprawnością, którzy są efektywnie wiedzy zerowej względem dowolnego wybranego systemu dowodowego i uzyskują każdą falsyfikowalną, definiowaną przez grę konsekwencję klasycznej wiedzy zerowej.

Co jest prawdopodobne, lecz nieudowodnione bezwarunkowo: Że potrzebne założenia z teorii złożoności dowodów i kryptografii są prawdziwe. To poważne, dobrze zbadane założenia — a praca pokazuje, że są zasadniczo konieczne i wystarczające — ale nadal tylko założenia.

Czego nie pokazuje: Klasycznej wiedzy zerowej bez interakcji i konfiguracji wstępnej, a zarazem z doskonałą poprawnością; praktycznego systemu gotowego do wdrożenia; zaprzeczalności ani niemożności ponownego użycia dowodów; ani tego, że samo twierdzenie Gödla o niezupełności zabezpiecza kryptografię.

Główne ograniczenia: Gwarancja jest osłabieniem wiedzy zerowej; najszersza wersja opiera się na kilku założeniach; twierdzenia o jednym uniwersalnym dowodzącym pozostają częściowo hipotetyczne; wynik ma przede wszystkim znaczenie fundamentalne.

Jaką pewność powinien mieć czytelnik niespecjalista? Wysoką, że po przyjęciu definicji jest to

ważny warunkowy wynik teoretyczny. Umiarkowaną, że założenia trafnie opisują rzeczywistość. Niską co do bezpośredniego praktycznego wdrożenia. Bezpieczny wniosek brzmi: praca nie przełamuje wyników o niemożliwości wiedzy zerowej; znajduje nową, wywodzącą się z teorii dowodów drogę wokół tych ich części, które mają znaczenie w wielu grach bezpieczeństwa.

Źródła

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