The Clean Paper

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

Sebuah bukti kriptografis dapat menyembunyikan rahasianya dengan membuat ketiadaan simulator sulit dibuktikan

Zero-knowledge proofs memungkinkan seseorang membuktikan sebuah pernyataan tanpa mengungkap witness. Teori klasik mengatakan kita tidak dapat memiliki itu, dalam arti penuh, dengan satu pesan, tanpa trusted setup, dan perfect soundness. Makalah Rahul Ilango tidak membuat ketidakmungkinan itu lenyap. Ia mengubah target: alih-alih mensyaratkan simulator benar-benar ada, ia meminta agar tidak ada proof system pilihan yang dapat secara efisien membuktikan bahwa simulator tidak ada. Di bawah asumsi besar dari proof complexity dan kriptografi, hal itu cukup untuk memulihkan konsekuensi zero-knowledge yang falsifiable dan berbasis game, sifat demi sifat. Intinya bukan primitif internet siap pasang. Ini adalah cara proof-theoretic untuk mengubah ketidakdapatdibuktian matematis menjadi perlindungan kriptografis.

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

Triknya bukan membuktikan bahwa rahasianya tersembunyi

Mulailah dari versi paling sederhana dari zero-knowledge.

Alice ingin meyakinkan Bob bahwa sebuah teka-teki Sudoku memiliki solusi. Jika ia mengirimkan solusinya, Bob akan yakin, tetapi teka-tekinya rusak. Yang Alice inginkan lebih aneh: bukti bahwa sebuah solusi ada, tanpa mengungkapkan solusinya.

Itulah janji sebuah zero-knowledge proof. Prover (Alice) meyakinkan verifier (Bob) bahwa suatu pernyataan benar sambil tidak mengungkapkan apa pun selain kebenaran pernyataan tersebut.

Masalahnya, janji ini ada harganya. Bukti matematika biasa memiliki dua sifat yang nyaman. Ia adalah **satu pesan**: Anda menuliskannya, menyerahkannya, lalu selesai. Dan ia memiliki **perfect soundness**: pernyataan yang salah sama sekali tidak memiliki bukti yang valid. Hasil-hasil ketidakmungkinan klasik mengatakan bahwa zero-knowledge harus melepaskan kedua sifat ini — dan bukan hanya keduanya sekaligus; masing-masing pun tidak dapat dipertahankan sendiri.

Pertama, zero-knowledge proof membutuhkan percakapan. Jika Alice hanya mengirim satu pesan, tanpa trusted setup yang diatur sebelumnya, jaminan zero-knowledge runtuh — berapa pun soundness yang bersedia Anda korbankan sebagai gantinya.

Kedua, zero-knowledge proof membutuhkan sedikit toleransi terhadap kesalahan. Menuntut perfect soundness ternyata diam-diam juga menghancurkan interaksi: verifier yang tidak pernah dapat ditipu, apa pun pilihan acaknya, pada dasarnya dapat menetapkan pilihan tersebut sejak awal — dan begitu verifier dapat diprediksi, Alice dapat menjawab semuanya dalam satu pesan, persis kasus yang sudah gagal sebelumnya.

Makalah Rahul Ilango membahas cara melewati dinding ganda itu. Bukan dengan berpura-pura dindingnya tidak ada, dan bukan dengan menghasilkan zero-knowledge klasik dalam situasi yang mustahil. Langkahnya lebih halus: melemahkan arti “tidak mengungkapkan apa pun”, tetapi dengan cara yang tetap mempertahankan sifat keamanan yang benar-benar dapat diuji oleh kriptografer.

Hasilnya disebut **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 terhalang di tiga pintu — interaksi, trusted setup, dan soundness yang tidak sempurna. Konstruksi Ilango melewati pintu lain: rulebook tidak dapat secara efisien membantah keberadaan simulator.

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.

Zero-knowledge klasik bertanya apakah simulator benar-benar ada; “effectively zero-knowledge” hanya bertanya apakah rulebook pilihan Anda dapat secara efisien membuktikan bahwa simulator tidak ada.

Pertanyaan yang lebih lemah itulah yang memungkinkan konstruksi mempertahankan satu pesan, tanpa setup, dan perfect soundness.

Original diagram — The Clean Paper CC BY 4.0

Pengujian lama: sebuah simulator benar-benar ada

Cara klasik untuk memformalkan zero-knowledge menggunakan pembantu fiktif bernama **simulator**.

Gagasannya begini: bayangkan Jane, yang **tidak** mengetahui rahasia Alice. Jika Jane dapat menghasilkan sendiri bukti-bukti yang tampak persis seperti bukti yang akan diterima Bob dari Alice, maka bukti Alice tidak mengajarkan sesuatu yang baru kepada Bob. Jane sudah dapat memalsukan pengalaman itu tanpa rahasia Alice.

Jadi zero-knowledge klasik menuntut adanya simulator sungguhan. Harus ada algoritme efisien yang dapat menghasilkan bukti palsu yang tampak asli tanpa mengetahui rahasianya — dalam jargon disebut *witness*; untuk Sudoku, witness hanyalah grid yang sudah terisi dengan benar.

Definisi itu kuat, tetapi justru di situlah ketidakmungkinan lama menggigit. Intuisinya begini. Bukti yang benar-benar non-interaktif hanyalah sebuah string. Setelah Bob memiliki string tersebut, ia dapat menunjukkannya kepada orang lain: ia memperoleh kemampuan untuk membuktikan pernyataan itu kepada pihak lain, yang sudah terdengar seperti lebih dari "tidak mendapat apa-apa." Teorema klasik mempertajam intuisi ini menjadi hasil ketidakmungkinan di atas.

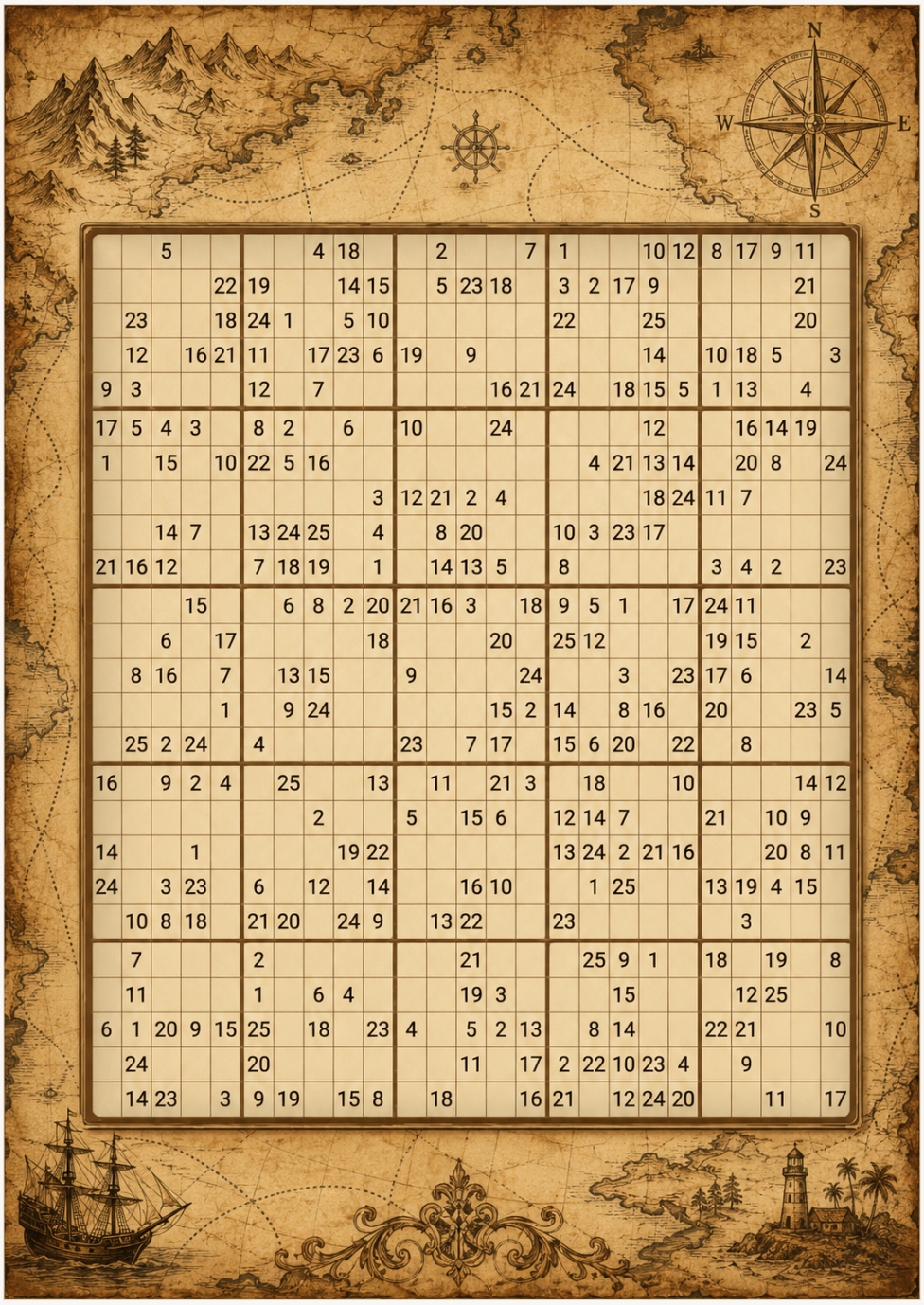
> **Tiga sifat yang dipertahankan makalah ini**
>
> Judul makalah menyebut tiga kendala:
> **Tanpa interaksi:** Alice mengirim satu string bukti. Tidak ada protokol bolak-balik.
> **Tanpa setup:** Alice dan Bob tidak bergantung pada common reference string tepercaya atau sumber randomness publik lain yang diatur sebelumnya. Banyak sistem yang disebut "non-interactive zero-knowledge" tetap bergantung pada setup; makalah ini benar-benar berarti tanpa setup.
> **Perfect soundness:** pernyataan yang salah tidak memiliki bukti valid. Bukan "hampir tidak pernah diterima"; bukti valid memang tidak ada.
> Ketiga sifat ini persis sifat matematika tertulis biasa — dan, seperti dijelaskan di atas, zero-knowledge klasik tidak dapat mempertahankannya.

Versi mega-Sudoku untuk merasakan perbedaannya

Berikut cara yang sengaja disederhanakan untuk merasakan perbedaannya.

Jangan gunakan Sudoku 9×9 biasa untuk bagian serius dari analogi ini. Ia terlalu kecil dan terlalu

berhingga: komputer dapat langsung menyelesaikannya, atau membuktikan bahwa ia tidak memiliki solusi. Bayangkan sebaliknya sebuah keluarga teka-teki **MegaSudoku(n)**. Besarkan aturan biasa: pilih ukuran blok n, tetapkan $N = n^2$, lalu bangun grid $N \times N$ yang dibagi menjadi blok $n \times n$, dengan N simbol. Sudoku biasa hanyalah kasus kecil $n = 3$, $N = 9$: grid 9×9, blok 3×3, dan sembilan simbol. Cerita kompleksitas bukti baru dimulai ketika n boleh tumbuh, dan ketika grid dapat membawa gadget tambahan yang membuatnya berperilaku seperti formula SAT yang menyamar sebagai Sudoku. Formula SAT hanyalah daftar kendala ya/tidak: dapatkah Anda memberi nilai true/false pada variabel sehingga setiap kendala terpenuhi?



Sudoku 25×25: aturannya dapat diperiksa tanpa mengungkap grid yang sudah selesai — pengganti visual untuk bukti yang memverifikasi solusi tersembunyi, yaitu witness.

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

Sudoku dan SAT: teka-teki yang sama dengan dua kostum

Klaim bahwa Sudoku dapat "berperilaku seperti formula SAT" bukan metafora. Terjemahannya berjalan dua arah, dan arah yang mudah dapat ditulis lengkap.

Dari Sudoku ke SAT. SAT hanya berbicara true/false, jadi berikan satu variabel boolean untuk setiap triple (baris, kolom, nilai): *x(r,c,v)* berarti "sel pada baris *r*, kolom *c* berisi nilai *v*." Sudoku 4×4 (blok 2×2, nilai 1–4) memerlukan 4·4·4 = 64 variabel; Sudoku klasik 9×9 memerlukan 729. Setiap aturan Sudoku kemudian menjadi sekumpulan klausa. (Klausa adalah OR dari variabel atau negasinya; seluruh formula adalah AND dari semua klausanya.)

Setiap sel memiliki setidaknya satu nilai — satu klausa per sel:

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

Setiap sel memiliki paling banyak satu nilai — klausa "tidak keduanya" untuk setiap pasangan nilai:

$\neg x(1,1,1) \lor \neg x(1,1,2)$ $\neg x(1,1,1) \lor \neg x(1,1,3)$ … dan seterusnya untuk keenam pasangan.

Setiap baris memuat setiap nilai — untuk baris 1 dan nilai 3: setidaknya sekali,

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

serta paling banyak sekali: $\neg x(1,1,3) \lor \neg x(1,2,3)$, dan seterusnya untuk setiap pasangan sel dalam baris.

Kolom dan blok — kumpulan klausa yang sama; hanya kelompok selnya yang berubah. Untuk blok kiri-atas dan nilai 2:

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

ditambah klausa "tidak keduanya" untuk setiap pasangan.

Petunjuk yang tercetak — bagian paling sederhana: setiap petunjuk adalah klausa dengan satu variabel. Angka 3 yang tercetak di sudut kiri atas menjadi klausa

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

AND dari semua ini satisfiable tepat ketika Sudoku memiliki solusi — dan sebuah satisfying assignment *adalah* solusinya: baca x(r,c,v) mana yang bernilai true lalu isi grid. Untuk Sudoku 9×9, jumlahnya 729 variabel dan beberapa ribu klausa, yang dapat diselesaikan SAT solver modern dalam milidetik. Perhatikan klausa petunjuk x(1,1,3): ia mengatakan "sel ini sama persis dengan 3", bukan "sel-sel ini semuanya berbeda" — asimetri yang sama yang nanti memerlukan trik tambahan untuk sel petunjuk dalam catatan protokol di bawah.

Dari SAT ke Sudoku. Makalah memerlukan arah sebaliknya yang lebih sulit: diberi formula SAT *sembarang*, bangun mega-Sudoku yang memiliki solusi tepat ketika formula itu memiliki solusi. Aturan asli Sudoku hanya dapat mengatakan "sel-sel ini semuanya berbeda", sehingga

kendala logis sembarang harus *dibangun* — dan itulah fungsi gadget. Gadget adalah kelompok kecil sel yang sudah dirancang, satu untuk setiap klausa formula, tempat sel tertentu berperan sebagai variabel (simbol yang dipegang mengodekan true atau false) dan kendala internal kelompok direkayasa sehingga satu-satunya pengisian legal sesuai dengan assignment yang memenuhi klausa tersebut. Ini adalah teknik standar dari pembuktian NP-completeness; untuk generalized Sudoku, konstruksi ini dikerjakan oleh Yato dan Seta pada 2003.
Kedua arah bersama-sama mengatakan bahwa Sudoku N×N dan SAT adalah masalah yang sama dengan kostum berbeda. Itulah yang mengizinkan artikel ini — dan makalahnya — menceritakan kisah tentang seluruh NP menggunakan grid dan simbol.

Witness-nya tetap mudah dibayangkan. Alice mengetahui pengisian lengkap yang valid untuk mega-Sudoku. Bob ingin diyakinkan bahwa pengisian seperti itu ada, tetapi Alice tidak ingin mengungkapkannya. Jika Alice mengirim seluruh grid, Bob yakin, tetapi rahasianya hilang.

Dalam versi zero-knowledge klasik, Alice dan Bob berinteraksi. Salah satu model mental lama menggunakan ubin tertutup. Alice menyembunyikan grid yang sudah terselesaikan, diam-diam mengganti nama simbol sebelum setiap putaran, lalu membiarkan Bob memeriksa satu kendala lokal yang dipilih secara acak: baris, kolom, kotak, atau gadget. Jika sel yang dibuka menunjukkan simbol yang semuanya berbeda, keyakinan Bob bertambah. Setelah itu semuanya ditutup kembali dan simbol diberi nama baru lagi. (Ada satu kerumitan: petunjuk yang sudah tercetak membutuhkan trik tambahan, karena penggantian nama juga menyembunyikannya. Catatan di bawah menjelaskan bagaimana protokol klasik menanganinya; gambaran sederhana ini cukup untuk pembahasan berikutnya.)

Bagaimana protokol klasik benar-benar menangani sel petunjuk

Trik penggantian nama memiliki titik buta. Aturan baris, kolom, dan kotak semuanya mengatakan "sel-sel ini semuanya berbeda", dan sifat *semuanya berbeda* bertahan di bawah penggantian nama simbol apa pun. Namun sebuah petunjuk mengatakan "sel ini berisi tepat 5", dan setelah penggantian nama Bob hanya melihat $\sigma(5)$ — sebuah simbol tersamarkan — tanpa mengetahui penggantian nama σ. Ia tidak dapat memeriksa apa pun. Jika dibiarkan, Alice dapat membuktikan bahwa *suatu* grid valid ada sambil sama sekali mengabaikan petunjuk tercetak, yang tidak membuktikan apa pun tentang *teka-teki ini*. Literatur klasik memiliki dua perbaikan standar.
Palet. Tambahkan satu baris ekstra berisi N sel ke grid tersembunyi — sebuah palet yang diisi Alice dengan simbol 1…N dalam urutan publik tetap, lalu ikut diberi nama ulang bersama semuanya, sehingga berisi $\sigma(1)$…$\sigma(N)$. Tantangan acak Bob sekarang memiliki satu pilihan tambahan. Selain memilih baris, kolom, kotak, atau gadget untuk dibuka, ia dapat memilih **palet plus satu sel petunjuk**. Alice membuka keduanya; palet mengungkap penggantian nama pada putaran itu, dan Bob memeriksa bahwa sel petunjuk menampilkan tepat versi yang telah diganti nama dari petunjuk tercetak. Ini tetap zero-knowledge karena Bob hanya mempelajari σ — yang digambar ulang setiap putaran dan tidak berguna sendirian — serta nilai sebuah sel yang sudah ia ketahui dari teka-teki. Tidak ada informasi tentang sel rahasia yang bocor, dan

simulator dapat memalsukan tampilan dengan memilih σ secara acak. Protokol tetap sound karena Alice yang curang tertangkap dengan probabilitas tetap per putaran, dan putaran diulang hingga keraguan menjadi sangat kecil.

Mengompilasi petunjuk hingga tidak perlu perlakuan khusus. Varian yang lebih struktural menghilangkan tantangan khusus alih-alih menambahkannya. Alih-alih *memverifikasi* nilai petunjuk, paksa nilainya dengan kendala perbedaan: hubungkan sel petunjuk ke setiap sel palet kecuali yang membawa nilainya sendiri — "berbeda dari $\sigma(1)$, berbeda dari $\sigma(2)$, ..., berbeda dari semuanya kecuali $\sigma(5)$." Satu-satunya simbol yang secara legal dapat dimiliki sel itu adalah nilai petunjuknya. Kini setiap kendala kembali berbentuk "dua ini berbeda" — invariant terhadap penggantian nama dan dapat diperiksa persis seperti baris. Ini adalah manuver yang sama yang digunakan untuk simpul yang telah diwarnai sebelumnya dalam protokol graph-coloring klasik, dan sejalan dengan makna *gadget* di atas: dalam gambaran MegaSudoku-sebagai-SAT, petunjuk dikompilasi menjadi gadget ketaksamaan seperti kendala lain.

Protokol fisik. Protokol kartu dunia nyata untuk Sudoku (Gradwohl, Naor, Pinkas, dan Rothblum, 2007) tidak menggunakan penggantian nama sama sekali dan menyelesaikan masalah petunjuk bahkan sebelum penyembunyian dimulai. Untuk setiap sel, Alice meletakkan tiga kartu identik dengan nilai sel — tertutup untuk sel rahasia, tetapi **terbuka untuk sel petunjuk**, sehingga Bob melihat sendiri bahwa petunjuk dipatuhi sebelum kartu dibalik. Lalu satu kartu dari setiap sel masuk ke paket barisnya, satu ke paket kolomnya, satu ke paket kotaknya; setiap paket diacak dan dibuka, dan Bob memeriksa bahwa paket memuat semua N simbol. Pengacakan menghancurkan informasi posisi (itulah zero-knowledge), tetapi petunjuk sudah dipastikan sejak kartu dibagikan.

Bagaimanapun caranya, pelajarannya sama dengan yang terus diulang artikel ini: protokol zero-knowledge adalah pencatatan yang hati-hati tentang *fakta mana* yang bertahan setelah penyembunyian. Penggantian nama mempertahankan "semuanya berbeda" dan menghapus "sama dengan 5" — sehingga "sama dengan 5" harus dimasukkan kembali dengan cara lain.

Itu bukan protokol dalam makalah. Itu adalah model mental untuk zero-knowledge klasik:

- Alice dan Bob saling berbalas pesan.
- Bob memilih pemeriksaan acak.
- Alice hanya mengungkap konsistensi lokal, bukan seluruh solusi.
- Bukti privasi bekerja dengan menunjukkan bahwa tampilan Bob dapat dihasilkan tanpa solusi rahasia Alice.

Jadi zero-knowledge klasik dibangun di atas fakta positif:

> Sebuah simulator benar-benar ada.

Sekarang hilangkan bagian yang nyaman. Alice mengirim satu string bukti lalu pergi. Tidak ada trusted setup, tidak ada string acak bersama yang dipersiapkan sebelumnya, dan Bob tidak boleh pernah menerima teka-teki yang salah. Inilah setting yang tidak dapat dipertahankan oleh zero-knowledge klasik.

Satu tokoh lagi diperlukan sebelum triknya. Tetapkan sebuah **rulebook**: sistem pembuktian formal dalam arti logika — satu set aksioma tetap plus aturan mekanis untuk memeriksa bukti matematika tertulis. ZFC, aksioma standar matematika, adalah contoh kanonik. Dari sini semua pernyataan dibuat relatif terhadap rulebook yang dipilih sebelumnya, dan pilihannya fleksibel: konstruksi bekerja untuk rulebook mana pun yang Anda tetapkan, termasuk ZFC.

(Catatan istilah, mengikuti makalah: "proof system" di sini selalu berarti rulebook ini — sistem formal yang memeriksa bukti matematika — bukan pesan yang dikirim Alice. Mesin Alice dan Bob disebut "prover dan verifier.")

Versi bergaya Gödel mempertahankan cerita mega-Sudoku tetapi mengubah buktinya.

Pilih sistem kendala kedua dengan ukuran tampilan yang sama, sebut **D**. Dalam cerita, S dan D adalah dua teka-teki MegaSudoku(n) dalam format yang sama. Di balik layar, D mungkin bermula sebagai formula logika sulit dengan ukuran berbeda; jika perlu, ia dapat dipenuhi dengan kendala dummy yang tidak berbahaya agar cocok dengan grid yang sama. D dibangun dari formula logika yang sebenarnya **unsatisfiable**: tidak ada assignment nilai yang dapat membuat semua kendalanya benar, seperti teka-teki rusak yang tidak memiliki grid lengkap legal. Contoh mainan adalah formula yang menuntut sekaligus "X benar" dan "X salah." Jadi D memang tidak memiliki pengisian valid.

Namun D tidak boleh menjadi teka-teki rusak yang *mudah dibongkar*. Contoh mainan tadi gagal dalam hal ini: rulebook mana pun dapat membantah "X dan bukan-X" dalam satu baris. D harus salah dengan cara yang tidak dapat disertifikasi oleh rulebook pilihan dengan argumen pendek. Jika rulebook dapat membantah D dengan bukti pendek, cerita di bawah akan runtuh: jalur alternatif yang mungkin menghasilkan bukti tanpa rahasia Alice dapat secara formal disingkirkan, dan bersama itu jaminan privasinya. Karena itu D dipilih dari keluarga yang tidak dapat dibantah secara efisien oleh rulebook tetap: di dalam rulebook tersebut tidak ada bukti pendek bahwa D tidak memiliki solusi.

Bukti satu-pesan Alice kemudian menyangkut pernyataan salah satu dari dua kemungkinan:

> mega-Sudoku nyata S memiliki solusi, atau decoy D memiliki solusi.

Inilah kaitan logisnya. D **tidak** dihasilkan dengan cara ajaib yang membuat S menjadi benar. Bukti tidak berargumen "D tidak punya solusi, maka S punya solusi." Yang dibuktikan adalah disjungsi **S atau D**. Perfect soundness berarti disjungsi palsu tidak dapat memiliki bukti valid. Karena dalam kenyataan D salah — tidak memiliki solusi — satu-satunya cara disjungsi itu benar adalah jika S benar. Jadi jika bukti diterima, S harus memiliki solusi. Decoy tidak dapat membuat S yang salah menjadi benar.

Namun untuk bagian bergaya zero-knowledge, tanyakan apa yang akan terjadi jika D *memiliki* solusi. Solusi decoy itu dapat bertindak sebagai witness alternatif. Ia memungkinkan seseorang menghasilkan bukti tanpa mengetahui solusi mega-Sudoku nyata milik Alice — dengan kata lain, sebuah simulator. Dalam kenyataan D tidak punya solusi, jadi jalur simulator ini tertutup. Intinya adalah rulebook tidak dapat secara efisien membuktikan bahwa jalur itu tertutup.

Jadi D memiliki dua tugas. Untuk **soundness**, D salah, sehingga bukti valid dari "S atau D" memaksa S benar. Untuk **effective zero-knowledge**, D sulit dibantah, sehingga rulebook tidak dapat dengan cepat menyingkirkan jalur decoy yang, seandainya tersedia, memungkinkan simulasi.

Maka pengujian keamanan bukan lagi:

> Dapatkah kita membuktikan bahwa sebuah simulator benar-benar ada?

Ia berubah menjadi:

> Dapatkah rulebook Anda secara efisien membuktikan bahwa simulator itu mustahil?

Jika jawabannya tidak, sesuatu yang mengejutkan kuat terjadi: setiap jaminan keamanan yang (a) dapat diamati dengan menjalankan pengujian, dan (b) secara terbukti mengikuti — di dalam rulebook itu — dari keberadaan simulator, benar-benar berlaku. Serangan yang berhasil terhadap salah satu jaminan itu sendiri akan menghasilkan refutasi pendek yang hilang tadi, sementara refutasi pendek itu tidak ada. Itulah bagian "effective" dari effectively zero-knowledge.

Jadi kontras versi kelasnya adalah:

Zero-knowledge klasik: bukti aman karena simulator ada.

Effective zero-knowledge bergaya Gödel: bukti diperlakukan aman untuk pengujian keamanan yang dapat diamati karena rulebook tidak dapat secara efisien membuktikan bahwa simulator mustahil.

Klaim kedua lebih lemah. Dan justru itu sebabnya makalah dapat mempertahankan tiga sifat yang merusak versi klasik: satu pesan, tanpa setup, dan perfect soundness.

Pengujian baru: Anda tidak dapat membuktikan simulator tidak ada

Relaksasi Ilango mengubah pertanyaannya.

Zero-knowledge klasik bertanya:

> Apakah sebuah simulator ada?

Effectively zero-knowledge mengajukan pertanyaan yang lebih lemah:

> Dapatkah rulebook pilihan Anda secara efisien membuktikan bahwa tidak ada simulator?

Ini terdengar seperti menghindar secara teknis, tetapi justru inti gagasannya. Konstruksi berada dalam keadaan aneh: simulator sebenarnya tidak ada — makalah menyatakannya secara eksplisit — tetapi rulebook yang Anda tetapkan tidak dapat secara efisien membuktikan bahwa simulator itu tidak ada. Jika setiap konsekuensi buruk yang Anda pedulikan memerlukan refutasi semacam itu, sistem tetap berperilaku seperti zero-knowledge untuk konsekuensi tersebut.

Di sinilah Gödel masuk. Bukan sebagai hiasan, dan bukan sebagai "Gödel membuat kriptografi aman." Hubungannya bersifat proof-theoretic. Sebuah rulebook disebut **optimal** jika, dalam arti yang tepat, ia adalah yang terbaik yang mungkin: kapan pun rulebook mana pun dapat membantah formula dari jenis yang relevan dengan bukti pendek, rulebook optimal juga dapat melakukannya dengan bukti yang paling banyak hanya lebih panjang secara polinomial. Krajíček dan Pudlák pada 1989 mengajukan konjektur bahwa **tidak ada proof system optimal**: rulebook apa pun yang Anda tetapkan, ada rulebook lain yang membuktikan suatu keluarga pernyataan benar jauh lebih ringkas. Ini salah satu konjektur terbuka utama dalam proof complexity, dan merupakan kerabat berhingga serta berorientasi kompleksitas dari teorema ketidaklengkapan Gödel: beberapa pernyataan benar tidak memiliki bukti pendek dalam rulebook yang Anda pilih — bukan karena tidak dapat dibuktikan secara prinsip, tetapi karena setiap rulebook tetap meninggalkan sebagian kebenaran pendek tanpa bukti pendek.

Makalah mengasumsikan konjektur ini (dalam bentuk "infinitely often" yang sedikit lebih kuat, seperti biasa ketika konjektur digunakan secara kriptografis). Hasilnya, melalui sebuah teorema Krajíček dan Pudlák, sangat konkret: untuk setiap rulebook ada urutan formula yang benar-benar unsatisfiable tetapi tidak dapat dibantah rulebook dengan bukti pendek — dan, yang penting, formula-formula itu dapat *dihasilkan* oleh algoritme efisien. Sifat terakhir ini, uniformity, mengubah seluruh gagasan dari klaim keberadaan menjadi algoritme nyata yang dapat dijalankan Alice: decoy D miliknya keluar dari jalur produksi, bukan muncul dari udara.

Langkah kriptografinya adalah memanfaatkan kekurangan daya pembuktian itu.

Apa yang dilakukan konstruksi ini

Berikut konstruksi makalah dalam bentuk paling ringkas.

Tetapkan sebuah rulebook — misalnya ZFC. Di bawah asumsi proof complexity, ada urutan formula yang dapat dihasilkan secara efisien, yang dalam kenyataan unsatisfiable, tetapi rulebook tidak memiliki bukti pendek bahwa formula-formula tersebut unsatisfiable.

Sekarang bangun bukti satu-pesan dengan bentuk berikut:

> pernyataan nyata satisfiable, atau formula sulit khusus ini satisfiable.

Formula sulit khusus itu sebenarnya tidak satisfiable. Jadi jika mesin pembuktian dasarnya memiliki perfect soundness, penerimaan pesan tetap berarti pernyataan nyata benar. Itulah yang memberi perfect soundness.

Namun untuk keamanan bergaya zero-knowledge, bayangkan formula sulit khusus itu *satisfiable*. Witness-nya kemudian dapat digunakan untuk mensimulasikan bukti tanpa mengetahui witness nyata. Dalam kenyataan formula itu tidak satisfiable — tetapi rulebook tidak dapat secara efisien membuktikannya. Jadi rulebook tidak dapat secara efisien membuktikan bahwa simulator mustahil.

Itulah engselnya. Sistem tidak menyembunyikan rahasia dengan menghasilkan simulator klasik. Un-

tuk kelas besar pengujian keamanan yang dapat diamati, sistem menyembunyikan rahasia di balik ketidakmampuan rulebook untuk mensertifikasi bahwa simulator tidak ada.

Apa yang diklaim makalah

Teorema utama hadir dalam beberapa lapisan. Hasil intinya adalah ini:

Di bawah asumsi kriptografi standar — keberadaan **non-interactive witness indistinguishable proofs**, objek yang telah dipelajari luas dan mengikuti dari beberapa paket asumsi yang mapan — serta di bawah konjektur proof complexity bahwa **tidak ada (infinitely often) optimal proof system**, makalah membangun, untuk setiap pilihan rulebook, prover dan verifier satu-pesan untuk NP/SAT dengan **perfect soundness** dan tanpa setup yang effectively zero-knowledge relatif terhadap rulebook tersebut. (NP/SAT adalah "penyebut bersama paling sulit" standar dari masalah-masalah berbentuk teka-teki; mega-Sudoku adalah salah satu kostumnya.)

Untuk klaim yang lebih luas tentang mempertahankan sifat keamanan yang falsifiable, makalah menambahkan satu asumsi standar lagi, keyakinan derandomisasi **P = BPP** (secara kasar: randomness tidak memberi algoritme kekuatan tambahan yang esensial).

Diterjemahkan dari bahasa teorema:

- Buktinya satu pesan.
- Tidak ada trusted setup.
- Pernyataan salah tidak dapat dibuktikan.
- Prover bukan zero-knowledge klasik — ia tidak memiliki simulator.
- Tetapi setiap konsekuensi keamanan zero-knowledge klasik yang falsifiable dan berbasis game dapat dicapai dalam setting ini.

Kata "falsifiable" penting. Artinya kegagalan keamanan dapat diuji dengan menjalankan adversary dalam sebuah game. Banyak definisi keamanan kriptografi berbentuk seperti ini: dapatkah adversary membedakan dua enkripsi, membalik fungsi, memulihkan witness, atau memenangkan eksperimen tertentu? Teorema memberikan prover untuk setiap sifat falsifiable, satu per satu. Satu prover yang sekaligus memiliki *semua* sifat falsifiable kemungkinan mustahil — serangan reusability lama ("Bob dapat menunjukkan bukti kepada orang lain") sendiri merupakan sifat yang dapat difalsifikasi, dan memang gagal di sini. Usulan makalah adalah bahwa satu prover secara masuk akal dapat mencakup semua sifat falsifiable yang *natural* — yang benar-benar muncul dalam praktik kriptografi — tetapi bagian itu adalah teorema kondisional yang bertumpu pada gagasan informal tentang "natural", ditambah konjektur eksplisit. Jaminannya ditujukan pada kegagalan yang dapat diamati, bukan setiap makna filosofis atau berbasis simulasi dari kerahasiaan.

Satu korolari konkret layak disebut: konstruksi menghasilkan bukti *witness hiding* non-interaktif pertama dengan prover uniform — "bukti bahwa sebuah teka-teki memiliki solusi tidak membantu Anda menemukan solusinya", tanpa interaksi dan tanpa setup — sebuah objek yang terdengar sederhana tetapi telah menolak konstruksi selama puluhan tahun.

Apa yang tidak dikatakan makalah

Bagian ini menjaga penjelasan tetap jujur.

Ini **tidak** mengatakan teorema ketidakmungkinan lama salah. Konstruksi menghindarinya dengan mengubah definisi.

Ini **tidak** memberikan zero-knowledge biasa dan klasik dengan tanpa interaksi, tanpa setup, serta perfect soundness. Makalah secara eksplisit mengatakan prover yang dibangun tidak memiliki simulator.

Ini **tidak** berarti bukti tidak dapat digunakan ulang. Bukti satu-pesan tetap dapat ditunjukkan kepada orang lain; makalah tidak mempertahankan sifat bergaya deniability. (Non-interactive zero-knowledge dengan trusted setup juga memiliki keterbatasan yang sama.)

Ini **tidak** berarti protokol ini praktis dan siap dipakai. Ini adalah teori kompleksitas dan fondasi kriptografi. Hasil bergantung pada asumsi besar dari proof complexity dan kriptografi, dan konstruksinya membahas apa yang mungkin secara prinsip.

Ini **tidak** menjadikan "Gödel" primitif keamanan ajaib. Hubungan Gödel datang melalui proof system, optimal proof system, dan analog berhingga dari ketidaklengkapan. Intuisi yang berguna bukan "ketidaklengkapan melindungi password Anda." Intuisinya adalah: jika rulebook tidak dapat secara efisien membuktikan bahwa simulator mustahil, maka serangan yang memerlukan pembuktian itu dapat diblokir pada tingkat definisi keamanan.

Mengapa tetap menarik

Kriptografi sering mengubah kesulitan menjadi keamanan. Faktorisasi sulit, maka asumsi bergaya RSA menjadi berguna. Masalah lattice sulit, maka kriptografi lattice menjadi berguna. Di sini bentuk kesulitannya lebih aneh: bukan "sulit menghitung rahasia", melainkan "sulit membuktikan bahwa suatu objek bukti tertentu tidak dapat ada."

Itulah sebabnya makalah terasa tidak biasa. Ia memperlakukan aksioma dan rulebook hampir seperti sumber daya kriptografis. Ketidakmungkinan biasa mengatakan ada ketegangan antara soundness dan simulasi. Langkah Ilango menempatkan ketegangan itu di balik tirai proof-theoretic: simulator tidak ada, tetapi sistem formal tidak dapat secara efisien menyingkap ketidakhadirannya.

Bagi pembaca, bagian mengejutkannya bukan bahwa ini akan menggantikan sistem zero-knowledge masa kini. Kemungkinan besar tidak, setidaknya tidak secara langsung. Yang mengejutkan adalah bahwa sebuah keterbatasan dari logika matematika dapat dipakai secara konstruktif: bukan hanya sebagai dinding, tetapi sebagai semacam penutup.

Seberapa kuat buktinya?

Ini makalah teorema, sehingga "bukti" memiliki arti berbeda dari makalah biologi atau astronomi. Pertanyaannya bukan apakah eksperimen direplikasi. Pertanyaannya adalah apakah definisi, asumsi, dan rantai pembuktian mendukung klaim.

Buktinya formal, dan makalah eksplisit tentang asumsi-asumsinya. Asumsinya bukan sembarangan. Non-interactive witness indistinguishable proofs adalah objek standar dalam kriptografi dan mengikuti dari beberapa paket asumsi yang sudah mapan. Konjektur no-optimal-proof-system adalah konjektur sentral dalam proof complexity. P = BPP adalah keyakinan derandomisasi standar yang hanya dipakai untuk teorema sifat falsifiable yang lebih luas.

Makalah juga berargumen bahwa asumsi tersebut adalah harga yang tepat, bukan penyangga arbitrer: ia membuktikan converse yang menunjukkan asumsi tersebut pada dasarnya diperlukan — jika konstruksi seperti ini ada sama sekali, maka non-interactive witness indistinguishable proofs harus ada, dan (dengan mengasumsikan one-way functions standar) optimal proof system tidak dapat ada. Asumsinya juga "win-win": jika salah satunya terbantahkan, pembantahan itu sendiri akan menjadi penemuan penting dalam proof complexity, kriptografi, atau teori kompleksitas.

Namun karena hasilnya kondisional, tingkat keyakinannya juga kondisional. Jika asumsi tersebut gagal, interpretasi teorema berubah. Dan bahkan jika asumsi benar, jaminannya bukan zero-knowledge klasik penuh; ia adalah versi relaksasi proof-theoretic dari makalah.

Jadi tingkat keyakinan yang tepat adalah tinggi bahwa makalah menetapkan hasil kemungkinan kondisional yang koheren; sedang bahwa asumsi-asumsinya menggambarkan dunia kriptografi yang benar-benar kita tinggali; dan rendah untuk konsekuensi praktis langsung.

Mengapa ini penting

Makalah membuka jalur yang sebelumnya dianggap tertutup.

Teori klasik mengatakan: zero-knowledge penuh tidak dapat berupa satu pesan tanpa setup, dan tidak dapat memiliki perfect soundness. Makalah Ilango mengatakan: jika kita meminta konsekuensi zero-knowledge yang dapat diuji dalam security game, dan mengizinkan definisi keamanan bergantung pada apa yang dapat atau tidak dapat dibantah secara efisien oleh sebuah rulebook, maka banyak perilaku berguna dapat dipulihkan — dengan satu pesan, tanpa setup, dan perfect soundness.

Ini bukan perubahan definisi kecil. Ini cara berbeda untuk memikirkan jaminan kriptografis. Alih-alih hanya bertanya apa yang ada, tanyakan apa yang dapat disingkirkan rulebook Anda. Alih-alih memperlakukan ketidakdapatdibuktian sebagai gangguan filosofis, gunakan sebagai struktur.

Dunia praktis mungkin tidak berubah besok. Tetapi peta konseptual berubah. Kini ada arti formal di mana "tidak ada yang dapat secara efisien membuktikan bahwa rahasia bocor" dapat cukup kuat untuk memulihkan banyak perlindungan berbasis game yang kita inginkan dari "rahasia tidak bocor."

Itulah sebabnya Gödel pantas berada di judul.

Singkatnya

Zero-knowledge proofs memungkinkan prover meyakinkan verifier bahwa sebuah pernyataan benar tanpa mengungkap witness. Hasil ketidakmungkinan klasik mengatakan zero-knowledge tidak dapat diperas menjadi satu pesan tanpa setup, dan tidak dapat memiliki perfect soundness. Makalah Rahul Ilango tidak membantah ketidakmungkinan itu. Ia mendefinisikan gagasan yang lebih lemah, effectively zero-knowledge: alih-alih mensyaratkan simulator benar-benar ada, ia mensyaratkan bahwa proof system pilihan — rulebook formal seperti ZFC — tidak dapat secara efisien membuktikan bahwa simulator tidak ada. Di bawah asumsi besar dari kriptografi (non-interactive witness indistinguishable proofs) dan proof complexity (tidak ada optimal proof system), makalah membangun prover satu-pesan untuk NP/SAT tanpa setup dan dengan perfect soundness yang mencapai konsekuensi zero-knowledge yang falsifiable dan berbasis game, sifat demi sifat. Satu prover yang mencakup semua sifat "natural" semacam itu merupakan perluasan lebih lanjut yang sebagian masih konjektural — dan mencakup secara literal setiap sifat falsifiable kemungkinan mustahil karena bukti tetap dapat digunakan ulang. Hasil ini teoretis dan kondisional, bukan primitif siap pakai, tetapi menunjukkan cara baru menggunakan ketidakdapatdibuktian proof-theoretic sebagai sumber daya kriptografis.

Penilaian tanpa berlebihan

Apa yang ditunjukkan makalah: Di bawah asumsi yang dinyatakan, dapat dibangun prover satu-pesan, tanpa setup, dan perfectly sound untuk NP/SAT yang effectively zero-knowledge relatif terhadap proof system pilihan mana pun, dan yang mencapai setiap konsekuensi zero-knowledge klasik berbasis game yang falsifiable.

Apa yang masuk akal tetapi belum terbukti tanpa syarat: Bahwa asumsi proof-complexity dan kriptografi yang diperlukan benar. Itu adalah asumsi serius dan dipelajari luas — dan makalah menunjukkan bahwa asumsi tersebut pada dasarnya diperlukan sekaligus cukup — tetapi tetap merupakan asumsi.

Apa yang tidak ditunjukkan: Zero-knowledge klasik dengan tanpa interaksi, tanpa setup, dan perfect soundness; sistem praktis yang siap digunakan; deniability atau non-reusability bukti; atau bahwa teorema ketidaklengkapan Gödel dengan sendirinya mengamankan kriptografi.

Keterbatasan utama: Jaminannya merupakan relaksasi dari zero-knowledge; versi terluas bergantung pada beberapa asumsi; klaim tentang satu prover universal tetap sebagian konjektural; dan hasil terutama bersifat fondasional.

Seberapa besar kepercayaan yang layak dimiliki pembaca umum? Tinggi bahwa ini adalah hasil teori kondisional penting jika definisinya diterima. Sedang bahwa asumsi-asumsinya mencerminkan kenyataan. Rendah untuk penerapan praktis segera. Kesimpulan amannya: makalah tidak

mematahkan ketidakmungkinan zero-knowledge; ia menemukan cara proof-theoretic baru untuk mengitari bagian dari ketidakmungkinan tersebut yang penting bagi banyak security game.

Sumber

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