The Clean Paper

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

Uma prova criptográfica pode esconder o segredo tornando difícil provar que o simulador está ausente

As provas de conhecimento zero permitem provar uma afirmação sem revelar a testemunha. A teoria clássica diz que, no sentido completo, não se pode ter isso com uma única mensagem, nenhuma configuração de confiança e correção perfeita. O artigo de Rahul Ilango não faz desaparecer essa impossibilidade. Muda o alvo: em vez de exigir que exista realmente um simulador, pede que nenhum sistema de prova escolhido consiga provar eficientemente que o simulador não existe. Sob hipóteses importantes de complexidade de provas e criptografia, isso basta para recuperar, propriedade a propriedade, consequências falsificáveis e baseadas em jogos do conhecimento zero. O ponto não é uma primitiva pronta a ligar à Internet. É uma forma baseada em teoria da prova de transformar indemonstrabilidade matemática em cobertura criptográfica.

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

O truque não é provar que o segredo está escondido

Comecemos pela versão mais simples de conhecimento zero.

Alice quer convencer Bob de que um Sudoku tem solução. Se lhe enviar a solução, Bob fica convencido, mas o puzzle fica estragado. O que ela quer é algo mais estranho: uma prova de que existe uma solução, sem revelar a solução.

Essa é a promessa de uma prova de conhecimento zero (*zero-knowledge proof*). O provador (Alice) convence o verificador (Bob) de que uma afirmação é verdadeira sem revelar nada para além da verdade dessa afirmação.

O problema é que esta promessa tem um custo. Uma prova matemática comum tem duas propriedades confortáveis. É **uma única mensagem**: escreve-se, entrega-se e vai-se embora. E tem **correção perfeita** (*perfect soundness*): uma afirmação falsa não tem prova válida. Resultados clássicos de impossibilidade dizem que o conhecimento zero tem de abdicar de ambas as propriedades — e não apenas das duas em conjunto; cada uma, isoladamente, já está fora de alcance.

Primeiro, uma prova de conhecimento zero precisa de conversa. Se Alice enviar uma única mensagem, sem uma configuração de confiança preparada antecipadamente, a garantia de conhecimento zero colapsa — independentemente de quanta correção se esteja disposto a sacrificar em troca.

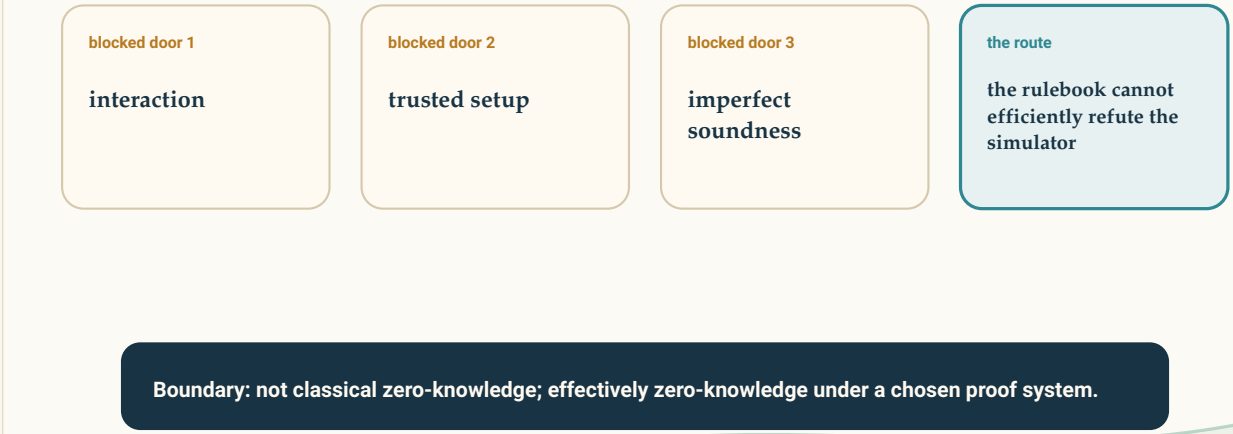
Segundo, uma prova de conhecimento zero precisa de tolerar uma pequena probabilidade de erro. Exigir correção perfeita acaba por destruir silenciosamente também a interação: um verificador que nunca pode ser enganado, sejam quais forem as escolhas aleatórias que faça, poderia simplesmente fixar essas escolhas antecipadamente — e, quando o verificador se torna previsível, Alice pode responder a tudo numa única mensagem, exatamente o caso que já tinha falhado.

O artigo de Rahul Ilango trata de uma forma de contornar esta dupla parede. Não fingindo que a parede não existe, nem produzindo conhecimento zero clássico no cenário impossível. O movimento é mais subtil: enfraquecer o significado de «não revela nada», mas fazê-lo de uma forma que preserve as propriedades de segurança que os criptógrafos conseguem efetivamente testar.

O resultado chama-se **conhecimento zero efetivo** (*effectively zero-knowledge*).

A proof without leaking the witness

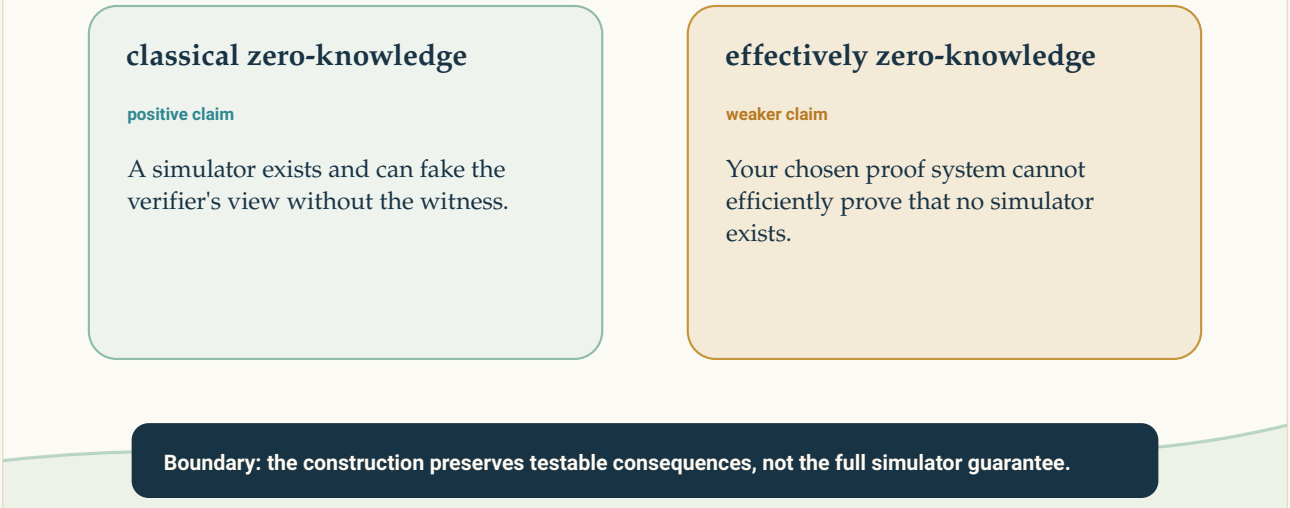
The paper keeps the ordinary-proof shape by weakening what privacy must prove.



O conhecimento zero encontra três portas bloqueadas — interação, configuração de confiança e correção imperfeita. A construção de Ilango passa por outra: o livro de regras não consegue refutar eficientemente o simulador.

Classical privacy vs effective privacy

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



O conhecimento zero clássico pergunta se existe um simulador; o «conhecimento zero efetivo» pergunta

apenas se o livro de regras escolhido consegue provar eficientemente que não existe. Esta pergunta mais fraca é o que permite à construção manter uma mensagem, nenhuma configuração e correção perfeita.

O teste antigo: existe um simulador

A forma clássica de formalizar conhecimento zero usa um auxiliar fictício chamado **simulador**.

A ideia é esta: imagine Jane, que **não** conhece o segredo de Alice. Se Jane conseguir gerar, inteiramente por si própria, provas que parecem exatamente as que Bob teria recebido de Alice, então as provas de Alice não ensinaram nada de novo a Bob. Jane já conseguia simular a experiência sem o segredo de Alice.

Assim, o conhecimento zero clássico exige um simulador real. Tem de existir um algoritmo eficiente capaz de produzir provas falsas mas indistinguíveis sem conhecer o segredo — a *testemunha* (*witness*), no jargão; no Sudoku, a testemunha é simplesmente a grelha resolvida.

Esta definição é poderosa, mas é também precisamente onde a antiga impossibilidade morde. Eis a intuição. Uma prova verdadeiramente não interativa é apenas uma sequência de símbolos. Assim que Bob a possui, pode mostrá-la a outra pessoa: ganhou a capacidade de provar a afirmação a terceiros, o que já soa a mais do que «nada». Os teoremas clássicos tornam esta intuição precisa nas impossibilidades acima.

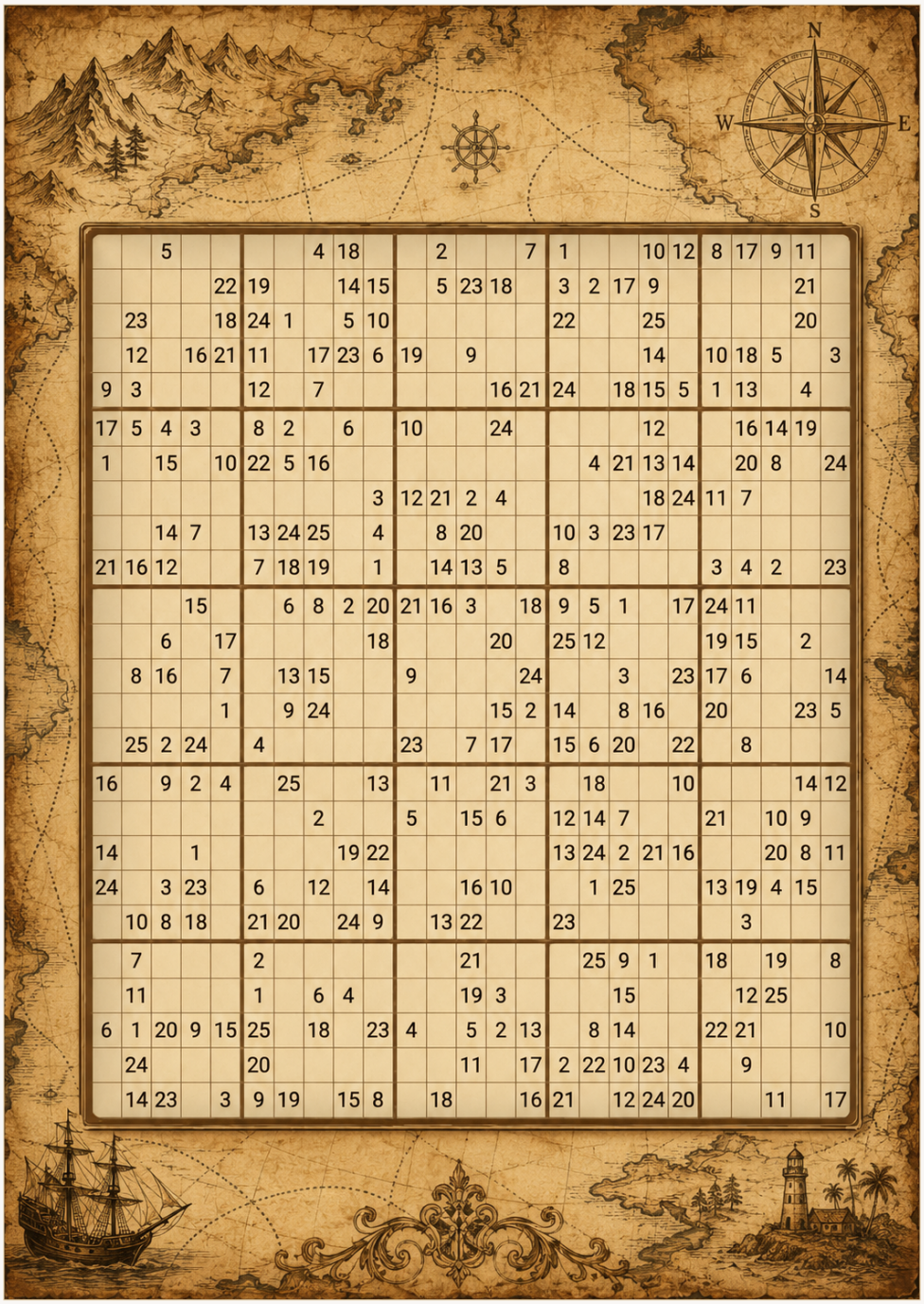
> **As três propriedades em que este artigo insiste**
>
> O título do artigo científico nomeia três restrições:
> **Sem interação:** Alice envia uma única sequência de prova. Não há protocolo de ida e volta.
> **Sem configuração:** Alice e Bob não dependem de uma cadeia de referência comum de confiança nem de outra aleatoriedade pública preparada antecipadamente. Muitos sistemas chamados «conhecimento zero não interativo» continuam a depender de uma configuração; aqui significa-se configuração zero.
> **Correção perfeita:** uma afirmação falsa não tem qualquer prova válida. Não «é aceite quase nunca»; não existe prova válida.
> Estas três propriedades são exatamente as que a matemática escrita comum possui — e, como explicado acima, o conhecimento zero clássico não consegue mantê-las.

Uma versão MegaSudoku da diferença

Aqui está uma forma deliberadamente simplificada de sentir a diferença.

Não usemos um Sudoku normal de 9 por 9 para a parte séria da analogia. É demasiado pequeno e demasiado finito: um computador pode simplesmente resolvê-lo ou provar que não tem solução.

Imagine-se, em vez disso, uma família de puzzles **MegaSudoku(n)**. Escale-se a regra habitual: escolha-se um tamanho de bloco *n*, defina-se *N = n^2* e construa-se uma grelha *N* por *N*, dividida em blocos *n* por *n*, com *N* símbolos. O Sudoku comum é apenas o pequeno caso *n = 3*, *N = 9*: uma grelha 9 por 9, blocos 3 por 3 e nove símbolos. A história de complexidade de provas só começa quando se deixa *n* crescer e quando a grelha pode conter dispositivos adicionais (*gadgets*) que a fazem comportar-se como uma fórmula SAT vestida de Sudoku. Uma fórmula SAT é apenas uma lista de restrições sim/não: é possível atribuir verdadeiro/falso às variáveis de modo que todas as restrições sejam satisfeitas?



Um Sudoku 25x25: as suas regras podem ser verificadas sem revelar a grelha terminada — uma representação visual de uma prova que verifica uma solução escondida, a testemunha.

Sudoku e SAT: o mesmo puzzle com dois disfarces

A afirmação de que um Sudoku pode «comportar-se como uma fórmula SAT» não é uma metáfora. A tradução funciona nos dois sentidos, e o sentido fácil pode ser escrito por completo.

De Sudoku para SAT. SAT só fala verdadeiro/falso, por isso dê-se-lhe uma variável booleana por cada triplo (linha, coluna, valor): *x(r,c,v)* significa «a célula na linha *r*, coluna *c*, contém o valor *v*». Um Sudoku 4 por 4 (blocos 2 por 2, valores 1–4) precisa de 4·4·4 = 64 variáveis; o clássico 9 por 9 precisa de 729. Cada regra do Sudoku transforma-se então num conjunto de cláusulas. (Uma cláusula é um OU de variáveis ou das suas negações; a fórmula completa é o E de todas as cláusulas.)

Cada célula contém pelo menos um valor — uma cláusula por célula:

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

Cada célula contém no máximo um valor — uma cláusula «não ambos» para cada par de valores:

$$\neg x(1,1,1) \lor \neg x(1,1,2) \quad \neg x(1,1,1) \lor \neg x(1,1,3)$$ … e assim por diante para os seis pares.

Cada linha contém cada valor — para a linha 1 e o valor 3: pelo menos uma vez,

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

e no máximo uma vez: $\neg x(1,1,3) \lor \neg x(1,2,3)$, e assim por diante para cada par de células da linha.

Colunas e blocos — conjuntos idênticos; muda apenas o grupo de células. Para o bloco superior esquerdo e o valor 2:

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

mais as cláusulas «não ambos» para cada par.

As pistas impressas — a parte mais simples: cada pista é uma cláusula com uma única variável. Um 3 impresso no canto superior esquerdo transforma-se na cláusula

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

O E de tudo isto é satisfazível exatamente quando o Sudoku tem solução — e uma atribuição satisfatória *é* a solução: basta ver quais x(r,c,v) são verdadeiros e preencher a grelha. Num 9 por 9, isto dá 729 variáveis e alguns milhares de cláusulas, que um solucionador SAT moderno resolve em milissegundos. Repare na cláusula da pista x(1,1,3): diz «esta célula é exatamente 3», e não «estas células são todas diferentes» — a mesma assimetria que obrigará ao truque adicional para as células com pistas na nota sobre o protocolo mais abaixo.

De SAT para Sudoku. O artigo precisa da direção oposta e mais difícil: dada uma fórmula SAT *arbitrária*, construir um mega-Sudoku que tenha solução exatamente quando a fórmula a tem. As regras nativas do Sudoku só conseguem dizer «estas células são todas diferentes», pelo que

restrições lógicas arbitrárias têm de ser *construídas* — e é precisamente para isso que servem os *gadgets*. Um *gadget* é um pequeno conjunto pré-fabricado de células, um por cláusula da fórmula, no qual certas células designadas desempenham o papel das variáveis (o símbolo que contêm codifica verdadeiro ou falso) e as restrições internas do conjunto são concebidas para que os únicos preenchimentos legais correspondam a atribuições que satisfazem essa cláusula. É trabalho padrão das demonstrações de NP-completude; para Sudoku generalizado, foi feito por Yato e Seta em 2003.
Em conjunto, as duas direções dizem que Sudoku N-por-N e SAT são o mesmo problema com dois disfarces. É isso que autoriza este artigo — e o artigo científico — a contar uma história sobre toda a classe NP usando grelhas e símbolos.

A testemunha continua fácil de visualizar. Alice conhece um preenchimento completo e válido do mega-Sudoku. Bob quer ficar convencido de que esse preenchimento existe, mas Alice não o quer revelar. Se enviar a grelha inteira, Bob fica convencido, mas o segredo desaparece.

Na versão clássica de conhecimento zero, Alice e Bob interagem. Um modelo mental antigo usa peças tapadas. Alice esconde a grelha resolvida, renomeia secretamente os símbolos antes de cada ronda e deixa Bob inspecionar uma restrição local escolhida aleatoriamente: uma linha, uma coluna, uma caixa ou um *gadget*. Se as células abertas mostrarem símbolos todos diferentes, Bob ganha confiança. Depois volta a tapar-se tudo e os símbolos são novamente renomeados. (Há uma complicação: as pistas dadas no puzzle precisam de um truque adicional, porque renomear os símbolos também as esconde. A nota abaixo explica como os protocolos clássicos resolvem isto; a imagem simplificada basta para o que se segue.)

Como os protocolos clássicos lidam realmente com as células das pistas

O truque de renomear tem um ponto cego. As regras das linhas, colunas e caixas dizem todas «estas células são todas diferentes», e *todas diferentes* sobrevive a qualquer renomeação dos símbolos. Mas uma pista diz «esta célula contém exatamente 5» e, depois da renomeação, Bob só vê $\sigma(5)$ — algum símbolo mascarado — sem conhecer a renomeação σ. Não consegue verificar nada. Sem correção, Alice poderia provar que existe *alguma* grelha válida ignorando completamente as pistas impressas, o que nada provaria sobre *este* puzzle. A literatura clássica tem duas correções padrão.
A paleta. Acrescente-se uma linha extra de N células à grelha escondida — uma paleta que Alice preenche com os símbolos 1…N numa ordem pública fixa e depois renomeia juntamente com tudo o resto, ficando com $\sigma(1)$…σ(N). O desafio aleatório de Bob passa a ter mais uma opção. Além de escolher uma linha, coluna, caixa ou *gadget* para abrir, pode escolher **a paleta mais uma célula com pista**. Alice revela ambas; a paleta mostra a renomeação dessa ronda, e Bob verifica que a célula da pista contém exatamente a versão renomeada da pista impressa. Isto continua a ser conhecimento zero porque Bob aprende apenas σ — escolhido de novo aleatoriamente em cada ronda e inútil por si só — e o valor de uma célula que já conhecia através do puzzle. Nada sobre as células secretas é revelado, e um simulador pode imitar a vista escolhendo um σ aleatório. É correto porque uma Alice desonesta é apanhada com uma

probabilidade fixa em cada ronda, e as rondas são repetidas até a dúvida se tornar desprezável.
Compilar as pistas para fora do problema. Uma variante mais estrutural elimina o desafio especial em vez de o acrescentar. Em vez de *verificar* o valor da pista, força-o com restrições de diferença: ligue a célula da pista a todas as células da paleta exceto aquela que transporta o seu próprio valor — «diferente de $\sigma(1)$, diferente de $\sigma(2)$, …, diferente de tudo menos $\sigma(5)$». O único símbolo que a célula pode legalmente conter é o da pista. Todas as restrições voltam assim a ser do tipo «estas duas diferem» — invariantes à renomeação e verificáveis exatamente como uma linha. É a mesma manobra usada para vértices pré-coloridos no protocolo clássico de coloração de grafos, e é o espírito da palavra *gadgets* acima: na imagem MegaSudoku-como-SAT, as pistas são compiladas em *gadgets* de desigualdade como qualquer outra restrição.
O protocolo físico. O protocolo real com cartas para Sudoku (Gradwohl, Naor, Pinkas e Rothblum, 2007) não usa qualquer renomeação e resolve as pistas antes sequer de começar a ocultação. Para cada célula, Alice coloca três cartas idênticas com o valor da célula — viradas para baixo nas células secretas, mas **viradas para cima nas células com pistas**, para Bob ver diretamente que as pistas são respeitadas antes de as cartas serem viradas. Depois, uma carta de cada célula vai para o maço da respetiva linha, uma para o da coluna e uma para o da caixa; cada maço é baralhado e revelado, e Bob verifica que contém todos os N símbolos. Baralhar destrói a informação de posição (é aí que está o conhecimento zero), mas as pistas já tinham sido fixadas no momento de distribuir as cartas.
Em qualquer das variantes, a lição é a mesma a que este artigo regressa repetidamente: um protocolo de conhecimento zero é uma contabilidade cuidadosa de *quais* factos sobrevivem à ocultação. Renomear preserva «todos diferentes» e apaga «igual a 5» — por isso «igual a 5» tem de ser reintroduzido por outros meios.

Este não é o protocolo do artigo científico. É apenas o modelo mental do conhecimento zero clássico:

- Alice e Bob trocam mensagens.
- Bob escolhe verificações aleatórias.
- Alice revela apenas consistência local, não a solução completa.
- A prova da privacidade funciona mostrando que a visão de Bob poderia ter sido gerada sem a solução secreta de Alice.

Assim, o conhecimento zero clássico assenta num facto positivo:

> Um simulador existe realmente.

Agora retire-se o conforto. Alice envia uma única sequência de prova e vai-se embora. Não existe configuração de confiança, não existe cadeia aleatória partilhada preparada antecipadamente, e Bob nunca pode aceitar um puzzle falso. É o cenário em que o conhecimento zero clássico não consegue sobreviver.

Falta uma personagem antes do truque. Fixe-se um **livro de regras**: um sistema formal de prova, no sentido dos lógicos — um conjunto fixo de axiomas mais regras mecânicas para verificar provas matemáticas escritas. ZFC, os axiomas padrão da matemática, é o exemplo canónico. Daqui em

diante, tudo é formulado relativamente a um livro de regras escolhido antecipadamente, e a escolha é flexível: a construção funciona para qualquer livro de regras que se fixe, incluindo ZFC.

(Uma nota sobre palavras, tomada do próprio artigo: «proof system» significa aqui sempre este livro de regras — o sistema formal que verifica provas matemáticas — nunca as mensagens enviadas por Alice. A maquinaria de Alice e Bob chama-se «o provador e o verificador».)

A versão ao estilo de Gödel mantém a história do mega-Sudoku, mas muda a prova.

Escolha-se um segundo sistema de restrições do mesmo tamanho apresentado, chamemos-lhe **D**. Na história, S e D são dois puzzles MegaSudoku(n) no mesmo formato. Nos bastidores, D pode ter começado como uma fórmula lógica difícil de tamanho diferente; se necessário, pode ser preenchida com restrições fictícias inofensivas para caber na mesma grelha. D é construída a partir de uma fórmula lógica que é realmente **insatisfazível**: não existe qualquer atribuição de valores capaz de tornar verdadeiras todas as suas restrições, tal como um puzzle estragado não tem uma grelha completa legal. Um exemplo de brinquedo seria uma fórmula que exige simultaneamente «X é verdadeiro» e «X é falso». Portanto, D não tem preenchimento válido.

Mas D não pode ser um puzzle estragado cuja falha seja *fácil de expor*. O exemplo de brinquedo acima falha aqui: qualquer livro de regras refuta «X e não-X» numa linha. D tem de ser falsa de uma forma que o livro de regras escolhido não consiga certificar com um argumento curto. Se o livro de regras pudesse refutar D com uma prova curta, a história abaixo colapsaria: a via alternativa que poderia ter produzido provas sem o segredo de Alice poderia ser formalmente excluída, e com ela desapareceria a garantia de privacidade. Por isso, D é escolhida de uma família que o livro de regras fixado não consegue refutar eficientemente: dentro desse livro de regras, não existe uma prova curta de que D não tem solução.

A prova de uma mensagem de Alice passa então a ser sobre uma afirmação «ou/ou»:

> ou o mega-Sudoku real S tem solução, ou o chamariz D tem solução.

Este é o elo lógico. D **não** é gerada de forma mágica para tornar S verdadeira. A prova não está a argumentar «D não tem solução, logo S tem solução». Está a provar a disjunção **S ou D**. Correção perfeita significa que uma disjunção falsa não pode ter uma prova válida. Como D é falsa na realidade — não tem solução — a única forma de a disjunção ser verdadeira é S ser verdadeira. Portanto, se a prova for aceite, S tem de ter solução. O chamariz não consegue tornar verdadeira uma S falsa.

Mas para a parte semelhante a conhecimento zero, pergunte-se o que aconteceria *se* D tivesse uma solução. Essa solução do chamariz funcionaria como testemunha alternativa. Permitiria a alguém produzir provas sem conhecer a verdadeira solução do mega-Sudoku de Alice — por outras palavras, um simulador. Na realidade, D não tem solução, pelo que esta via para o simulador está fechada. O ponto é que o livro de regras não consegue provar eficientemente que ela está fechada.

Assim, D tem duas funções. Para a **correção**, D é falsa, pelo que uma prova válida de «S ou D» força S. Para o **conhecimento zero efetivo**, D é difícil de refutar, pelo que o livro de regras não consegue excluir rapidamente a via do chamariz que teria tornado a simulação possível.

O teste de segurança deixa então de ser:

> Conseguimos provar que existe realmente um simulador?

E passa a ser:

> O seu livro de regras consegue provar eficientemente que o simulador é impossível?

Se a resposta for não, segue-se algo surpreendentemente forte: toda a garantia de segurança que (a) possa ser observada executando um teste e (b) decorra demonstravelmente — dentro desse livro de regras — da existência de um simulador, é efetivamente satisfeita. Um ataque bem-sucedido a qualquer uma dessas garantias equivaleria à refutação curta que falta, e essa refutação curta não existe. É esta a parte «efetiva» do conhecimento zero efetivo.

Assim, o contraste de sala de aula é:

Conhecimento zero clássico: as provas são seguras porque existe um simulador.

Conhecimento zero efetivo ao estilo de Gödel: as provas são tratadas como seguras para testes de segurança observáveis porque o livro de regras não consegue provar eficientemente que o simulador é impossível.

A segunda afirmação é mais fraca. E é precisamente por isso que o artigo consegue conservar as três propriedades que destruíam a versão clássica: uma mensagem, nenhuma configuração e correção perfeita.

O novo teste: não se consegue provar que o simulador está ausente

A flexibilização de Ilango muda a pergunta.

O conhecimento zero clássico pergunta:

> Existe um simulador?

O conhecimento zero efetivo pergunta algo mais fraco:

> O livro de regras escolhido consegue provar eficientemente que não existe simulador?

Pode soar a manobra técnica, mas é a ideia central. A construção vive num estado estranho: um simulador não existe de facto — o artigo é explícito sobre isto — mas o livro de regras fixado não consegue provar eficientemente que não existe. Se todas as consequências más que nos preocupam exigissem uma refutação desse tipo, o sistema continuaria a comportar-se como conhecimento zero relativamente a essas consequências.

É aqui que entra Gödel. Não como decoração e não como «Gödel torna a criptografia segura». A ligação é de teoria da prova. Um livro de regras chama-se **ótimo** se for, num sentido preciso, o melhor possível: sempre que qualquer livro de regras consiga refutar uma fórmula do tipo relevante com uma prova curta, o livro de regras ótimo também consegue, com uma prova no máximo polinomialmente mais longa. Krajíček e Pudlák conjecturaram em 1989 que **não existe um sistema de prova ótimo**: seja qual for o livro de regras fixado, algum outro livro de regras prova de forma muito mais sucinta alguma família de afirmações verdadeiras. É uma das conjecturas centrais em aberto na complexidade de provas, e é uma prima finita, em teoria da complexidade, do teorema da incompletude de Gödel: algumas afirmações verdadeiras não têm uma prova curta no livro de regras fixado — não porque sejam indemonstráveis em princípio, mas porque todo o livro de regras fixo deixa algumas verdades curtas sem provas curtas.

O artigo assume esta conjectura (numa forma «infinitely often» ligeiramente mais forte, padrão quando conjecturas são usadas criptograficamente). A recompensa, por um teorema de Krajíček e Pudlák, é concreta: para cada livro de regras existe uma sequência de fórmulas genuinamente insatisfazíveis que esse livro não consegue refutar com provas curtas — e que, crucialmente, um algoritmo eficiente consegue *gerar*. Esta última propriedade, uniformidade, é o que transforma toda a ideia de uma afirmação de existência num algoritmo que Alice pode realmente executar: os chamarizes D saem de uma linha de montagem, não aparecem do nada.

O movimento criptográfico consiste em pôr essa falta de poder de prova a trabalhar.

O que a construção está a fazer

Eis a construção do artigo reduzida à sua forma essencial.

Fixe-se um livro de regras — ZFC, por exemplo. Sob a hipótese de complexidade de provas, existe uma sequência gerável eficientemente de fórmulas que são realmente insatisfazíveis, mas para as quais o livro de regras não tem provas curtas de insatisfazibilidade.

Construa-se agora uma prova de uma mensagem com esta forma:

> ou a afirmação real é satisfazível, ou esta fórmula especial difícil é satisfazível.

A fórmula especial difícil não é satisfazível. Portanto, se a maquinaria de prova subjacente tiver correção perfeita, aceitar a mensagem continua a significar que a afirmação real é verdadeira. É daí que vem a correção perfeita.

Mas, para a segurança semelhante a conhecimento zero, imagine-se que a fórmula especial difícil *fosse* satisfazível. A sua testemunha poderia então ser usada para simular provas sem conhecer a testemunha real. Na realidade, a fórmula não é satisfazível — mas o livro de regras não consegue provar isso eficientemente. Por isso, não consegue provar eficientemente que o simulador é impossível.

Essa é a dobradiça. O sistema não esconde o segredo produzindo um simulador clássico. Para uma grande classe de testes de segurança observáveis, esconde-o por detrás da incapacidade do livro de

regras para certificar que o simulador está ausente.

O que o artigo afirma

O teorema principal vem em camadas. O resultado central é este:

Sob uma hipótese criptográfica padrão — a existência de **provas não interativas com indistinguibilidade de testemunhas** (*non-interactive witness indistinguishable proofs*), objetos bem estudados que decorrem de vários conjuntos de hipóteses estabelecidos — e sob a conjectura de complexidade de provas de que **não existe um sistema de prova ótimo (infinitely often)**, o artigo constrói, para qualquer livro de regras escolhido, um provador e verificador de uma mensagem para NP/SAT, com **correção perfeita** e sem configuração, que é efetivamente de conhecimento zero relativamente a esse livro de regras. (NP/SAT é o «denominador comum mais difícil» padrão dos problemas semelhantes a puzzles; mega-Sudoku é apenas um dos seus disfarces.)

Para a afirmação mais ampla sobre preservar propriedades de segurança falsificáveis, o artigo acrescenta mais uma hipótese padrão, a crença de desaleatorização **P = BPP** (grosseiramente: a aleatoriedade não dá aos algoritmos um poder adicional essencial).

Traduzido da linguagem do teorema:

- A prova é uma única mensagem.
- Não existe configuração de confiança.
- Afirmações falsas não podem ser provadas.
- O provador não é de conhecimento zero clássico — não tem simulador.
- Mas cada consequência de segurança falsificável e baseada em jogos do conhecimento zero clássico pode ser alcançada neste cenário.

«Falsificável» importa. Significa que uma falha de segurança pode ser testada executando um adversário num jogo. Muitas definições de segurança criptográfica têm esta forma: consegue o adversário distinguir duas cifras, inverter uma função, recuperar uma testemunha ou ganhar uma experiência especificada? O teorema fornece um provador para cada propriedade falsificável, uma de cada vez. Um único provador que usufrua de *todas* as propriedades falsificáveis em simultâneo é provavelmente impossível — o antigo ataque de reutilização («Bob pode mostrar a prova a outras pessoas») é, ele próprio, uma propriedade falsificável, e aqui falha realmente. A proposta do artigo é que um único provador pode plausivelmente cobrir todas as propriedades falsificáveis *naturais* — aquelas que aparecem de facto na prática criptográfica — mas essa parte é um teorema condicional apoiado numa noção informal de «natural», além de uma conjectura explícita. A garantia visa falhas observáveis, não todos os significados filosóficos ou baseados em simulação de segredo.

Vale a pena nomear um corolário concreto: a construção produz as primeiras provas não interativas de *witness hiding* com um provador uniforme — «uma prova de um puzzle não ajuda a encontrar a sua solução», sem interação e sem configuração — um objeto aparentemente modesto cuja construção tinha resistido durante décadas.

O que isto não diz

Esta é a secção que mantém o texto honesto.

Não diz que os antigos teoremas de impossibilidade estavam errados. A construção contorna-os alterando a definição.

Não fornece conhecimento zero clássico comum, sem interação, sem configuração e com correção perfeita. O artigo diz explicitamente que o provador construído não tem simulador.

Não significa que a prova não possa ser reutilizada. Uma prova de uma mensagem continua a poder ser mostrada a outra pessoa; o artigo não preserva propriedades do tipo negabilidade. (Conhecimento zero não interativo com configuração de confiança tem a mesma limitação.)

Não significa que seja um protocolo prático pronto a instalar. Estamos em teoria da complexidade e fundações da criptografia. O resultado depende de hipóteses importantes de complexidade de provas e criptografia, e a construção diz respeito ao que é possível em princípio.

Não transforma «Gödel» numa primitiva mágica de segurança. A ligação a Gödel passa por sistemas de prova, sistemas de prova ótimos e análogos finitos da incompletude. A intuição útil não é «a incompletude protege a sua palavra-passe». É esta: se um livro de regras não consegue provar eficientemente que um simulador é impossível, então ataques que exigiriam essa prova podem ser bloqueados ao nível das definições de segurança.

Porque é interessante na mesma

A criptografia transforma frequentemente dificuldade em segurança. Fatorizar é difícil, por isso hipóteses ao estilo RSA tornam-se úteis. Problemas em reticulados são difíceis, por isso a criptografia baseada em reticulados torna-se útil. Aqui, a dificuldade é mais estranha: não é «difícil calcular um segredo», mas «difícil provar que um determinado objeto de prova não pode existir».

É por isso que o artigo parece invulgar. Trata axiomas e livros de regras quase como recursos criptográficos. A impossibilidade habitual diz que existe tensão entre correção e simulação. O movimento de Ilango consiste em colocar essa tensão atrás de uma cortina de teoria da prova: o simulador está ausente, mas o sistema formal não consegue expor eficientemente essa ausência.

Para um leitor, o surpreendente não é isto vir substituir os sistemas de conhecimento zero atuais. Provavelmente não substituirá, pelo menos não diretamente. O surpreendente é uma limitação da lógica matemática poder ser usada de forma construtiva: não apenas como uma parede, mas como uma espécie de cobertura.

Quão forte é a evidência?

Este é um artigo de teoremas, por isso «evidência» significa algo diferente do que num artigo de biologia ou astronomia. A pergunta não é se uma experiência foi replicada. É se as definições, hipóteses e cadeia de demonstração sustentam a afirmação.

A prova é formal e o artigo é explícito quanto às suas hipóteses. Não são hipóteses casuais. Provas não interativas com indistinguibilidade de testemunhas são objetos padrão em criptografia e decorrem de vários conjuntos estabelecidos de hipóteses. A conjectura de inexistência de um sistema de prova ótimo é uma conjectura central da complexidade de provas. P = BPP é uma crença padrão de desaleatorização usada apenas para o teorema mais amplo sobre propriedades falsificáveis.

O artigo argumenta ainda que as hipóteses são o preço certo, não um andaime arbitrário: prova uma implicação inversa mostrando que são essencialmente necessárias — se construções deste tipo existirem de todo, então têm de existir provas não interativas com indistinguibilidade de testemunhas e, assumindo funções de sentido único padrão, não pode existir um sistema de prova ótimo. E as hipóteses são «ganha-se de qualquer forma»: refutar qualquer uma delas seria, por si só, uma descoberta marcante em complexidade de provas, criptografia ou teoria da complexidade.

Mas, como o resultado é condicional, a confiança também o é. Se essas hipóteses falharem, a interpretação do teorema muda. E mesmo que se mantenham, a garantia não é conhecimento zero clássico completo; é a versão relaxada e baseada em teoria da prova proposta no artigo.

A confiança adequada é, portanto, elevada quanto ao facto de o artigo estabelecer um resultado condicional coerente de possibilidade; moderada quanto a as hipóteses descreverem o mundo criptográfico em que realmente vivemos; e baixa quanto a qualquer consequência prática imediata.

Porque importa

O artigo abre uma via que supostamente estava fechada.

A teoria clássica diz: conhecimento zero completo não pode ser uma única mensagem sem configuração e não pode ter correção perfeita. O artigo de Ilango diz: se pedirmos as consequências do conhecimento zero que podem ser testadas em jogos de segurança, e permitirmos que a definição de segurança dependa daquilo que um livro de regras consegue ou não refutar eficientemente, então muito do comportamento útil pode ser recuperado — com uma mensagem, sem configuração e com correção perfeita.

Não é uma pequena mudança de definição. É uma forma diferente de pensar sobre garantias criptográficas. Em vez de perguntar apenas o que existe, pergunte o que o seu livro de regras consegue excluir. Em vez de tratar a indemonstrabilidade como um incómodo filosófico, use-a como estrutura.

O mundo prático pode não mudar amanhã. Mas o mapa conceptual muda. Existe agora um sentido formal em que «ninguém consegue provar eficientemente que o segredo vazou» pode ser suficientemente forte para recuperar muitas das proteções baseadas em jogos que queríamos de «o segredo

não vazou».

É por isso que Gödel pertence ao título.

Resumo claro

As provas de conhecimento zero permitem a um provador convencer um verificador de que uma afirmação é verdadeira sem revelar a testemunha. Resultados clássicos de impossibilidade dizem que o conhecimento zero não pode ser comprimido numa única mensagem sem configuração e não pode ter correção perfeita. O artigo de Rahul Ilango não refuta essas impossibilidades. Define uma noção mais fraca, conhecimento zero efetivo: em vez de exigir que exista realmente um simulador, exige que um sistema de prova escolhido — um livro de regras formal como ZFC — não consiga provar eficientemente que não existe simulador. Sob hipóteses importantes da criptografia (provas não interativas com indistinguibilidade de testemunhas) e da complexidade de provas (não existe um sistema de prova ótimo), o artigo constrói provadores de uma mensagem para NP/SAT, sem configuração e com correção perfeita, que alcançam, propriedade a propriedade, as consequências falsificáveis e baseadas em jogos do conhecimento zero. Um único provador que cubra todas as propriedades «naturais» deste tipo é uma extensão adicional e em parte conjectural — e cobrir literalmente todas as propriedades falsificáveis é provavelmente impossível, porque as provas continuam reutilizáveis. O resultado é teórico e condicional, não uma primitiva implantada, mas mostra uma nova forma de usar a indemonstrabilidade da teoria da prova como recurso criptográfico.

Verificação sem exageros

O que o artigo mostra: Sob as hipóteses declaradas, é possível construir provadores de uma mensagem para NP/SAT, sem configuração e com correção perfeita, que são efetivamente de conhecimento zero relativamente a qualquer sistema de prova escolhido e que alcançam cada consequência de segurança falsificável e baseada em jogos do conhecimento zero clássico.

O que é plausível, mas não está demonstrado incondicionalmente: Que as hipóteses necessárias de complexidade de provas e criptografia sejam verdadeiras. São hipóteses sérias e muito estudadas — e o artigo mostra que são essencialmente necessárias, além de suficientes — mas continuam a ser hipóteses.

O que não mostra: Conhecimento zero clássico sem interação, sem configuração e com correção perfeita; um sistema prático pronto a instalar; negabilidade ou não reutilização das provas; nem que o teorema da incompletude de Gödel, por si só, torne a criptografia segura.

Principais limitações: A garantia é uma relaxação do conhecimento zero; a versão mais ampla depende de várias hipóteses; as afirmações sobre um único provador universal continuam em parte conjecturais; e o resultado é sobretudo fundacional.

Que grau de confiança deve ter um leitor não especialista? Elevado quanto a este ser um resultado

teórico condicional importante, aceitando as definições. Moderado quanto a as hipóteses captarem a realidade. Baixo quanto à aplicação prática imediata. A conclusão segura é: o artigo não quebra as impossibilidades do conhecimento zero; encontra uma nova forma, baseada em teoria da prova, de contornar as partes dessas impossibilidades que importam em muitos jogos de segurança.

Fontes

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`