The Clean Paper

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

暗号証明は、「存在しないシミュレータ」を証明しにくくすることで秘密を隠せる

ゼロ知識証明では、*witness* を明かさずに主張を証明できる。古典理論によれば、その完全な意味でのゼロ知識を、一通のメッセージ、信頼されたセットアップなし、完全健全性という条件で同時に得ることはできない。*Rahul Ilango* の論文はこの不可能性を消してはいない。狙いを変える。シミュレータが本当に存在することを要求する代わりに、選んだ証明体系が「シミュレータは存在しない」と効率よく証明できないことを要求する。証明複雑性と暗号学の大きな仮定の下では、それでゼロ知識の反証可能なゲームベース帰結を、性質ごとに回収できる。これはそのままインターネットへ組み込むプリミティブではない。数学的な証明不能性を暗号上の覆いへ変える、証明論的な方法である。

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

秘密が隠れていることを証明するのが仕掛けではない

まず、ゼロ知識の最も単純な形から始めよう。

Alice は、ある数独に解があると Bob へ納得させたい。解そのものを送れば Bob は納得するが、パズルは台無しになる。Alice が欲しいのはもっと奇妙なものだ。解を明かさずに、解が存在することだけを証明する。

それがゼロ知識証明の約束だ。証明者（prover、Alice）は、検証者（verifier、Bob）に対し、ある主張が真であることを納得させる。ただし、その主張が真であるという事実以上のものは何も明かさない。

問題は、この約束には代償があることだ。普通の数学的証明には、心地よい二つの特徴がある。**一通のメッセージ**で済む。書いて渡し、その場を離れられる。そして**完全健全性（perfect soundness）**がある。偽の主張には、そもそも有効な証明が一つも存在しない。古典的な不可能性結果によれば、ゼロ知識はこの二つを諦めなければならない。しかも「二つを同時には持てない」だけではなく、それぞれ単独でも禁じられる。

第一に、ゼロ知識証明には対話が必要だ。事前に信頼できるセットアップを用意せず、Alice が一通だけ送るなら、ゼロ知識保証は崩れる。代わりに健全性をどれだけ犠牲にしても、この点は変わらない。

第二に、ゼロ知識証明には、ごく小さな誤り許容が必要だ。完全健全性を要求すると、実は静かに対話まで壊してしまう。どんな乱数を選んでも絶対に騙されない検証者なら、その乱数を最初から固定しても同じだ。そして検証者の行動が予測可能になれば、Alice はすべての返答を一通のメッセージへまとめられる。まさに、すでに壊れると分かっている場合へ戻ってしまう。

Rahul Ilango の論文は、この二重の壁を回り込む方法についての研究だ。壁が存在しないふりをするのでも、不可能な条件下で古典的ゼロ知識を実現するのでもない。動きはもっと微妙だ。「何も明かさない」の意味を弱める。ただし、暗号研究者が実際にテストできるセキュリティ性質を保つ形で弱める。

その概念を**実効的ゼロ知識（effectively zero-knowledge）**と呼ぶ。

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.

ゼロ知識には三つの扉 —— 対話、信頼できるセットアップ、不完全な健全性 —— が立ちはだかる。Ilango の構成は別の扉を通る。選んだ形式的証明体系が、シミュレータの不在を効率よく証明できないという扉だ。

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.

古典的ゼロ知識は「シミュレータが存在するか」を問う。実効的ゼロ知識は、「選んだ証明体系が、シミュレータは存在しないと効率よく証明できるか」だけを問う。この弱い問いに変えることで、一通のメッセージ、

セットアップなし、完全健全性を同時に維持できる。

Original diagram —The Clean Paper CC BY 4.0

従来のテスト：シミュレータが存在する

ゼロ知識を古典的に形式化するとき、**シミュレータ（simulator）**という架空の助っ人を使う。

考え方はこうだ。Alice の秘密を**知らない** Jane を想像する。もし Jane が Alice の秘密を一切知らないまま、Bob が Alice から受け取ったものと見分けのつかない証明を自力で生成できるなら、Alice の証明は Bob に新しい情報を教えていない。Jane は秘密なしで、同じ体験を偽造できたのだから。

したがって古典的ゼロ知識は、本物のシミュレータを要求する。秘密 —— 専門用語でいう**証拠（witness）** —— を知らなくても、本物らしい証明を効率よく生成できるアルゴリズムが実際に存在しなければならない。数独なら、この証拠は単に完成した解答盤面だ。

この定義は強力だが、古い不可能性が噛み付く場所もまさにここである。直感はこうだ。本当に非対話型の証明は、ただの文字列になる。Bob がその文字列を受け取れば、他人にも見せられる。つまり Bob は、主張を第三者へ証明する能力を得たことになる。それはすでに「何も得ていない」以上のものに聞こえる。古典的定理は、この直感を上の不可能性へ厳密化する。

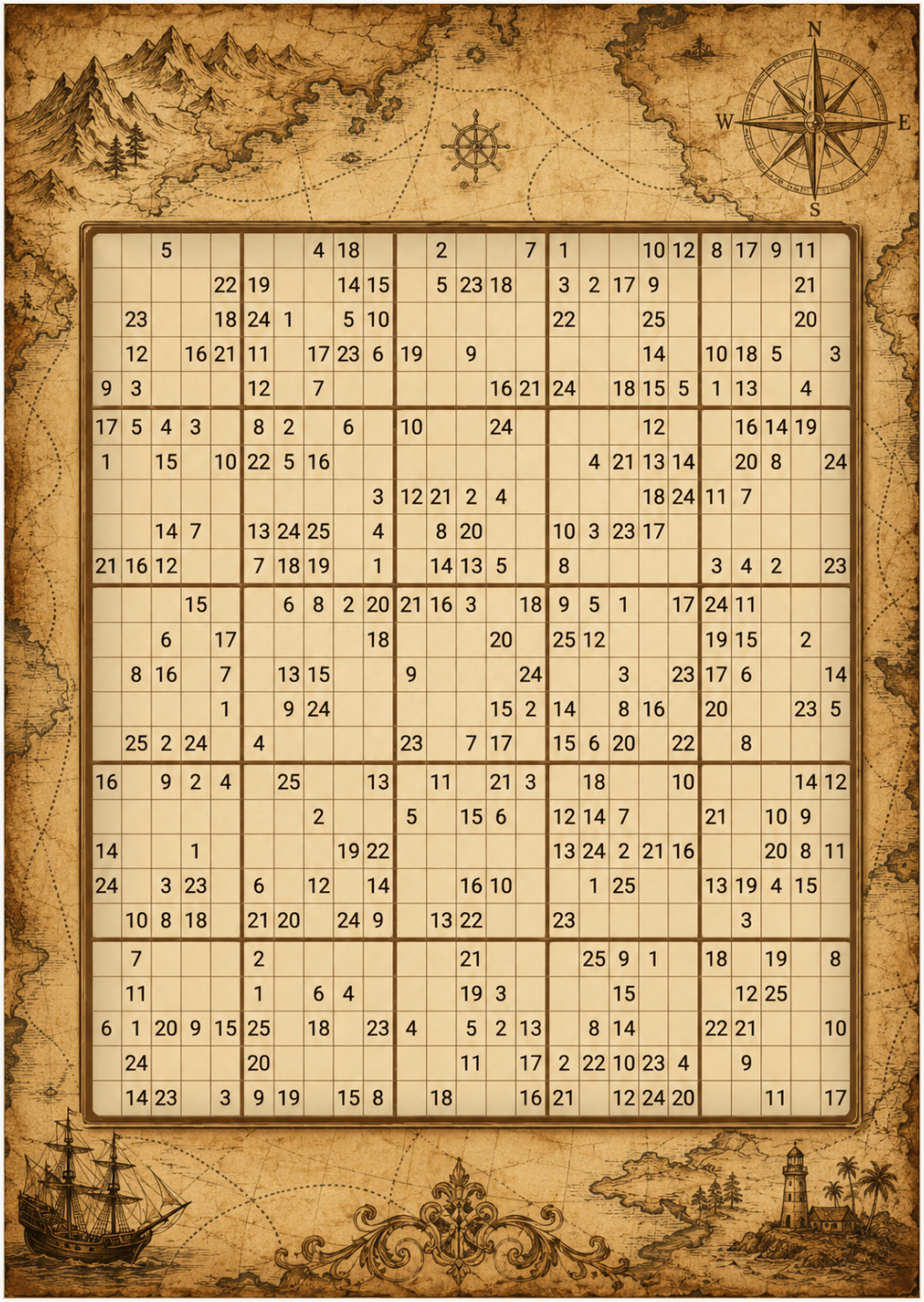
> **この論文が譲らない三つの性質**
>
> 論文のタイトルは、三つの制約を掲げている。
> **対話なし：** Alice は証明文字列を一通だけ送る。往復するプロトコルはない。
> **セットアップなし：** Alice と Bob は、信頼された共通参照文字列や、事前に用意された公開乱数に依存しない。「非対話型ゼロ知識」と呼ばれる方式の多くもセットアップを必要とするが、この論文が意味するのはセットアップがゼロということだ。
> **完全健全性：**偽の主張には有効な証明が存在しない。「ほとんど受理されない」ではなく、有効な証明が一つもない。
> この三つは、普通の書かれた数学的証明がそのまま持つ性質である。そして先ほど述べたとおり、古典的ゼロ知識は三つを維持できない。

メガ数独で感じる違い

違いを掴むため、意図的に単純化した例を使おう。

本気の比喩に普通の 9×9 数独を使ってはいけない。小さすぎ、有限すぎる。コンピュータなら解いてしまうか、解がないことを証明できる。代わりに、**MegaSudoku(n)** というパズル族を想像する。通常の規則を拡大する。ブロックサイズを n とし、$N = n^2$ と置く。そして $N \times N$ の盤面を $n \times n$ のブロックへ分け、N 種類の記号を使う。普通の数独は、ごく小さな $n = 3$、$N = 9$ の場合 ——9×9 盤面、3×3 ブロック、9 記号 —— にすぎない。証明複雑性の話が始まるのは、n を大きくでき、さらに

盤面へ追加の「ガジェット」を組み込み、SAT 式を数独の衣装で表せるようになってからだ。SAT 式は、要するに yes/no 制約のリストである。変数へ真／偽を割り当て、すべての制約を同時に満たせるか、という問題だ。



		5					4	18			2			7	1			10	12	8	17	9	11	
				22	19			14	15		5	23	18		3	2	17	9					21	
	23			18	24	1		5	10						22			25					20	
	12		16	21	11		17	23	6	19		9						14		10	18	5		3
9	3				12		7						16	21	24		18	15	5	1	13		4	
17	5	4	3		8	2		6		10			24					12			16	14	19	
1		15		10	22	5	16									4	21	13	14		20	8		24
									3	12	21	2	4					18	24	11	7			
		14	7		13	24	25		4		8	20			10	3	23	17						
21	16	12			7	18	19		1		14	13	5		8					3	4	2		23
			15			6	8	2	20	21	16	3		18	9	5	1		17	24	11			
		6		17					18				20		25	12				19	15		2	
	8	16		7		13	15			9				24			3		23	17	6			14
				1		9	24						15	2	14		8	16		20			23	5
	25	2	24		4					23		7	17		15	6	20		22		8			
16		9	2	4		25			13		11		21	3		18			10				14	12
							2			5		15	6		12	14	7			21		10	9	
14			1					19	22						13	24	2	21	16			20	8	11
24		3	23		6		12		14			16	10			1	25			13	19	4	15	
	10	8	18		21	20		24	9		13	22			23						3			
	7				2							21				25	9	1		18		19		8
	11				1		6	4				19	3				15				12	25		
6	1	20	9	15	25		18		23	4		5	2	13		8	14			22	21			10
	24				20							11		17	2	22	10	23	4		9			
	14	23		3	9	19		15	8		18			16	21		12	24	20			11		17

25×25 の数独。完成盤面という証拠そのものを明かさなくても、規則が守られているかを検査できる。隠れた解を検証する証明の視覚的な代役である。

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

数独と SAT：同じパズルを二つの衣装で

数独が「SAT 式のように振る舞える」というのは比喩ではない。変換は両方向にでき、簡単な方向なら完全に書き下せる。

数独から SAT へ。SAT が扱うのは真／偽だけなので、（行、列、値）の各三つ組にブール変数を一つ与える。*x(r,c,v)* を「行 *r*、列 *c* のセルには値 *v* が入る」と読む。4×4 数独（2×2 ブロック、値 1〜4）なら 4·4·4 = 64 変数、普通の 9×9 なら 729 変数が必要だ。すると、数独の各規則を節（clause）の集合へ変えられる。（節とは変数またはその否定を OR でつないだもの。式全体はすべての節の AND になる。）

各セルには少なくとも一つの値が入る —— セルごとに一つの節：

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

各セルには高々一つの値しか入らない —— 値の各組について「両方は不可」の節：

¬x(1,1,1) ∨ ¬x(1,1,2)　¬x(1,1,1) ∨ ¬x(1,1,3)　… 6 組すべてについて同様。

各行にはすべての値が現れる ——1 行目の値 3 なら、少なくとも一度：

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

そして高々一度：¬x(1,1,3) ∨ ¬x(1,2,3)、以下、その行のセルの各組について同様。

列とブロック —— 同じ種類の節を使い、対象セル群だけを変える。左上ブロックの値 2 なら：

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

さらに各組について「両方は不可」の節を加える。

印刷済みのヒント数字 —— 最も簡単だ。各ヒントは変数一つだけの節になる。左上に印刷された 3 は、

x(1,1,3)

という節になる。

これらすべての AND は、数独に解があるとき、かつそのときに限って充足可能になる。そして充足割当そのものが解である。真になった x(r,c,v) を読み取り、盤面を埋めればよい。9×9 なら 729 変数と数千の節になり、現代の SAT solver ならミリ秒で処理する。ヒント節 x(1,1,3) に注目しよう。「このセルは**ちょうど 3**」と言っているのであって、「これらのセルはすべて異なる」とは言っていない。この非対称性が、後のプロトコル注記でヒントセル用の追加仕掛けを必要にする。

SAT から数独へ。論文が必要とするのは逆向きで、こちらのほうが難しい。任意の SAT 式を与え、それが充足可能なとき、かつそのときに限って解を持つメガ数独を作る。数独固有の規則は「このセル群はすべて異なる」としか言えないので、任意の論理制約を組み立てる必要がある。そこで使うのがガジェットだ。ガジェットは、式の各節に対応する小さな既製セル群で、指定されたセルが変数の役割を持ち（そこに入る記号が真／偽を符号化する）、内部制約を調整して、その節を満たす割当だけが合法的な埋め方になるよう作る。これは NP 完全性証明でおなじみの技巧で、一般化数独については Yato と Seta が 2003 年に示している。

両方向を合わせると、N×N 数独と SAT は、別の衣装を着た同じ問題だと言える。だからこの記事 —— そして論文 —— では、盤面と記号を使って NP 全体の話をしてよい。

この「証拠」という考え方は、ここでも直感的に描ける。Alice はメガ数独の完全に正しい完成盤面を知っている。Bob はそのような盤面が存在すると納得したいが、Alice はそれを明かしたくない。完成盤面を丸ごと送れば Bob は納得するが、秘密は失われる。

古典的ゼロ知識の版では、Alice と Bob は対話する。昔ながらの直感モデルの一つでは、伏せたタイルを使う。Alice は完成盤面を隠し、各ラウンドの前に記号を秘密裏に別名へ置換し、Bob にはランダムに選んだ局所制約一つ —— 行、列、ブロック、またはガジェット —— だけを検査させる。開いたセルの記号がすべて異なれば、Bob の確信は増す。その後すべてを再び隠し、記号を新しく置換する。（一つ問題がある。元から印刷されたヒントも記号置換で隠れてしまうため、別の仕掛けが必要になる。下の注記で古典的プロトコルの処理法を説明する。ここから先の理解には、この単純化した絵で十分だ。）

古典的プロトコルはヒントセルを実際にはどう扱うのか

記号置換には盲点がある。行、列、ブロックの規則はいずれも「これらのセルはすべて異なる」であり、すべて異なるという性質は記号をどう置換しても保たれる。しかしヒントは「このセルにはちょうど 5 が入る」と言う。置換後に Bob が見るのは $\sigma(5)$—— 何らかの覆われた記号 —— だけで、置換 σ を知らなければ何も確認できない。このままでは Alice は、印刷されたヒントを完全に無視して、何らかの合法盤面が存在すると証明できてしまう。それではこのパズルについて何も証明していない。古典文献には標準的な修正が二つある。

パレット。隠した盤面へ N セルの追加行を一つ加える。Alice はこのパレットへ 1…N を公開された固定順で置き、他のすべてと一緒に記号置換するので、パレットには $\sigma(1)$…$\sigma(N)$ が並ぶ。Bob のランダムな挑戦に、追加の選択肢を一つ増やす。行、列、ブロック、ガジェットのどれかを開く代わりに、**パレットとヒントセル一つ**を選んでもよい。Alice は両方を開く。パレットからそのラウンドの置換が分かり、Bob はヒントセルが印刷されたヒントの置換後の値を正確に示しているか確認する。ゼロ知識は保たれる。Bob が学ぶのは、そのラウンドごとに新しく無作為に選ばれ、単独では価値のない σ と、パズルからすでに知っている一セルの値だけだからだ。秘密セルの情報は漏れず、シミュレータもランダムな σ を選ぶだけで同じ見え方を偽造できる。不正な Alice は各ラウンドで一定確率で捕まり、疑いが無視できるほど小さくなるまでラウンドを繰り返すので健全性も得られる。

ヒントを制約へコンパイルして消す。より構造的な方法では、特殊な挑戦を追加する代わりに不要にする。ヒント値を検証するのではなく、不等制約で強制する。ヒントセルを、自分の値を持つもの以外のすべてのパレットセルへ結び付ける ——「$\sigma(1)$ とは違う、$\sigma(2)$ とは違う、…、$\sigma(5)$ 以外のすべてと違う」。すると、そのセルが合法的に持てる記号はヒント値だけになる。すべての制約が再び「この二つは違う」という形になり、記号置換に不変で、行と同じように検査できる。これは古典的なグラフ彩色プロトコルで事前彩色された頂点に使う操作と同じで、上でいうガジェットの精神でもある。MegaSudoku-as-SAT の絵では、ヒントも他の制約と同じように不等式ガジェットへコンパイルされる。

物理カードプロトコル。実世界の数独カードプロトコル（Gradwohl, Naor, Pinkas and Rothblum, 2007）は記号置換を使わず、隠す前にヒントを確定する。各セルについて Alice は、そのセルの値を持つ同じカードを 3 枚置く。秘密セルは裏向きだが、**ヒントセルだけは表向き**にする。したがってカードを伏せる前に、Bob はヒントが守られていることを自分の目で確認できる。次に、各セルから一枚をその行の束へ、一枚を列の束へ、一枚をブロックの束へ入れる。各束をシャッフルして公開し、Bob は N 種類すべての記号が入っているか確認する。シャッフルによって位置情報が失われる —— そこがゼロ知識 —— が、ヒントはカー

> ドを配った時点ですでに固定されている。
> どの方法でも教訓は同じで、この記事が何度も戻ってくる点でもある。ゼロ知識プロトコルとは、隠した後にもどの事実が残るかを慎重に記帳する仕組みだ。記号置換は「すべて異なる」を保ち、「5 である」を消す。したがって「5 である」は別の方法で戻さなければならない。

これは論文そのもののプロトコルではない。古典的ゼロ知識を理解するための思考モデルだ。

- Alice と Bob は往復する。
- Bob がランダムな検査を選ぶ。
- Alice は局所的な整合性だけを明かし、完全な解は明かさない。
- プライバシーの証明は、Bob が見たものが Alice の秘密の解なしでも生成できた、と示すことで成り立つ。

したがって、古典的ゼロ知識は肯定的な事実を中心に立っている。

シミュレータが本当に存在する。

では、心地よい部分を取り去ろう。Alice は一通の証明文字列を送り、その場を去る。信頼されたセットアップも、事前に用意された共有乱数列もなく、Bob は偽のパズルを絶対に受理してはならない。古典的ゼロ知識が生き残れない設定である。

仕掛けへ進む前に、もう一人だけ登場人物が必要だ。**形式的証明体系**を一つ固定する。直感のため、ここではこれを「ルールブック」と呼ぼう。固定された公理群と、書かれた数学的証明を機械的に検査する規則の集合である。数学の標準的公理系 ZFC が典型例だ。ここから先のすべては、あらかじめ選んだ証明体系に対して述べられる。選択自体は柔軟で、ZFC を含め、どの証明体系を固定しても構成は働く。

（用語上の注意：論文でいう「proof system」は、ここで「ルールブック」と呼んでいる形式的証明体系を指し、Alice が送る証明メッセージそのものではない。Alice と Bob 側の仕組みは、それぞれ**証明者（prover）**と**検証者（verifier）**と呼ばれる。）

Gödel 風の版はメガ数独の物語を維持し、証明の作り方だけを変える。

同じ表示サイズの第二の制約系を選び、**D** と呼ぼう。物語では S と D は同じ形式の二つの MegaSudoku(n) である。裏側では D は異なるサイズの難しい論理式から始まっていてもよく、必要なら無害なダミー制約を加えて同じ盤面サイズへ合わせられる。D は、実際には**充足不能（unsatisfiable）**な論理式から作る。すべての制約を真にする値の割当は存在せず、壊れたパズルに合法的な完成盤面がないのと同じだ。おもちゃの例なら、「X は真」と「X は偽」を同時に要求する式である。したがって D には有効な完成形がない。

しかし D は、簡単に壊れていると暴けるパズルではいけない。今のおもちゃ例は失格だ。どんな証明体系でも「X かつ ¬X」のような矛盾なら一行で反証できる。D は、偽でありながら、選んだ証明体系が短い議論ではその偽を証明できない形でなければならない。証明体系が D を短い証明で反証できれば、下の物語は崩れる。Alice の秘密なしで証明を作れたかもしれない代替経路が形式的に排除され、それとともにプライバシー保証も消える。そこで D は、固定された証明体系が効率よく反証できない族から選ぶ。その証明体系の内部には、「D に解はない」という短い証明が存在しない。

Alice が一通で送る証明は、次の「どちらか」の主張についてになる。

> 本物のメガ数独 S に解がある、または、おとり D に解がある。

ここが論理的な接続点だ。D は、魔法のように S を真にするために生成されるのではない。証明は「D に解がない、だから S に解がある」と論じているわけではない。**S または D** という選言そのものを証明する。完全健全性によれば、偽の選言には有効な証明は存在しない。現実には D は偽 —— 解なし —— なので、選言が真になる唯一の方法は S が真であることだ。したがって証明が受理されれば、S には必ず解がある。おとり D が、偽の S を真へ変えることはできない。

一方、ゼロ知識風の部分では、もし D に解があったら何が起きるかを考える。そのおとりの解が代わりの証拠として働き、Alice の本物のメガ数独解を知らなくても証明を作れる。つまりシミュレータになる。現実には D に解がないので、このシミュレーション経路は閉じている。重要なのは、証明体系が「閉じている」と効率よく証明できないことだ。

したがって D には二つの仕事がある。**健全性**については D が偽なので、「S または D」の有効な証明は S を強制する。**実効的ゼロ知識**については D の反証が難しいので、証明体系は「シミュレーションを可能にしたはずのおとり経路が閉じている」と素早く証明できない。

するとセキュリティテストは、もはや

> シミュレータが本当に存在すると証明できるか？

ではなく、

> あなたの証明体系は、シミュレータが不可能だと効率よく証明できるか？

になる。

答えが「できない」なら、驚くほど強いことが従う。(a) 実際にテストを走らせれば観測でき、(b) シミュレータの存在から、その証明体系の内部で証明可能な形で導かれるセキュリティ保証は、どれも実際に成立する。もしそのいずれかへの攻撃が成功すれば、その成功自体が、存在しないはずの短い反証を与えることになる。しかしその短い反証は存在しない。ここが「実効的」ゼロ知識と呼ばれる理由だ。

教室的に対比するとこうなる。

古典的ゼロ知識：シミュレータが存在するから証明は安全である。

Gödel 風の実効的ゼロ知識：シミュレータが不可能だと証明体系が効率よく証明できないので、観測可能なセキュリティテストについて証明は安全として振る舞う。

後者の主張は弱い。だからこそ、古典版では同時に保てなかった三つ —— 一通のメッセージ、セットアップなし、完全健全性 —— を維持できる。

新しいテスト：シミュレータが存在しないと証明できない

Ilango の緩和は問いを変える。

古典的ゼロ知識はこう問う。

> シミュレータは存在するか？

実効的ゼロ知識は、もっと弱いことを問う。

> 選んだ証明体系は、シミュレータが存在しないと効率よく証明できるか？

技術的な抜け道に聞こえるかもしれないが、ここが中心である。構成は奇妙な状態にある。シミュレータは実際には存在しない —— 論文は明記している —— が、固定した証明体系は、その不在を効率よく証明できない。もし気にする悪い結果がすべて、そのような反証を必要とするなら、その結果についてシステムは依然としてゼロ知識のように振る舞う。

ここで Gödel が入ってくる。飾りとしてでも、「Gödel が暗号を安全にする」という意味でもない。つながりは証明論的だ。証明体系は、厳密な意味で可能な限り「最良」であるとき**最適（optimal）**と呼ばれる。関連する種類の式について、どんな別の証明体系が短い証明で反証できるものでも、その最適な体系も高々多項式倍長い証明で反証できる、という意味だ。Krajíček と Pudlák は 1989 年に、**最適証明体系は存在しない**と予想した。どの証明体系を固定しても、ある真の主張の族について、別の証明体系ならはるかに短く証明できる。これは証明複雑性における中心的な未解決予想の一つであり、Gödel の不完全性定理の有限・計算量論的な親戚と見なせる。固定した証明体系には短い証明を持たない真の主張がある。ただし原理的に証明不能だからではなく、どの固定された証明体系にも「短い真理なのに短い証明がない」ものが残るからだ。

論文はこの予想を仮定する。暗号学でこの予想を使う際に標準的な、やや強い「無限回（infinitely often）」版である。Krajíček–Pudlák の定理による見返りは具体的だ。どの証明体系に対しても、本当に充足不能なのに、その証明体系では短く反証できず、しかも重要なことに、効率的なアルゴリズムで生成できる式の列が存在する。この最後の性質である**一様性（uniformity）**が、存在するだけの話を Alice が実際に走らせられるアルゴリズムへ変える。おとり D は空中から出てくるのではなく、組立ラインから出てくる。

暗号学上の動きは、この「証明能力の不足」をセキュリティへ使うことだ。

構成が実際にしていること

論文の構成を、形だけに削るとこうなる。

証明体系を一つ固定する。たとえば ZFC だ。証明複雑性の仮定の下では、実際には充足不能だが、その証明体系には「充足不能である」という短い証明がなく、しかも効率的に生成できる式の列が

ある。

そこで、一通の証明を次の形にする。

> 本物の主張が充足可能である、または、この特別な難しい式が充足可能である。

特別な難しい式は、実際には充足不能だ。したがって基礎となる証明機構が完全健全なら、このメッセージを受理することは依然として本物の主張が真であることを意味する。これで完全健全性が得られる。

一方、ゼロ知識風の安全性については、この特別な難しい式がもし充足可能だったならと想像する。その証拠を使えば、本物の証拠を知らなくても証明をシミュレートできる。現実には充足不能だが、証明体系はその事実を効率よく証明できない。したがって「シミュレータは不可能だ」とも効率よく証明できない。

ここが蝶番だ。システムは古典的シミュレータを実際に作って秘密を隠すのではない。大きなクラスの観測可能なセキュリティテストに対し、シミュレータの不在を証明体系が証明できないことの陰に秘密を置く。

論文が主張していること

主定理は段階に分かれる。核心となる結果はこうだ。

標準的な暗号仮定 —— 複数の確立した仮定群から得られる、よく研究された**非対話型証拠識別不能証明（non-interactive witness-indistinguishable proofs）**の存在 —— と、**（無限回という意味で）最適証明体系は存在しない**という証明複雑性の予想の下で、論文は、どの証明体系を選んでも、それに対して実効的ゼロ知識であり、**完全健全性**を持ち、セットアップを必要としない、NP/SAT 用の、一通のメッセージだけで動く証明者・検証者の仕組みを構成する。（NP/SAT はパズル的問題の標準的な「最難共通分母」で、メガ数独はその衣装の一つだ。）

さらに、反証可能なセキュリティ性質を広く維持するという主張には、標準的なもう一つの仮定、脱乱択化に関する信念 **P = BPP**（大ざっぱに言えば、乱数はアルゴリズムへ本質的な追加能力を与えない）を加える。

定理の言葉をほどくと：

- 証明は一通のメッセージ。
- 信頼されたセットアップはない。
- 偽の主張を証明できない。
- 証明者は古典的な意味ではゼロ知識ではない —— シミュレータが存在しない。
- しかし、古典的ゼロ知識から得られる反証可能な、ゲームベースのセキュリティ帰結を、この設定で一つずつ達成できる。

「反証可能（falsifiable）」が重要だ。セキュリティ失敗を、敵対者をゲームで走らせることでテストできるという意味である。暗号セキュリティ定義の多くはこの形をしている。敵対者は二つの暗号

文を区別できるか、関数を逆算できるか、証拠を回収できるか、指定された実験に勝てるか。定理は、反証可能な性質ごとに、それを満たす証明者を与える。

一つの証明者がすべての反証可能な性質を同時に満たすのは、おそらく不可能だ。古くからの再利用攻撃 ——「Bob が証明を他人へ見せられる」—— それ自体が反証可能な性質であり、ここでは本当に失敗する。論文は、一つの証明者で、暗号実務に現れるすべての自然な（*natural*）反証可能性質を覆える可能性を提案する。ただし、この部分は「natural」という非形式的概念と、明示された追加予想に依存する条件付き定理である。保証が狙うのは観測可能な失敗であって、秘密性について考え得るすべての哲学的、あるいはシミュレーションベースの意味ではない。

具体的な系として一つ名前を挙げる価値がある。この構成は、一様な証明者を持つ最初の**非対話型証拠秘匿（witness hiding）証明**を与える。「パズルの証明を見ても解を見つける助けにならない」を、対話なし、セットアップなしで実現する。控えめに聞こえる対象だが、何十年も構成できなかった。

この研究が言っていないこと

ここが記事を正直に保つ節だ。

古い不可能性定理が間違っていたとは**言っていない**。構成は定義を変えることでそれを回避する。

対話なし、セットアップなし、完全健全性を同時に持つ、普通の古典的ゼロ知識を与えたわけでは**ない**。論文は、構成された証明者にはシミュレータが存在しないと明記している。

証明を再利用できないという意味でも**ない**。一通メッセージの証明は、依然として他人に見せられる。論文は否認可能性（deniability）のような性質を保たない。（信頼されたセットアップを使う非対話型ゼロ知識にも同じ制約がある。）

実運用へすぐ投入できる実践的プロトコルという意味でも**ない**。これは計算量理論と暗号基礎論の研究だ。結果は証明複雑性と暗号学の大きな仮定に依存し、構成が示すのは原理的に何が可能かである。

「Gödel」が魔法のセキュリティプリミティブになったわけでも**ない**。Gödel とのつながりは、証明体系、最適証明体系、不完全性の有限版を通している。有用な直感は「不完全性がパスワードを守る」ではない。「形式的証明体系がシミュレータの不可能性を効率よく証明できないなら、その証明を必要とする攻撃をセキュリティ定義のレベルで阻止できる」である。

それでも面白い理由

暗号学は、しばしば「難しさ」を安全性へ変換する。因数分解が難しいので、RSA 型の仮定が役に立つ。格子問題が難しいので、格子暗号が役に立つ。ここでの難しさはもっと奇妙だ。「秘密を計算するのが難しい」ではなく、「ある証明オブジェクトが存在し得ないことを証明するのが難しい」。

そこがこの論文を変わったものにしている。公理や形式的証明体系を、ほとんど暗号資源のように

扱う。通常の不可能性は、健全性とシミュレーションの間に緊張があると言う。Ilango の動きは、その緊張を証明論的なカーテンの後ろへ置く。シミュレータは存在しないが、形式体系はその不在を効率よく暴けない。

読者にとって驚くべき点は、これが現在のゼロ知識システムを置き換えることではない。少なくとも直接には、おそらく置き換えない。驚きは、数理論理の限界を建設的に利用できることだ。壁としてだけでなく、一種の覆いとして使える。

証拠をどう評価するか

これは定理論文なので、「証拠」の意味は生物学や天文学の論文とは違う。実験が再現されたかを問うのではない。定義、仮定、証明の連鎖が主張を支えるかを問う。

証明は形式的で、論文は仮定を明示している。それらは軽い仮定ではない。非対話型証拠識別不能証明は暗号学で標準的に研究される対象で、複数の確立した仮定パッケージから導かれる。最適証明体系が存在しないという予想は、証明複雑性の中心的予想の一つだ。P = BPP は標準的な脱乱択化仮説で、より広い「反証可能性質」定理にだけ使われる。

さらに論文は、これらの仮定が任意に積み上げた足場ではなく、ほぼ必要な代償だと論じる。逆方向の結果を証明し、この種の構成が存在するなら非対話型証拠識別不能証明も存在しなければならず、さらに標準的な一方向関数を仮定すれば最適証明体系は存在できないことを示す。そして仮定には「win-win」の性質もある。どれかを反証できれば、それ自体が証明複雑性、暗号学、計算量理論における画期的発見になる。

ただし結果が条件付きなので、確信も条件付きだ。仮定が失敗すれば、定理の解釈は変わる。そして仮定が成立しても、保証は完全な古典的ゼロ知識ではなく、この論文の緩和された証明論的版である。

したがって適切な確信はこうだ。論文が整合的な条件付き可能性結果を確立していることへの確信は高い。その仮定が私たちの実際の暗号世界を記述していることには中程度。すぐに使える実践的帰結には低い。

なぜ重要なのか

閉じているはずだった経路を、論文は一つ開いた。

古典理論は言う。完全なゼロ知識は、セットアップなしの一通メッセージにはできず、完全健全にもできない。Ilango の論文はこう言う。セキュリティゲームでテストできるゼロ知識の帰結を求め、さらにセキュリティ定義が「形式的証明体系が何を効率よく反証できるか、できないか」に依存することを許せば、有用な振る舞いのかなりの部分を、一通のメッセージ、セットアップなし、完全健全性とともに回収できる。

これは小さな定義変更ではない。暗号保証を考える別の方法だ。何が存在するかだけを問うのでは

なく、自分が採用した証明体系で何を排除できるかを問う。証明不能性を哲学的な厄介者として扱うのではなく、構造として利用する。

実務の世界は明日変わらないかもしれない。しかし概念地図は変わる。「秘密が漏れなかった」から欲しかった多くのゲームベース保護を、「秘密が漏れたと誰も効率よく証明できない」という条件で回収できる、という形式的な意味が生まれた。

だから Gödel がタイトルに入る。

まとめ

ゼロ知識証明では、証明者が証拠（witness）を明かさずに、主張が真であることを検証者に納得させる。古典的不可能性結果によれば、ゼロ知識はセットアップなしで一通のメッセージへ圧縮できず、完全健全性も持てない。Rahul Ilango の論文は、その不可能性を反証しない。代わりに、実効的ゼロ知識（effectively zero-knowledge）という弱い概念を定義する。シミュレータが本当に存在することを要求するのではなく、ZFC のような選んだ証明体系 —— 形式的なルールブック —— が「シミュレータは存在しない」と効率よく証明できないことを要求する。暗号学（非対話型証拠識別不能証明）と証明複雑性（最適証明体系は存在しない）に関する大きな仮定の下で、論文は NP/SAT について、セットアップなし、完全健全性を持つ一通のメッセージだけで動く証明者を構成し、古典的ゼロ知識の反証可能なゲームベースの帰結を性質ごとに実現する。すべての「自然な」性質を一つの証明者で覆う拡張は、さらに一部予想的であり、文字どおりすべての反証可能性質を覆うのはおそらく不可能だ。証明は依然として再利用できるからである。結果は理論的・条件付きで、配備可能な暗号プリミティブではないが、証明論的な証明不能性を暗号資源として使う新しい方法を示している。

冷静に見ると

論文が示したこと：明示された仮定の下で、どの証明体系を選んでも、それに対して実効的ゼロ知識となり、古典的ゼロ知識の各反証可能なゲームベース帰結を達成する、一通のメッセージ、セットアップなし、完全健全性を備えた NP/SAT 用の証明者を構成できる。

もっともらしいが、無条件には証明されていないこと：必要な証明複雑性・暗号学上の仮定が成立すること。どれも真剣に研究されている仮定で、論文は十分条件だけでなく本質的に必要でもあることを示しているが、それでも仮定である。

示していないこと：対話なし、セットアップなし、完全健全性を持つ古典的ゼロ知識。実運用できるシステム。証明の否認可能性や再利用不能性。あるいは Gödel の不完全性定理そのものが暗号を安全にする、ということ。

主な限界：保証はゼロ知識の緩和版である。最も広い版は複数の仮定に依存する。一つの万能な証明者に関する主張は一部予想的なまま。結果の中心は基礎理論である。

一般読者はどの程度信頼すべきか？定義を受け入れるなら、重要な条件付き理論結果であることへ

の確信は高い。仮定が現実を正しく捉えていることには中程度。すぐ実用化できるという確信は低い。安全な結論はこうだ。この論文はゼロ知識の不可能性を壊したのではない。多くのセキュリティゲームで重要な部分を、証明論的な別の方法で回り込んだ。

出典

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