The Clean Paper

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

암호학적 증명은“없는 시뮬레이터”를 증명하기 어렵게 만들어 비밀을 숨길 수 있다

영지식 증명은 증인을 공개하지 않고도 명제를 증명하게 한다. 고전 이론은 상호작용도 신뢰 설정도 없고 완전 건전한 한 메시지 증명에서는 완전한 의미의 영지식을 얻을 수 없다고 말한다. *Rahul Ilango* 의 논문은 그 불가능성을 없애지 않는다. 대신 목표를 바꾼다. 시뮬레이터가 실제로 존재해야 한다고 요구하는 대신, 선택한 증명 체계가 시뮬레이터가 존재하지 않는다는 사실을 효율적으로 증명할 수 없어야 한다고 요구한다. 증명 복잡도와 암호학의 주요 가정 아래에서 이는 영지식의 반증 가능하고 게임 기반인 결과들을 속성별로 회복하기에 충분하다. 인터넷에 바로 꽂아 쓸 원시요소가 아니라, 수학적 비증명성을 암호학적 가림막으로 바꾸는 증명 이론적 방법이다.

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

비결은 비밀이 숨겨졌음을 증명하는 데 있지 않다

가장 단순한 영지식 (zero-knowledge) 부터 시작해 보자.

Alice 는 Sudoku 퍼즐에 해답이 있다는 사실을 Bob 에게 납득시키고 싶다. 해답 자체를 보내면 Bob 은 믿겠지만 퍼즐은 망가진다. Alice 가 원하는 것은 더 이상한 일이다. **해답을 공개하지 않고도 해답이 존재한다는 증명**이다.

이것이 영지식 증명의 약속이다. 증명자 (prover, Alice) 는 검증자 (verifier, Bob) 에게 어떤 명제가 참임을 납득시키면서 그 명제가 참이라는 사실 외에는 아무것도 드러내지 않는다.

문제는 이 약속에 대가가 따른다는 것이다. 보통의 수학적 증명에는 편안한 두 속성이 있다. 하나는 **메시지 하나**면 된다는 점이다. 써서 건네고 떠나면 끝이다. 다른 하나는 **완전 건전성 (perfect soundness)** 이다. 거짓 명제에는 유효한 증명이 아예 존재하지 않는다. 고전적인 불가능성 결과는 영지식이 이 두 속성을 모두 포기해야 한다고 말한다. 둘을 동시에 가질 수 없다는 정도가 아니다. 각각을 따로 요구해도 문제가 된다.

첫째, 영지식 증명에는 대화가 필요하다. 사전에 마련한 신뢰 설정 (trusted setup) 이 없는 상태에서 Alice 가 메시지 하나만 보낸다면 영지식 보장은 무너진다. 이 결과는 그 대가로 건전성을 얼마나 양보할 의향이 있는지와도 무관하다.

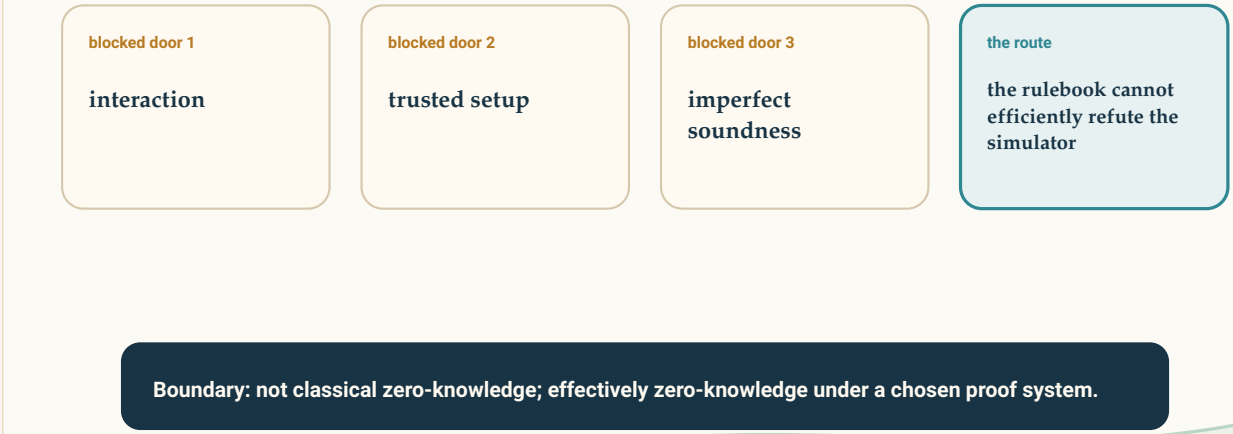
둘째, 영지식 증명에는 아주 작은 오류 허용이 필요하다. 완전 건전성을 요구하면 조용히 상호작용 자체가 사라진다. 어떤 무작위 선택을 하더라도 절대 속지 않는 검증자는 그 선택을 미리 고정해도 마찬가지다. 검증자가 예측 가능해지면 Alice 는 모든 답을 한 번에 메시지 하나로 보낼 수 있다. 바로 이미 불가능했던 경우다.

Rahul Ilango 의 논문은 이 이중 벽을 우회하는 방법을 다룬다. 벽이 없다고 우기는 것도 아니고, 불가능한 조건에서 고전적 영지식을 만들어 내는 것도 아니다. 움직임은 더 미묘하다. “아무것도 드러내지 않는다”의 뜻을 약하게 만들되, 암호학자가 실제로 시험할 수 있는 보안 속성은 보존하는 방식으로 약화한다.

그 결과를 **실효적 영지식 (effectively zero-knowledge)** 이라고 부른다.

A proof without leaking the witness

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

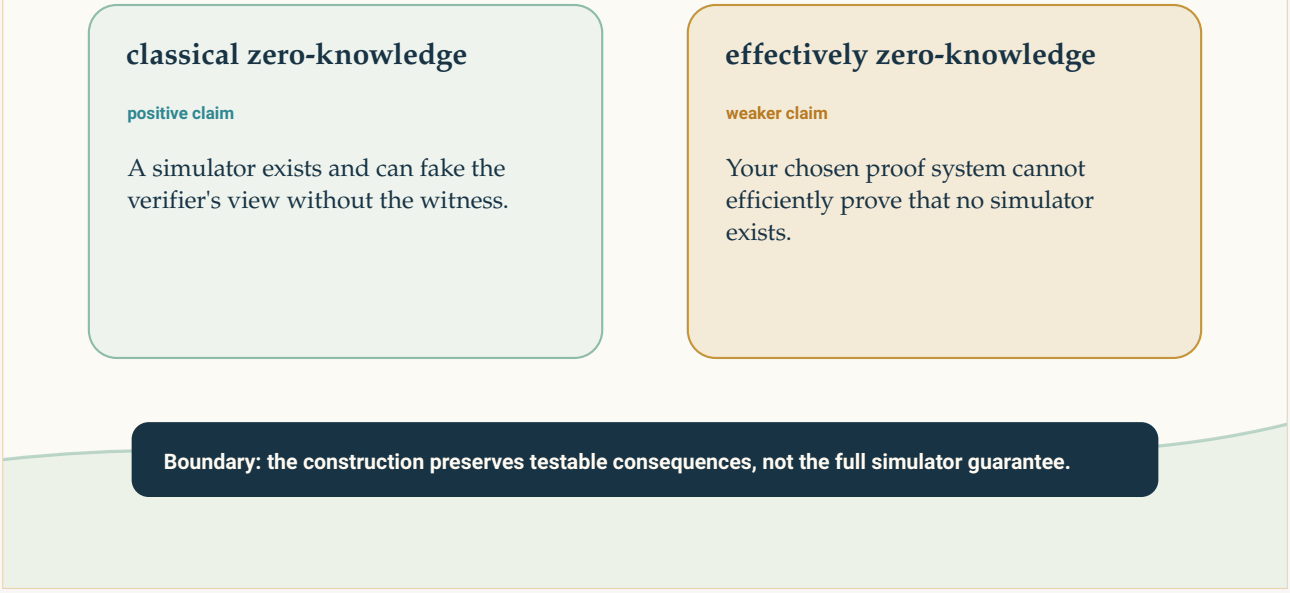


영지식은 세 개의 문에서 막힌다. 상호작용, 신뢰 설정, 불완전 건전성이다. Ilango 의 구성은 다른 문으로 빠져나간다. 선택한 규칙서가 시뮬레이터를 효율적으로 반박할 수 없게 한다.

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.



고전적 영지식은 시뮬레이터가 실제로 존재하는지를 묻는다. "실효적 영지식" 은 선택한 규칙서가 시뮬레이터가 존재할 수 없음을 효율적으로 증명할 수 있는지만 묻는다. 이 더 약한 질문 덕분에 메시지 하나, 설정 없음, 완전 건

전성을 유지할 수 있다.

오래된 기준: 시뮬레이터가 존재한다

영지식을 고전적으로 형식화할 때는 **시뮬레이터 (simulator)** 라는 가상의 조력자를 쓴다.

아이디어는 이렇다. Alice 의 비밀을 **모르는** Jane 을 상상하자. Jane 이 Alice 의 비밀 없이도 Bob 이 Alice 에게서 받았을 법한 증명과 똑같아 보이는 것을 혼자 만들어 낼 수 있다면, Alice 의 증명은 Bob 에게 새로운 것을 가르쳐 주지 않은 셈이다. Jane 은 이미 Alice 의 비밀 없이 그 경험을 가짜로 만들어 낼 수 있었다.

따라서 고전적 영지식은 실제 시뮬레이터를 요구한다. 비밀을 모른 채 그럴듯한 가짜 증명을 만들어 내는 효율적인 알고리즘이 존재해야 한다. 전문용어로 그 비밀을 **증인 (witness)** 이라고 부른다. Sudoku 에서는 완성된 해답 격자가 바로 증인이다.

이 정의는 강력하지만 바로 그 지점에서 오래된 불가능성이 작동한다. 직관은 이렇다. 진정한 비대화형 증명은 결국 하나의 문자열이다. Bob 이 그 문자열을 손에 넣으면 다른 사람에게 보여 줄 수 있다. 즉 Bob 은 그 명제를 다른 사람에게 다시 증명할 능력을 얻는다. 이미 "아무것도" 보다 많은 것을 얻은 것처럼 보인다. 고전 정리들은 이 직관을 위의 불가능성으로 날카롭게 만든다.

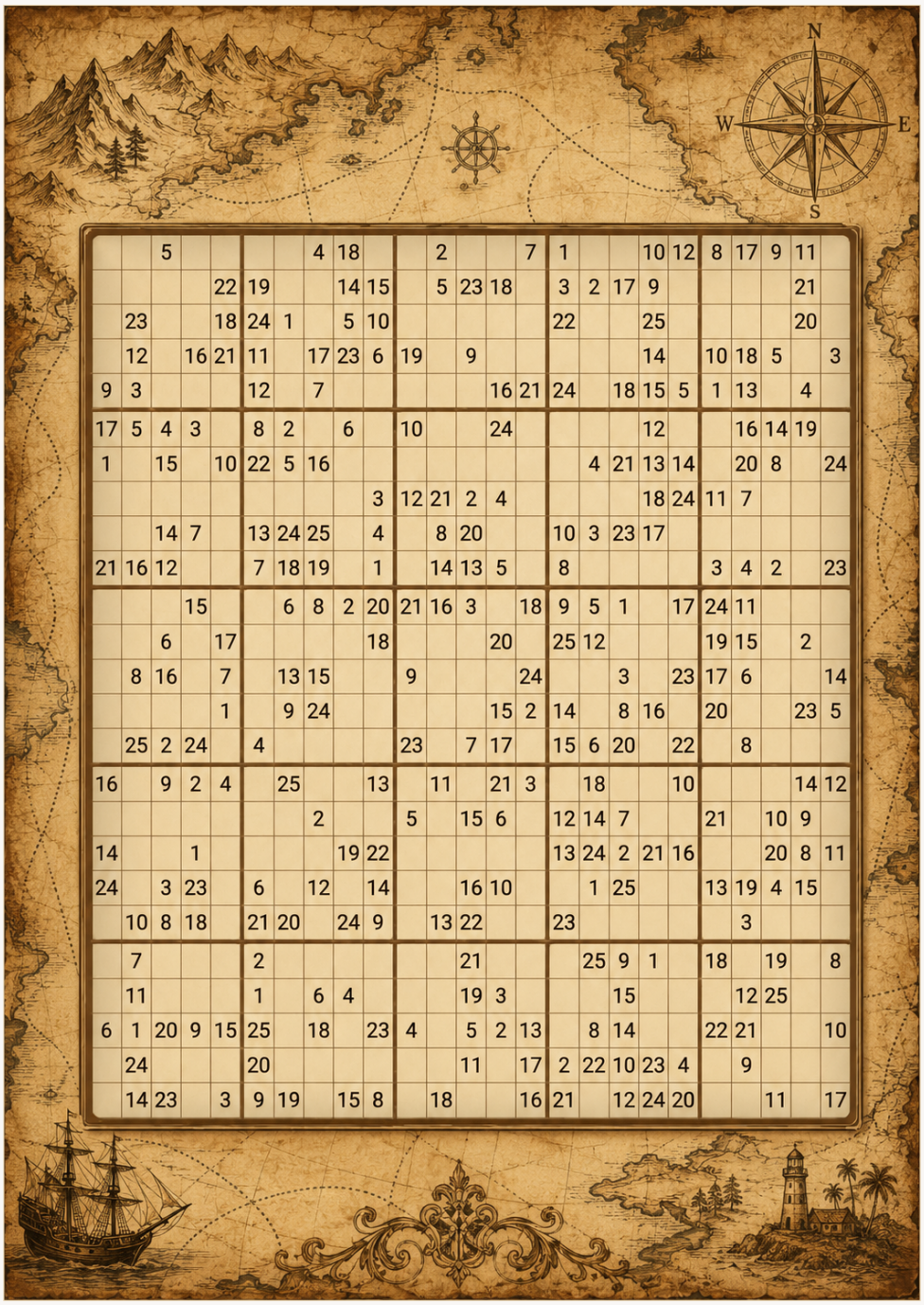
> **이 논문이 고집하는 세 속성**
>
> 논문 제목은 세 제약을 직접 이름 붙인다.
> **상호작용 없음:** Alice 는 증명 문자열 하나를 보낸다. 왕복 프로토콜이 없다.
> **설정 없음:** Alice 와 Bob 은 신뢰된 공통 참조 문자열이나 사전에 마련한 공개 난수에 의존하지 않는다. "비대화형 영지식" 이라 불리는 많은 시스템도 여전히 설정에 의존한다. 이 논문은 설정이 전혀 없는 경우를 뜻한다.
> **완전 건전성:** 거짓 명제에는 유효한 증명이 존재하지 않는다. "거의 항상 거부된다" 가 아니라 유효한 증명 자체가 없다.
> 이 세 속성은 보통의 서면 수학 증명이 갖는 속성과 정확히 같다. 그리고 앞서 설명했듯 고전적 영지식은 이 셋을 유지할 수 없다.

MegaSudoku 로 보는 차이

차이를 감각적으로 이해하기 위해 일부러 단순화한 비유를 써 보자.

진지한 부분에서는 보통의 9×9 Sudoku 를 쓰지 말자. 너무 작고 유한해서 컴퓨터가 그냥 풀어 버리거나 해답이 없음을 증명할 수 있다. 대신 **MegaSudoku(n)** 이라는 퍼즐 계열을 상상한다. 평소 규칙을 확장한다. 블록 크기를 n 으로 정하고, $N = n^2$ 라 두며, N 개의 기호를 쓰는 $N \times N$ 격자를 $n \times n$ 블록들로 나눈다. 보통 Sudoku 는 아주 작은 $n = 3$, $N = 9$ 인 경우에 불과하다. 9×9 격자, 3×3 블록, 기호 9 개다. 증명 복잡도 이야기는 n 이 커질 수 있고, 격자에 추가 장치를 넣어 Sudoku 옷을 입은 SAT 공식처럼 행동하게 할 때 비

로소 시작된다. SAT 공식은 결국 예/아니오 제약의 목록이다. 모든 제약을 만족하도록 변수에 참/거짓을 배정할 수 있는가?



25×25 Sudoku. 완성된 격자를 공개하지 않고도 규칙이 지켜지는지 확인할 수 있다. 숨겨진 해답, 즉 증인을 검증하는 증명을 시각적으로 대신 보여 주는 그림이다.

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

Sudoku 와 SAT: 같은 퍼즐의 두 가지 옷

Sudoku 가 “SAT 공식처럼 행동할 수 있다” 는 말은 비유가 아니다. 변환은 양방향으로 가능하며, 쉬

운 방향은 전부 써 내려갈 수 있다.
Sudoku 에서 SAT 로. SAT 는 참/거짓만 말하므로 (행, 열, 값) 세쌍마다 불리언 변수 하나를 둔다. *x(r,c,v)* 는 "*r* 행 *c* 열 칸에 값 *v* 가 들어 있다" 는 뜻이다. 4×4 Sudoku(2×2 블록, 값 1–4) 는 4·4·4 = 64 개 변수가 필요하고, 고전적인 9×9 에는 729 개가 필요하다. 그러면 모든 Sudoku 규칙을 절들의 묶음으로 바꿀 수 있다. (절 (clause) 은 변수 또는 그 부정의 OR 이고, 전체 공식은 모든 절의 AND 다.)
모든 칸에는 적어도 하나의 값이 들어간다 — 칸마다 절 하나:

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

모든 칸에는 많아야 하나의 값만 들어간다 — 값 쌍마다 "둘 다는 아님" 절:

$$\neg x(1,1,1) \lor \neg x(1,1,2) \quad \neg x(1,1,1) \lor \neg x(1,1,3) \quad \text{... 나머지 여섯 쌍도 같은 식이다.}$$

각 행에는 각 값이 하나씩 들어간다 — 1 행의 값 3 이라면 적어도 한 번:

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

그리고 많아야 한 번: $\neg x(1,1,3) \lor \neg x(1,2,3)$, 같은 방식으로 행의 모든 칸 쌍에 적용한다.
열과 블록 — 같은 절 묶음을 쓰되 칸의 집합만 달라진다. 왼쪽 위 블록에서 값 2 라면:

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

여기에 쌍별 "둘 다는 아님" 절을 더한다.
인쇄된 단서 — 가장 간단하다. 단서 하나는 변수 하나짜리 절이다. 왼쪽 위 칸에 인쇄된 3 은 다음 절이 된다.

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

이 모든 절의 AND 는 Sudoku 에 해답이 있을 때, 그리고 그때에만 만족 가능하다. 만족하는 변수 배정 자체가 해답이다. 참인 x(r,c,v) 를 읽어 격자를 채우면 된다. 9×9 라면 변수 729 개와 수천 개의 절 정도가 되며 현대 SAT 솔버는 밀리초 안에 처리한다. 단서 절 x(1,1,3) 을 주목하자. "이 칸은 정확히 3 이다" 라는 뜻이지 "이 칸들은 모두 다르다" 는 뜻이 아니다. 바로 이 비대칭 때문에 아래 프로토콜 설명에서 단서 칸에는 추가 장치가 필요하다.
SAT 에서 Sudoku 로. 논문에는 반대 방향, 더 어려운 변환이 필요하다. 임의의 SAT 공식을 받아 그 공식이 만족 가능할 때, 그리고 그때에만 해답이 있는 mega-Sudoku 를 만든다. Sudoku 의 원래 규칙은 "이 칸들은 모두 다르다" 정도밖에 말할 수 없으므로 임의의 논리 제약을 구축해야 한다. 바로 이것이 장치 (gadget) 다. 장치는 공식의 절마다 쓰는 작은 미리 제작된 칸 묶음이다. 지정된 칸들이 변수 역할을 하고 (그 칸에 든 기호가 참/거짓을 부호화한다), 내부 제약은 해당 절을 만족하는 배정만 합법적인 채움이 되도록 설계된다. NP-완전성 증명에서 표준적으로 쓰이는 제작 기술이며, 일반화 Sudoku 에 대해서는 Yato 와 Seta 가 2003 년에 수행했다.
두 방향을 합치면 N×N Sudoku 와 SAT 는 서로 다른 옷을 입은 같은 문제가 된다. 그래서 이 글과 논문이 격자와 기호를 이용해 NP 전체에 관한 이야기를 할 수 있다.

증인은 여전히 쉽게 상상할 수 있다. Alice 는 mega-Sudoku 의 완전하고 유효한 해답을 알고 있다. Bob 은 그런 해답이 존재함을 납득하고 싶지만 Alice 는 그것을 공개하고 싶지 않다. 전체 해답을 보내면 Bob 은

믿겠지만 비밀은 사라진다.

고전적인 영지식 버전에서는 Alice 와 Bob 이 상호작용한다. 오래된 정신적 모델 중 하나는 덮개가 있는 타일을 쓴다. Alice 는 완성된 격자를 숨기고, 매 라운드 기호 이름을 비밀리에 무작위로 바꾼 다음, Bob 에게 행·열·박스·장치 중 무작위로 고른 하나의 국소 제약만 보게 한다. 열린 칸들이 서로 다른 기호를 모두 보여 주면 Bob 의 신뢰가 커진다. 그런 다음 다시 전부 덮고 기호를 새로 바꾼다. (단, 퍼즐에 주어진 단서는 별도 장치가 필요하다. 기호를 바꾸면 단서도 함께 숨겨지기 때문이다. 아래 노트에서 고전 프로토콜이 이를 어떻게 해결하는지 설명한다. 지금은 이 장난감 그림만으로 충분하다.)

> **고전 프로토콜은 단서 칸을 실제로 어떻게 처리하나**
>
> 기호 이름 바꾸기에는 맹점이 있다. 행·열·박스 규칙은 모두 "이 칸들은 서로 다르다" 이고, 서로 다름은 기호를 어떤 순열로 바꿔도 유지된다. 그러나 단서는 "이 칸은 정확히 5 다" 라고 말한다. 기호를 바꾸고 나면 Bob 은 $\sigma(5)$, 즉 어떤 가려진 기호만 보며 순열 σ 를 모른다. 확인할 방법이 없다. 이를 그대로 두면 Alice 는 인쇄된 단서를 모두 무시하고 어떤 유효한 격자가 존재한다는 것만 증명할 수 있다. 그러면 이 퍼즐에 대해서는 아무것도 증명하지 못한다. 고전 문헌에는 두 가지 표준적인 수선법이 있다.
>
> **팔레트.** 숨긴 격자에 N 개 칸으로 된 행 하나를 추가한다. Alice 가 공개적으로 정해진 순서대로 기호 1…N 을 채운 팔레트다. 이후 다른 모든 것과 함께 기호를 바꾸므로 팔레트에는 $\sigma(1)$…σ(N) 이 들어간다. 이제 Bob 의 무작위 도전에는 선택지가 하나 더 생긴다. 행·열·박스·장치를 여는 것 외에 **팔레트와 단서 칸 하나**를 고를 수 있다. Alice 가 둘 다 공개하면 팔레트가 그 라운드의 순열을 보여 주고, Bob 은 단서 칸이 인쇄된 단서의 순열된 버전을 정확히 나타내는지 확인한다. 이 방식은 여전히 영지식이다. Bob 이 배우는 것은 매 라운드 새로 뽑혀 그 자체로 아무 가치가 없는 σ 와 이미 퍼즐에서 알고 있던 칸의 값뿐이다. 비밀 칸에 관한 정보는 새지 않고, 시뮬레이터도 무작위 σ 를 뽑아 같은 장면을 위조할 수 있다. 건전성도 있다. 속이는 Alice 는 라운드마다 고정된 확률로 들키며, 의심이 무시할 만큼 작아질 때까지 라운드를 반복한다.
>
> **단서를 제약으로 컴파일하기.** 더 구조적인 변형은 특별한 도전을 추가하지 않고 제거한다. 단서 값을 검증하는 대신 서로 다름 제약으로 강제한다. 단서 칸을 자신의 값이 있는 팔레트 칸을 제외한 모든 팔레트 칸과 연결해 "$\sigma(1)$ 과 다름, $\sigma(2)$ 와 다름, …, $\sigma(5)$ 를 제외한 모든 것과 다름" 이라고 만든다. 그러면 그 칸이 합법적으로 가질 수 있는 유일한 기호가 단서 값이 된다. 이제 모든 제약이 다시 "이 둘은 다르다" 꼴이 된다. 이름 바꾸기에 불변이고 행과 똑같이 검증할 수 있다. 고전 그래프 색칠 프로토콜에서 미리 색칠된 정점에 쓰는 것과 같은 기법이며, 위에서 말한 장치의 정신이기도 하다. MegaSudoku-as-SAT 그림에서는 단서도 다른 모든 제약처럼 부등식 장치로 컴파일된다.
>
> **물리적 프로토콜.** 실제 카드 기반 Sudoku 프로토콜 (Gradwohl, Naor, Pinkas, Rothblum, 2007) 은 기호 이름 바꾸기를 전혀 쓰지 않고 숨기기 시작하기 전에 단서를 확정한다. Alice 는 각 칸에 그 값이 적힌 같은 카드 세 장을 놓는다. 비밀 칸은 뒷면으로 놓지만 **단서 칸은 앞면으로** 놓아 Bob 이 카드가 뒤집히기 전에 단서가 지켜졌음을 직접 확인한다. 그런 다음 각 칸에서 한 장씩 행 묶음, 열 묶음, 박스 묶음에 넣고 각각 섞어서 공개한다. Bob 은 각 묶음에 N 개 기호가 모두 있는지 확인한다. 섞기는 위치 정보를 파괴하는데, 이것이 영지식 부분이다. 단서는 카드 배치 단계에서 이미 고정됐다. 어느 방식을 쓰든 교훈은 이 글 전체에서 반복되는 것과 같다. 영지식 프로토콜은 숨긴 뒤에도 어떤 사실이 남는지를 세심하게 장부 정리하는 일이다. 이름 바꾸기는 "모두 다름" 은 보존하고 "5 와 같음" 은 지워 버린다. 따라서 "5 와 같음" 은 다른 수단으로 다시 끼워 넣어야 한다.

이것이 논문의 프로토콜은 아니다. 고전적 영지식을 이해하기 위한 정신적 모델이다.

- Alice 와 Bob 은 서로 왕복한다.
- Bob 은 무작위 검사를 고른다.
- Alice 는 전체 해답이 아니라 국소적 일관성만 공개한다.
- 프라이버시 증명은 Bob 이 본 장면이 Alice 의 비밀 해답 없이도 생성될 수 있음을 보여 주는 방식으로 작동한다.

따라서 고전적 영지식은 하나의 긍정적 사실을 중심으로 세워진다.

> 시뮬레이터가 실제로 존재한다.

이제 편안한 조건들을 없애 보자. Alice 는 증명 문자열 하나를 보내고 떠난다. 신뢰 설정도 없고, 미리 준비한 공통 난수 문자열도 없으며, Bob 은 거짓 퍼즐을 절대로 받아들여서는 안 된다. 고전적 영지식이 살아남지 못하는 조건이다.

비결을 설명하기 전에 개념 하나를 더 도입해야 한다. **규칙서 (rulebook)**, 즉 논리학에서 말하는 형식적 증명 체계를 하나 고정하자. 고정된 공리 집합과, 쓰인 수학적 증명을 기계적으로 검사하는 규칙들이다. 표준 수학 공리계인 ZFC 가 대표적인 예다. 여기서부터 모든 주장은 미리 고른 규칙서에 상대적으로 서술된다. 선택은 유연하다. ZFC 를 포함해 어떤 규칙서를 고정해도 구성이 작동한다.

(용어를 하나 짚고 가자. 논문에서 “proof system” 은 Alice 가 보내는 메시지가 아니라, 수학적 증명을 검사하는 이 **형식적 증명 체계**를 뜻한다. Alice 와 Bob 쪽 알고리즘은 각각 **증명자 (prover)** 와 **검증자 (verifier)** 라고 부른다.)

Gödel 식 버전은 mega-Sudoku 이야기를 유지하되 증명을 바꾼다.

화면상 같은 크기의 두 번째 제약 시스템을 하나 고르고 **D** 라고 부르자. 이야기에서는 S 와 D 가 같은 형식의 두 MegaSudoku(n) 퍼즐이다. 실제 내부에서는 D 가 다른 크기의 어려운 논리 공식에서 출발했을 수 있다. 필요하면 무해한 더미 제약을 채워 같은 격자 크기에 맞춘다. D 는 실제로 **만족 불가능 (unsatisfiable)** 한 논리 공식에서 만들어진다. 모든 제약을 동시에 만족시키는 값 배정이 존재하지 않는다. 망가진 퍼즐에 합법적인 완성 격자가 없는 것과 같다. 장난감 예로는 “X 는 참이다” 와 “X 는 거짓이다” 를 동시에 요구하는 공식이 있다. 그러므로 D 에는 유효한 해답이 없다.

하지만 D 는 쉽게 망가졌다고 들통나는 퍼즐이어서는 안 된다. 방금 예는 실패한다. 어떤 규칙서든 “X 그리고 not-X” 는 한 줄로 반박한다. D 는 선택한 규칙서가 짧은 논증으로 그 거짓됨을 인증할 수 없는 방식으로 거짓이어야 한다. 규칙서가 D 를 짧은 증명으로 반박할 수 있다면 아래 이야기는 무너진다. Alice 의 비밀 없이도 증명을 만들 수 있었을 대체 경로를 형식적으로 배제할 수 있고, 그러면 프라이버시 보장도 함께 사라진다. 따라서 D 는 고정된 규칙서가 효율적으로 반박할 수 없는 계열에서 고른다. 즉 그 규칙서 안에는 D 에 해답이 없다는 짧은 증명이 없다.

Alice 의 한 메시지 증명은 다음의 또는 (or) 명제에 관한 것이다.

> 실제 mega-Sudoku S 에 해답이 있거나, 미끼 D 에 해답이 있다.

이것이 논리적 연결고리다. D 가 어떤 마법으로 S 를 참으로 만드는 것은 **아니다**. 증명은 “D 에 해답이 없으므로 S 에 해답이 있다” 라고 논증하지 않는다. **S 또는 D** 라는 논리합을 증명한다. 완전 건전성은 거짓 논리합에 유효한 증명이 있을 수 없다고 말한다. 실제로 D 는 거짓, 즉 해답이 없다. 따라서 논리합이 참일 수

있는 유일한 방법은 S 가 참인 것이다. 그러므로 증명이 받아들여지면 S 에는 반드시 해답이 있다. 미끼가 거짓 S 를 참으로 만들 수는 없다.

하지만 영지식 스타일의 부분에서는 D 에 해답이 있었다면 어떤 일이 일어날지를 묻는다. 그 미끼 해답은 대체 증인으로 작용한다. 누군가 Alice 의 실제 mega-Sudoku 해답을 모른 채 증명을 만들 수 있게 한다. 즉 시뮬레이터가 된다. 현실에서는 D 에 해답이 없으므로 이 시뮬레이터 경로는 닫혀 있다. 핵심은 **규칙서가 그 경로가 닫혀 있음을 효율적으로 증명할 수 없다는 것**이다.

따라서 D 는 두 가지 일을 한다. **건전성**을 위해서는 D 가 거짓이므로 "S 또는 D" 의 유효한 증명이 S 를 강제한다. **실효적 영지식**을 위해서는 D 가 반박하기 어려우므로 규칙서가 시뮬레이션을 가능하게 했을 미끼 경로를 빠르게 배제할 수 없다.

그러므로 보안 시험은 더 이상 다음이 아니다.

> 시뮬레이터가 실제로 존재한다고 증명할 수 있는가?

대신 이렇게 된다.

> 당신의 규칙서가 시뮬레이터가 불가능하다는 것을 효율적으로 증명할 수 있는가?

답이 아니라면 놀랍도록 강한 결과가 따라온다. (a) 시험을 실제로 실행해 관찰할 수 있고, (b) 시뮬레이터의 존재에서 따라온다는 것이 그 규칙서 안에서 증명되는 모든 보안 보장은 실제로 성립한다. 그중 하나라도 성공적으로 공격한다면 그 공격 자체가 바로 빠져 있던 짧은 반박을 만들어 내는 셈인데, 그런 짧은 반박은 존재하지 않는다. 이것이 실효적 영지식에서 "실효적" 인 부분이다.

교실식 대비로 정리하면 이렇다.

고전적 영지식: 시뮬레이터가 존재하기 때문에 증명이 안전하다.

Gödel 식 실효적 영지식: 관찰 가능한 보안 시험에 대해서는, 규칙서가 시뮬레이터가 불가능함을 효율적으로 증명할 수 없기 때문에 증명을 안전한 것으로 다룰 수 있다.

두 번째 주장은 더 약하다. 바로 그래서 논문은 고전 버전을 깨뜨렸던 세 속성, 즉 메시지 하나, 설정 없음, 완전 건전성을 유지할 수 있다.

새로운 기준: 시뮬레이터가 없음을 증명할 수 없다

Ilango 의 완화는 질문 자체를 바꾼다.

고전적 영지식은 묻는다.

> 시뮬레이터가 존재하는가?

실효적 영지식은 더 약한 것을 묻는다.

> 선택한 규칙서가 시뮬레이터가 존재하지 않는다는 것을 효율적으로 증명할 수 있는가?

기술적인 말장난처럼 들릴 수 있지만 이것이 핵심 아이디어다. 이 구성은 이상한 상태에 놓인다. 실제로는 시뮬레이터가 존재하지 않는다. 논문도 이를 명시한다. 하지만 미리 고정한 규칙서는 그것이 존재하지 않는다는 사실을 효율적으로 증명할 수 없다. 관심 있는 나쁜 결과가 모두 그런 반박을 필요로 한다면, 그 결과들에 대해서는 시스템이 여전히 영지식처럼 행동한다.

여기서 Gödel 이 등장한다. 장식도 아니고 "Gödel 이 암호를 안전하게 만든다" 는 뜻도 아니다. 연결은 증명 이론적이다. 규칙서가 **최적 (optimal)** 이라고 불리려면 정밀한 의미에서 가능한 최선이어야 한다. 관련 종류의 어떤 공식을 다른 규칙서가 짧은 증명으로 반박할 수 있다면 최적 규칙서도 많아야 다항식 정도 더 긴 증명으로 반박할 수 있어야 한다. Krajíček 과 Pudlák 은 1989 년에 **최적 증명 체계는 존재하지 않는다** 고 추측했다. 어떤 규칙서를 고르든 다른 규칙서가 어떤 참인 명제 계열을 훨씬 더 짧게 증명할 수 있다는 것이다. 이는 증명 복잡도의 핵심 미해결 추측 가운데 하나이며, Gödel 불완전성 정리의 유한하고 복잡도 이론적인 사촌이다. 미리 고정한 규칙서에는 짧은 증명이 없는 참인 명제가 존재한다. 원리적으로 증명 불가능해서가 아니라, 어떤 고정된 규칙서도 일부' 짧게 표현되는 진실' 을 짧게 증명하지 못하기 때문이다.

논문은 이 추측의 약간 더 강한 "무한히 자주 (infinitely often)" 형태를 가정한다. 추측을 암호학적으로 사용할 때 표준적인 형태다. Krajíček–Pudlák 정리에 따라 보상은 구체적이다. 어떤 규칙서를 고르더라도 실제로 만족 불가능하지만 그 규칙서가 짧은 증명으로 반박할 수 없는 공식의 수열이 존재한다. 더 중요하게는 효율적인 알고리즘이 그 수열을 **생성할 수 있다**. 이 마지막 속성인 균일성 (uniformity) 이 순수한 존재 주장과 Alice 가 실제로 실행할 수 있는 알고리즘의 차이를 만든다. 즉 미끼 D 는 단순한 존재 증명이 아니라, 실제 알고리즘으로 만들어 낼 수 있다.

암호학적 움직임은 이 증명 능력의 부족을 보안에 이용하는 것이다.

구성은 무엇을 하는가

논문의 구성을 형태만 남겨 보자.

규칙서 하나, 예를 들어 ZFC 를 고정한다. 증명 복잡도 가정 아래에는 실제로 만족 불가능하지만 그 규칙서가 만족 불가능함을 짧게 증명할 수 없는 공식의 효율적으로 생성 가능한 수열이 존재한다.

이제 다음 꼴의 한 메시지 증명을 만든다.

> 실제 명제가 만족 가능하거나, 이 특별한 어려운 공식이 만족 가능하다.

특별한 어려운 공식은 만족 가능하지 않다. 따라서 바탕의 증명 장치가 완전 건전하다면 메시지가 받아들여졌다는 사실은 여전히 실제 명제가 참임을 뜻한다. 이것이 완전 건전성을 준다.

하지만 영지식과 비슷한 보안에서는 특별한 어려운 공식이 만족 가능하다고 상상해 보자. 그러면 그 공식의 증인을 이용해 실제 증인을 모른 채 증명을 시뮬레이션할 수 있다. 현실에서는 그 공식이 만족 가능하지 않다. 그러나 규칙서는 이를 효율적으로 증명할 수 없다. 따라서 시뮬레이터가 불가능함도 효율적으로 증명할 수 없다.

이것이 경첩이다. 시스템은 고전적인 시뮬레이터를 만들어 비밀을 숨기는 것이 아니다. 광범위한 관찰 가

능한 보안 시험에 대해 **시뮬레이터의 부재를 규칙서가 인증하지 못하는 뒤편에 비밀을 숨긴다.**

논문이 주장하는 것

주요 정리는 여러 층으로 나뉜다. 핵심 결과는 다음과 같다.

표준적인 암호학적 가정인 **비대화형 증인 비구별 증명 (non-interactive witness indistinguishable proofs)** 의 존재와, **(무한히 자주) 최적 증명 체계가 존재하지 않는다**는 증명 복잡도 추측 아래에서, 논문은 어떤 규칙서를 선택하더라도 그 규칙서에 상대적으로 실효적 영지식인 NP/SAT 용 한 메시지 증명자·검증자 쌍을 구성한다. **완전 건전성**을 가지며 설정도 필요 없다. (NP/SAT 는 퍼즐 같은 문제들의 표준적인 "가장 어려운 공통분모" 이고, mega-Sudoku 는 그중 하나의 모습이다.)

반증 가능한 보안 속성을 보존한다는 더 넓은 주장에는 표준적인 탈무작위화 믿음인 **P = BPP** 를 하나 더 가정한다. 대략 말하면 무작위성이 알고리즘에 본질적인 추가 능력을 주지 않는다는 믿음이다.

정리의 언어를 풀어 쓰면 다음과 같다.

- 증명은 메시지 하나다.
- 신뢰 설정이 없다.
- 거짓 명제는 증명할 수 없다.
- 증명자는 고전적 영지식이 아니다. 시뮬레이터가 없다.
- 하지만 고전적 영지식에서 따라오는 반증 가능하고 게임 기반인 보안 결과를 각각 이 조건에서 달성할 수 있다.

"반증 가능 (falsifiable)" 하다는 점이 중요하다. 보안 실패를 공격자를 어떤 게임에 넣어 실제로 실행함으로써 시험할 수 있다는 뜻이다. 많은 암호학적 보안 정의가 이런 형태다. 공격자가 두 암호문을 구별할 수 있는가? 함수를 역산할 수 있는가? 증인을 복구할 수 있는가? 정해진 실험에서 이길 수 있는가? 정리는 반증 가능한 속성마다 하나씩 그 속성을 만족하는 증명자를 준다. 하나의 증명자가 모든 반증 가능한 속성을 동시에 만족하는 것은 아마 불가능하다. 오래된 재사용 공격, 즉 "Bob 이 증명을 다른 사람에게 보여 줄 수 있다" 는 것 자체가 반증 가능한 속성인데 실제로 여기서는 실패한다. 논문은 하나의 증명자가 암호학 실무에서 실제로 등장하는 모든 자연스러운 반증 가능 속성을 포괄할 수 있을 것이라고 제안한다. 하지만 이 부분은 "자연스럽다" 라는 비형식적 개념에 더해 명시적인 추측에 기대는 조건부 정리다. 보장은 관찰 가능한 실패를 겨냥하지, 모든 철학적 의미나 시뮬레이션 기반 의미의 비밀성을 보장하지 않는다.

구체적인 따름정리 하나는 이름을 붙일 만하다. 이 구성은 균일한 증명자를 갖는 최초의 비대화형 **증인 은닉 (witness hiding)** 증명을 준다. "퍼즐의 증명을 봐도 해답을 찾는 데 도움이 되지 않는다" 를 상호작용도 설정도 없이 달성하는 것이다. 소박해 보이지만 수십 년 동안 구성이 어려웠던 객체다.

이 논문이 말하지 않는 것

이 부분이 글을 정직하게 만든다.

오래된 불가능성 정리가 틀렸다는 뜻이 **아니다**. 구성은 정의를 바꿔 그것을 피한다.

상호작용도, 설정도 없이 완전 건전성을 가진 보통의 고전적 영지식을 주는 것도 **아니다**. 논문은 구성된 증명자에 시뮬레이터가 없다고 명시한다.

증명을 재사용할 수 없다는 뜻도 **아니다**. 한 메시지 증명은 여전히 다른 사람에게 보여 줄 수 있다. 논문은 부인 가능성 (deniability) 같은 속성을 보존하지 않는다. (신뢰 설정이 있는 비대화형 영지식도 같은 한계가 있다.)

배포할 준비가 된 실용 프로토콜이라는 뜻도 **아니다**. 이는 복잡도 이론과 암호학의 기초 연구다. 결과는 증명 복잡도와 암호학의 중요한 가정에 의존하며, 구성은 원리적으로 무엇이 가능한지를 묻는다.

"Gödel" 이 마법의 보안 원시요소가 된다는 뜻도 **아니다**. Gödel 과의 연결은 증명 체계, 최적 증명 체계, 불완전성의 유한한 유사물을 통해 이뤄진다. 유용한 직관은 "불완전성이 비밀번호를 지켜 준다" 가 아니다. **규칙서가 시뮬레이터가 불가능하다는 것을 효율적으로 증명할 수 없다면, 그런 증명을 필요로 하는 공격은 보안 정의 수준에서 막을 수 있다**는 것이다.

그래도 왜 흥미로운가

암호학은 자주 어려움을 안전으로 바꾼다. 인수분해가 어려우므로 RSA 류 가정을 활용한다. 격자 문제가 어려우므로 격자 암호를 활용한다. 여기서 어려움은 더 낯설다. "비밀을 계산하기 어렵다" 가 아니라 **"어떤 증명 객체가 존재할 수 없다는 사실을 증명하기 어렵다"** 다.

그래서 이 논문은 특이하게 느껴진다. 공리와 규칙서를 거의 암호학적 자원처럼 다룬다. 평소의 불가능성은 건전성과 시뮬레이션 사이에 긴장이 있다고 말한다. Ilango 의 움직임은 그 긴장을 증명 이론의 장막 뒤에 놓는다. 시뮬레이터는 없지만 형식 체계가 그 부재를 효율적으로 드러내지 못한다.

독자에게 놀라운 점은 이것이 오늘날의 영지식 시스템을 대체하리라는 데 있지 않다. 적어도 직접적으로는 아마 그렇지 않을 것이다. 놀라운 점은 수리논리의 한계를 건설적으로 쓸 수 있다는 것이다. 벽으로만 쓰는 것이 아니라 일종의 가림막으로 쓴다.

근거는 얼마나 탄탄한가?

이는 정리 논문이므로 "증거" 는 생물학이나 천문학 논문과 다른 뜻을 가진다. 실험이 재현됐는지가 질문이 아니다. 정의, 가정, 증명 사슬이 주장을 지지하는지가 질문이다.

증명은 형식적이고 논문은 가정을 명시한다. 가정도 가볍지 않다. 비대화형 증인 비구별 증명은 암호학에서 표준적으로 연구돼 온 객체이며 여러 확립된 가정 묶음에서 따라온다. 최적 증명 체계 부재 추측은 증명 복잡도의 핵심 추측이다. P = BPP 는 표준적인 탈무작위화 믿음이며 더 넓은 반증 가능 속성 정리에만 사용된다.

논문은 이 가정들이 임의로 세운 비계가 아니라 필요한 대가라는 논거도 제시한다. 역방향 결과를 증명해 이런 구성이 존재한다면 비대화형 증인 비구별 증명도 반드시 존재해야 하고, 표준적인 일방향 함수가 있다고 할 때 최적 증명 체계가 존재해서는 안 된다는 것을 보인다. 가정에는 흥미로운 양면성이 있다. 그중 하나라도 반박된다면 증명 복잡도, 암호학, 복잡도 이론에서 그 자체로 획기적인 발견이 된다.

하지만 결과가 조건부인 만큼 신뢰도 조건부다. 이 가정들이 실패하면 정리의 해석도 달라진다. 그리고 가정들이 참이어도 보장은 완전한 고전적 영지식이 아니라 논문이 제안한 완화된 증명 이론적 버전이다.

따라서 적절한 신뢰 수준은 이렇다. **논문이 일관된 조건부 가능성 결과를 세웠다는 데에는 높은 신뢰, 가정들이 우리가 실제로 사는 암호학 세계를 묘사한다는 데에는 중간 정도의 신뢰, 즉각적인 실용적 결과에는 낮은 신뢰**가 적절하다.

왜 중요한가

논문은 닫혀 있다고 여겨졌던 길 하나를 연다.

고전 이론은 말한다. 완전한 영지식은 설정 없이 메시지 하나로 만들 수 없고, 완전 건전성을 가질 수도 없다. Ilango 의 논문은 이렇게 말한다. 보안 게임에서 시험할 수 있는 영지식의 결과들을 요구하고, 규칙서가 무엇을 효율적으로 반박할 수 있는지에 보안 정의를 의존시키면 유용한 행동의 상당 부분을 회복할 수 있다. 메시지 하나, 설정 없음, 완전 건전성을 유지하면서 말이다.

이는 작은 정의 수정이 아니다. 암호학적 보장을 생각하는 다른 방식이다. 무엇이 존재하는지만 묻지 말고 규칙서가 무엇을 배제할 수 있는지 묻는다. 증명 불가능성을 철학적 골칫거리로만 다루지 말고 구조로 이용한다.

실용 세계가 내일 바로 바뀌지는 않을 것이다. 하지만 개념 지도는 바뀐다. 이제 “누구도 비밀이 새었다는 것을 효율적으로 증명할 수 없다” 가 “비밀이 새지 않았다” 에서 원했던 많은 게임 기반 보호를 회복하기에 충분할 수 있다는 형식적인 의미가 생겼다.

그래서 제목에 Gödel 이 들어간다.

핵심 요약

영지식 증명은 증명자가 증인을 공개하지 않고도 명제가 참임을 검증자에게 납득시킬 수 있게 한다. 고전적 불가능성 결과는 영지식을 설정 없는 한 메시지로 압축할 수 없고 완전 건전성도 가질 수 없다고 말한다. Rahul Ilango 의 논문은 이 불가능성을 반박하지 않는다. 대신 더 약한 개념인 **실효적 영지식**을 정의한다. 시뮬레이터가 실제로 존재해야 한다고 요구하는 대신, 선택한 증명 체계—ZFC 같은 형식적 규칙서—가 시뮬레이터가 존재하지 않는다는 사실을 효율적으로 증명할 수 없어야 한다고 요구한다. 암호학의 주요 가정 (비대화형 증인 비구별 증명) 과 증명 복잡도의 주요 가정 (최적 증명 체계가 존재하지 않음) 아래에서 논문은 설정 없이 완전 건전성을 가지는 NP/SAT 용 한 메시지 증명자를 구성하며, 영지식에서 따라오는 반증 가능하고 게임 기반인 보안 결과를 속성별로 달성한다. 모든 “자연스러운” 속성을 하나의 증명자가 동시에 포괄한다는 더 넓은 주장은 추가로 일부 추측적이며, 문자 그대로 모든 반증 가능 속성을 포괄하는 것은 증명이 재사용 가능하기 때문에 아마 불가능하다. 이 결과는 이론적이고 조건부이며 배포 가능한 원시요소가 아니다. 하지만 **증명 이론적 비증명성을 암호학적 자원으로 쓰는 새로운 방법**을 보여 준다.

과장 없이 보면

논문이 보여 주는 것: 명시된 가정 아래에서 어떤 선택된 증명 체계에 상대적으로 실효적 영지식이며, 고전적 영지식의 반증 가능하고 게임 기반인 각 보안 결과를 달성하는 NP/SAT 용 한 메시지·무설정·완전 건전 증명자를 만들 수 있다.

그럴듯하지만 무조건적으로 입증되지 않은 것: 필요한 증명 복잡도 및 암호학 가정이 실제로 참이라는 것. 모두 진지하게 연구되는 가정이고 논문은 본질적으로 충분조건일 뿐 아니라 필요조건이기도 함을 보이지만, 여전히 가정이다.

보여 주지 않는 것: 상호작용도 설정도 없이 완전 건전한 고전적 영지식, 실제 배포 준비가 끝난 시스템, 증명의 부인 가능성 또는 비재사용성, 혹은 Gödel 불완전성 정리 자체가 암호학을 안전하게 만든다는 주장을 보여 주지 않는다.

주요 한계: 보장은 영지식의 완화된 버전이다. 가장 넓은 형태는 여러 가정에 의존한다. 하나의 보편적 증명자에 관한 주장은 일부 추측적인 상태로 남아 있으며, 결과의 주된 의미는 기초 이론에 있다.

일반 독자는 어느 정도 신뢰해야 하나? 정의를 받아들인다면 중요한 조건부 이론 결과라는 데 높은 신뢰를 둘 수 있다. 가정이 현실을 포착한다는 데에는 중간 정도, 즉각적인 실용 배포에는 낮은 신뢰가 적절하다. 안전한 결론은 이것이다. **이 논문은 영지식 불가능성을 깨뜨리지 않는다. 많은 보안 게임에서 중요한 부분을 증명 이론적으로 우회하는 새로운 길을 찾는다.**

출처

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