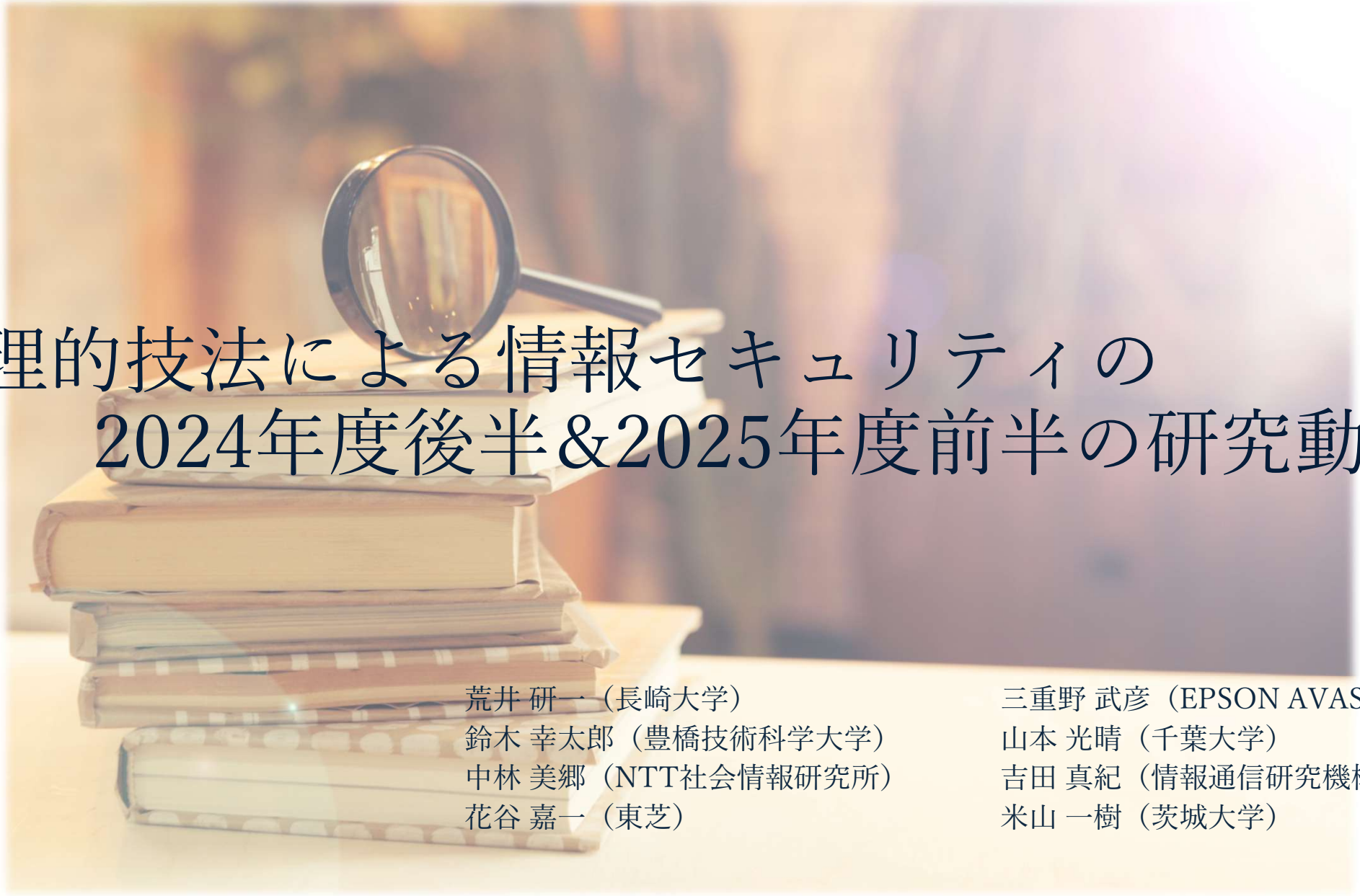


# 数理的技法による情報セキュリティの2024年度後半&2025年度前半の研究動向

2025.9.4


荒井 研一（長崎大学）
鈴木 幸太郎（豊橋技術科学大学）
中林 美郷（NTT社会情報研究所）
花谷 嘉一（東芝）
三重野 武彦（EPSON AVASYS）
山本 光晴（千葉大学）
吉田 真紀（情報通信研究機構）
米山 一樹（茨城大学）

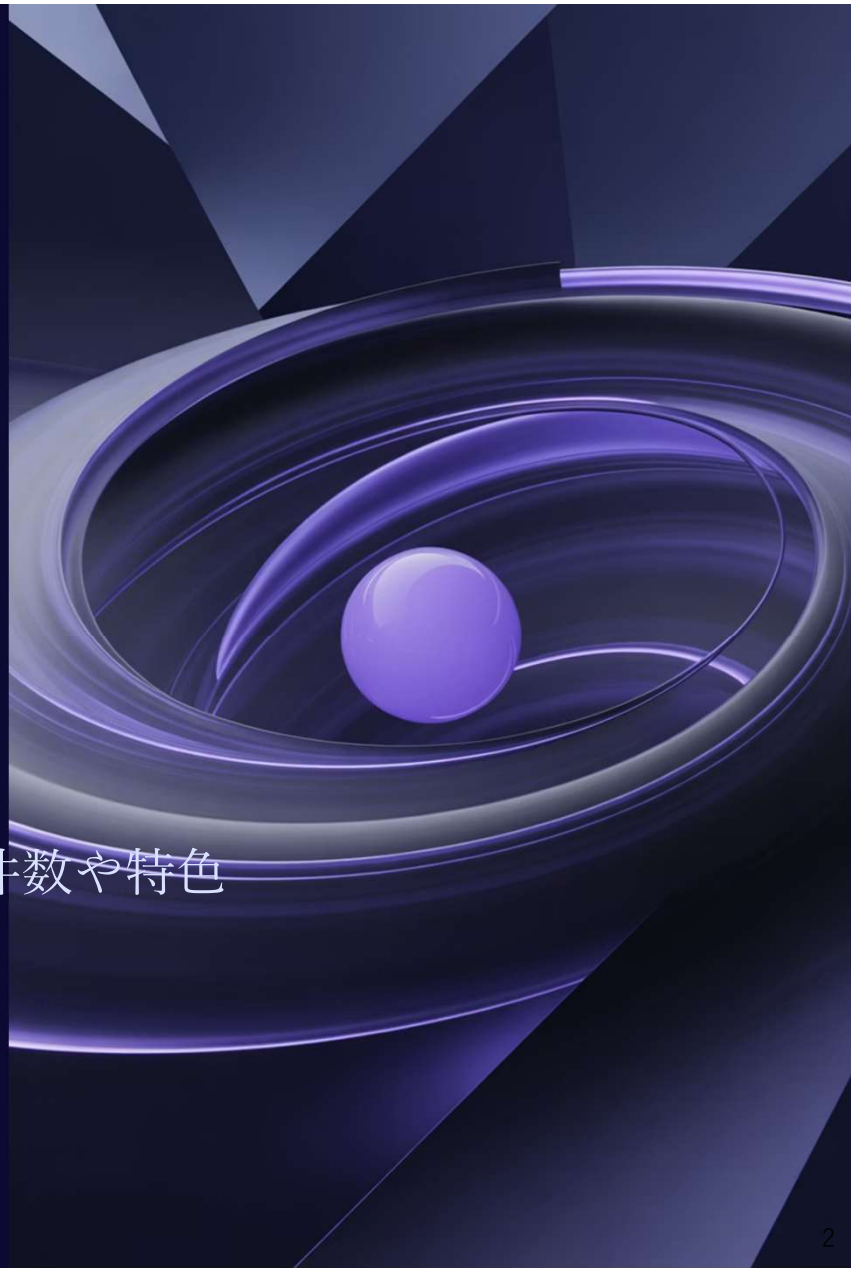
# 目的:
# 数理的技法による
# 情報セキュリティ分野の研究
# および実用化の促進

## 最近の研究動向を紹介

・各セキュリティトップ会議における関連論文の発表件数や特色

・全体を通したトレンド(分析)

・関連論文の紹介(受賞した一部のみ)



# 目次

セキュリティの国際トップ会議（CCS, NDSS, S&P, CSF, CAV, USENIX）
暗号・プログラム検証・セキュリティ分野の研究動向を分析・紹介

# 1.
# 調査概要
## 収集対象論文と会議の特徴

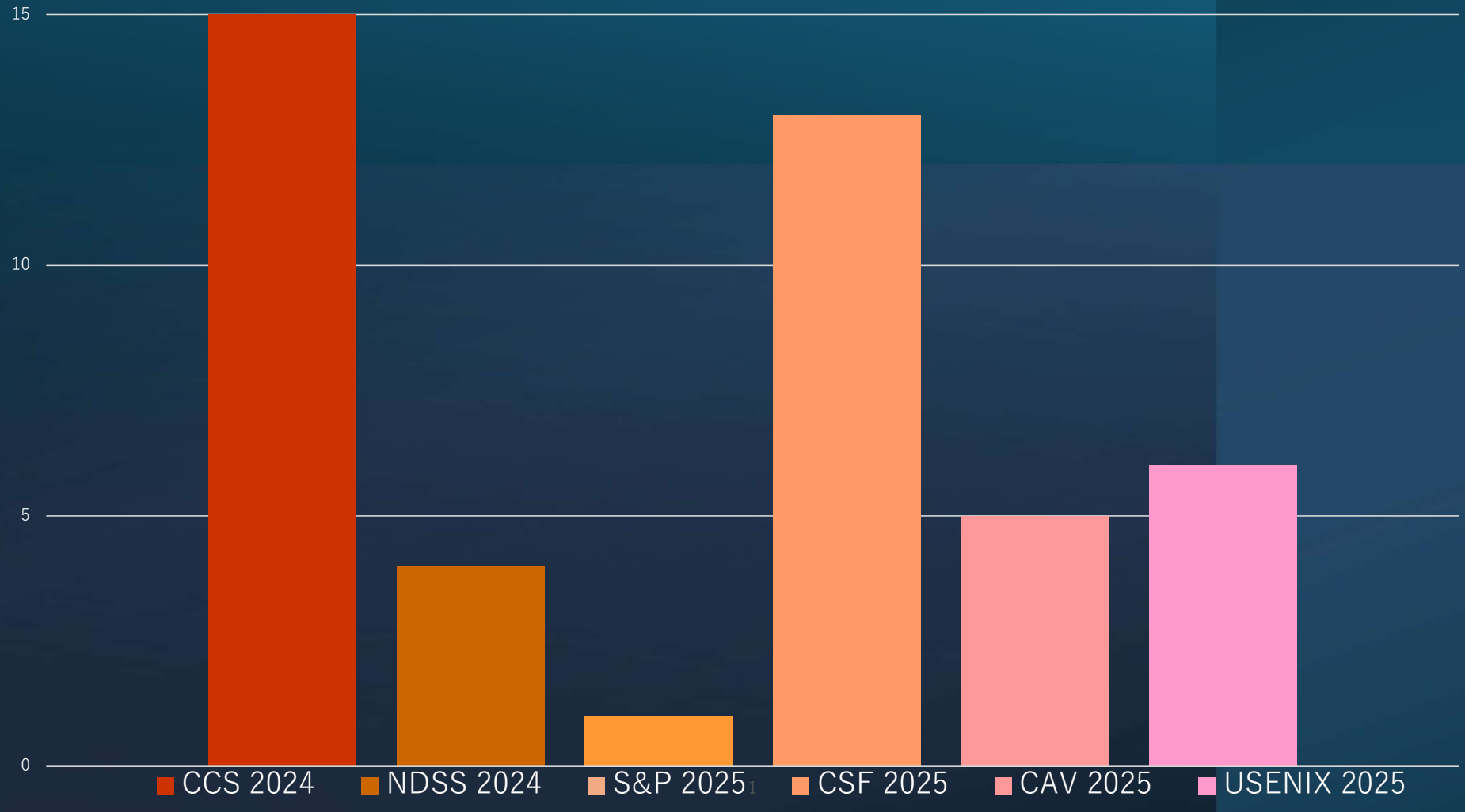
# 収集対象論文

| 44 | 6 | 2024-2025 |
| --- | --- | --- |
| 論文総数 | 会議数 | 発表年 |
| | CCS, NDSS, S&P, CSF, CAV, USENIX | 2024年後半～2025年前半 |

収集基準
- Title または, Abstractに "Formal"の語句が入る
- 各会議で開催する"Formal Analysis Session"などで取り上げている
- 明らかに内容が遠いものは除く

# 会議別論文数

CCS2024とCSF2025が最も多くの収集論文を発表
形式手法によるセキュリティ検証の中心的な会議となっている



# 6つのセキュリティ会議の特徴

| 会議名 | 分野 | 理論 or 実装 | 開催時期 | 特徴・傾向 |
|---|---|---|---|---|
| ACM CCS | セキュリティ全般＋AIなど広範 | 理論・実装両方 | 10月 | 国際性が高く幅広いテーマ,暗号理論やMLセキュリティを扱う |
| NDSS | ネットワーク／IoTセキュリティ | 実装寄り | 2月 | 分散システム、IoTなどに強い実装重視 |
| IEEE S&P | セキュリティ全般 | 理論・実装両方 | 5月 | 歴史的に最も古い,形式手法とプライバシーに強い,理論～実装まで広く扱う |
| IEEE CSF | セキュリティ理論・形式手法 | 理論中心 | 7月 | 形式モデルやプロトコル検証,ツールを用いた検証研究が多い |
| CAV | 形式検証<br>(セキュリティ以外含む) | 理論＋ツール重視 | 7月 | 検証全般, 安全性・正当性の形式的保証やツール開発を扱う |
| USENIX Security | 実システム・攻撃・応用 | 実装寄り | 8月 | 実用・実装寄り,攻撃手法・OSセキュリティ・UXを扱う |

# 2.
# 会議別動向分析
## 各会議における研究傾向

# ACM CCS2024の研究動向

ACM CCS2024では、暗号プロトコル検証から低レベルシステム検証まで幅広い研究が発表された:
特に以下のような傾向

## プロトコル検証の高度化

Tamarin proverなどの既存ツールを拡張し、より複雑な暗号プロトコルの検証を可能にする内容（KEMベースのプロトコル、ライトニングネットワークなど）

## プライバシー分析の形式化

Googleのトピック APIなど、プライバシー保護技術の理論的分析と定量的評価

## サイドチャネル耐性の検証

マイクロアーキテクチャレベルのサイドチャネル漏洩の検出と形式的検証

## 低レベルコードの安全性

コンパイラの安全性保証やバイナリレベルの検証に関する内容

# NDSS 2025の研究動向

NDSS2025では、スマートコントラクトと耐量子プロトコルに関する研究が発表された:
特に以下のような傾向

## スマートコントラクト検証

PropertyGPT: 大規模言語モデルを活用した新しい検証アプローチの提案
従来の形式検証手法とAIを組み合わせることで、検証の効率化と自動化を目指している

## 耐量子プロトコル

PQ Connect: 耐量子暗号プロトコルの形式的検証が行われている
特にTEEと組み合わせた多層権限モデルの安全性検証に焦点が当てられている

# S&P2025の研究動向

S&P2025では、モバイル通信インフラのセキュリティに関する研究が発表された:
特に以下のような傾向

## 5Gコアネットワークの形式分析

5Gコアネットワークのアクセス制御メカニズムを形式的にモデル化し、61のセキュリティ特性を検証する包括的な研究が発表された。間接通信モードと5Gローミングを考慮した初の形式モデルを構築し、10種類のアクセス制御攻撃（5つは新規発見）を発見

## 構成的検証技術の応用

Assume-Guarantee型の推論アプローチを活用した構成的検証技術が採用され、スプリットアサーションの原理を活用してシステムモデルとセキュリティ特性を独立したコンポーネントに分解することで、スケーラブルで効果的なモデルチェックを実現

# CSF2025の研究動向

CSF2025では、暗号プロトコルの理論的安全性から差分プライバシーまで幅広い研究が発表された:
特に以下のような傾向

## 暗号理論の形式化

等式理論の有限変異性（FVP）の検証や、ゼロ知識証明の形式的分析など、暗号理論の形式的基礎に関する研究

## 差分プライバシーの型システム

インタラクティブな差分プライバシーメカニズムのための型システムや、Gradual Sensitivity Typingなど、より柔軟な差分プライバシー保証の研究

## ハードウェアセキュリティ

投機的実行によるサイドチャネル攻撃からの保護や、エンクレーブシステムのタイミング隔離に関する形式的検証

## 証明技術の改良

Rocqを用いた証明の自動化や、名義状態分離証明（Nominal State-Separating Proofs）など、形式証明技術そのものの改良

# CAV2025の研究動向

CAV2025では、低レベル実装の検証と代数的手法に関する研究が発表された:
特に以下のような傾向

## 低レベル実装の検証

パイプライン化されたプロセッサのセキュリティ検証や、アセンブリプログラムのRelational Hoare Logicによる検証など、ハードウェアに近いレベルでの形式的検証手法の研究が進められている

## 代数的手法の応用

ゼロ知識証明回路の検証やmultimodular systemsの検証など、代数的手法を用いた暗号実装の検証に関する研究が行われている
これらの研究では、SMTソルバと代数計算を組み合わせた新しいアプローチが提案されている

# USENIX2025の研究動向

USENIX2025では、実用的な暗号プロトコル実装の検証に関する研究が発表された:
特に以下のような傾向

## PQ-WireGuardの検証

耐量子暗号プロトコルPQ-WireGuardの形式的安全性検証と、脆弱性を修正したHybrid-WireGuardの提案

## iMessage PQ3の分析

AppleのiMessage PQ3プロトコルのTamarin proverによる安全性検証と設計上のトレードオフの指摘

## 検証済み実装の自動生成

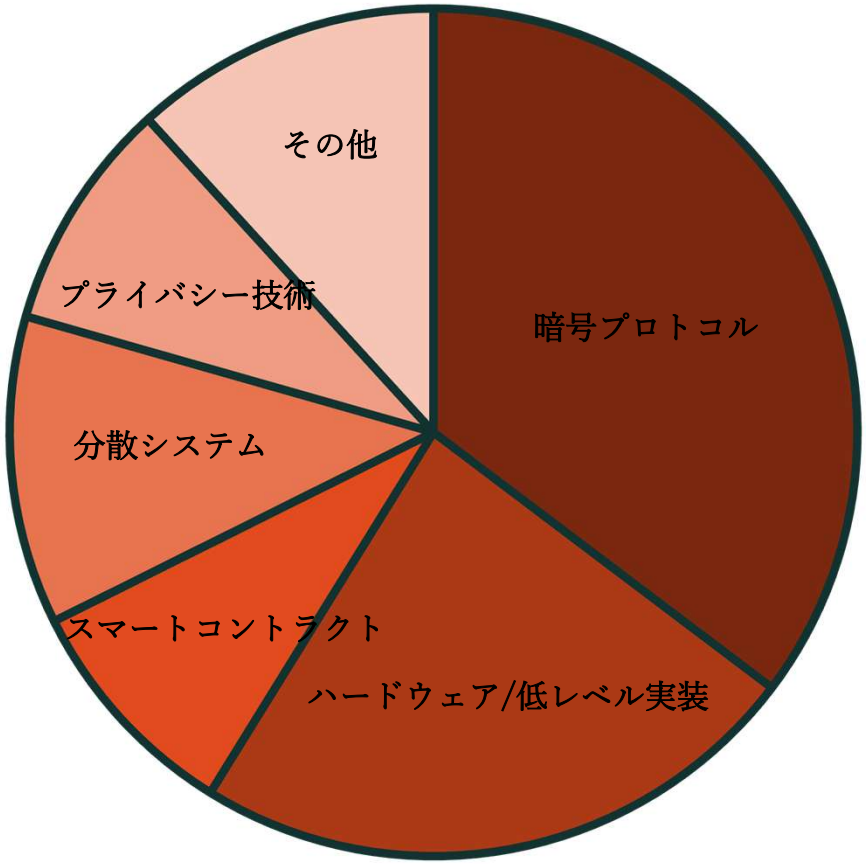
OwlCによる計算論的に安全な暗号プロトコルの検証済み実装の自動生成

## 実用的な証明書検証

X.509証明書のためのエンドツーエンド形式検証済み検証ツールの開発

# 3.
# 研究対象の傾向
## どのような技術・システムが対象か

# 研究対象の傾向




・暗号プロトコルの検証が最も多く、次いでハードウェア/低レベル実装の検証が多くなっている
・スマートコントラクト、プライバシー技術の検証も一定数発表

# 研究対象の詳細 1/3：暗号プロトコル

## 耐量子暗号プロトコル

PQ-WireGuard, iMessage PQ3, PQConnectなど、量子コンピュータによる攻撃に耐性のある暗号プロトコルの検証

## KEM（鍵カプセル化機構）

KEMベースのプロトコルの安全性分析と「再カプセル化攻撃」などの新たな攻撃の発見

## 認証プロトコル

OpenID Connect、OPC UAなどの標準的な認証プロトコルの形式的検証

## ゼロ知識証明

LPNベースのゼロ知識証明の最適化と形式的検証

暗号プロトコルの分野では、耐量子暗号への移行に伴う安全性検証が増加、
特に既存プロトコルのハイブリッド構成（古典暗号と耐量子暗号の組み合わせ）の検証が注目されている

# 研究対象の詳細 2/3：ハードウェア/低レベル実装

## マイクロアーキテクチャのセキュリティ

- $\mu$CFI: マイクロアーキテクチャレベルの制御フロー整合性検証
- サイドチャネルに関する非干渉性のテスト
- FSLHによる投機的実行攻撃からの保護

## コンパイラとバイナリ検証

- SECOMP: 安全なコンパイルの形式的保証
- VeriBin: バイナリレベルでのパッチ検証
- x86-64バイナリのsymbolized NASMアセンブリへの変換

ハードウェアや低レベル実装では、サイドチャネル脆弱性への対策が継続的に研究されている
また、コンパイルやバイナリレベルでの安全性保証も重要なトピックとなっている

# 研究対象の詳細 3/3： スマートコントラクト,分散システム,プライバシー技術,その他


## ブロックチェーンとスマートコントラクト

ライトニングネットワークの形式的分析、ブロックチェーンのDoS脆弱性の自動発見、LLMを用いたスマートコントラクトの形式検証など

## プライバシー技術

GoogleのトピックAPIの理論的分析、差分プライバシーのための型システム、セッション型による差分プライバシーの検証など

## 通信インフラ(分散システム含)

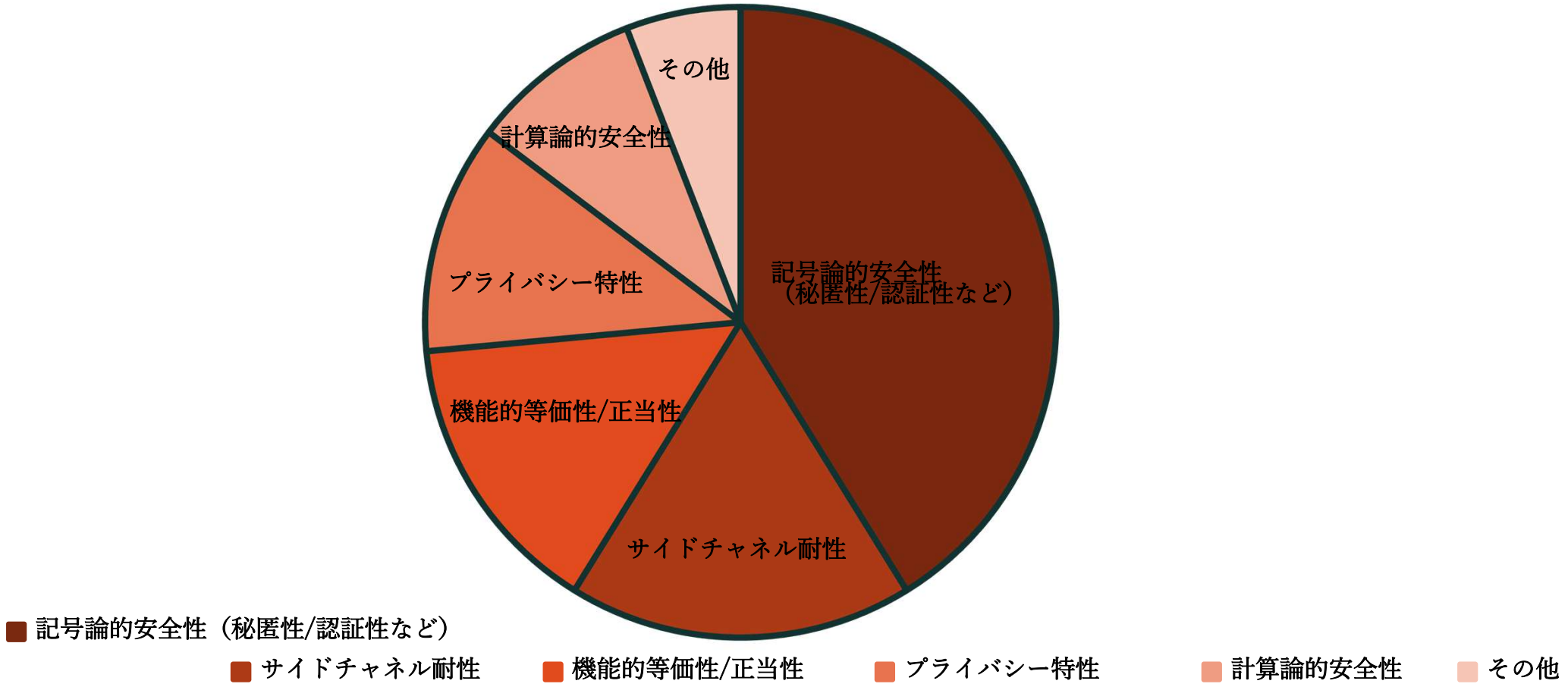
5Gコアネットワークのアクセス制御メカニズムの形式的分析、OPC UAプロトコルの安全性検証など

## 信頼できるシステム

エンクレーブシステムのタイミング隔離の形式化、TEEのための多層権限モデル、データ利用ポリシー強制の形式的検証など

# 4.
# 安全性の種類と傾向
## どのような安全性特性が対象か

# 安全性特性の傾向




・記号論的安全性(秘匿性/認証性など)が最も扱われている安全性特性
・サイドチャネル耐性や機能的等価性の検証も多い
・耐量子性を含む計算論的安全性の検証も増加傾向にある

# 安全性特性の詳細：秘匿性と認証性,サイドチャネル耐性

## 秘匿性と認証性

秘匿性と認証性は暗号プロトコルの基本的な安全性特性だが、
現在の研究では単なる古典的な定義を超え、より細かな変種や組み合わせ、より強い保証を提供する定式化が研究されている

## サイドチャネル耐性

定数時間実行（Constant-Time）
プログラムの実行時間が秘密データに依存しないことを保証する特性
マイクロアーキテクチャレベルでは、キャッシュアクセスパターンなどの観測可能な挙動が秘密に依存しないことを要求する

- FSLHによる投機的実行からの保護
- マイクロアーキテクチャCFI（$\mu$CFI）
- エンクレーブシステムのタイミング隔離

# 安全性特性の詳細：機能的等価性と正当性

## 機能的等価性

二つのプログラムやシステムが同じ機能を提供することを保証する特性。特にコンパイルやバイナリ変換の前後で挙動が変わらないことを検証する研究で用いられる

## コンパイル正当性

コンパイルされたプログラムが元のプログラムの意味論を保存することを保証する特性。SECOMPのようなセキュアコンパイラの研究で重要

## 実装可能性

プロトコルやシステムの仕様が実際に実装可能であることを保証する特性。SPROUTのようなシンボリックマルチパーティプロトコル検証で用いられる

## 回路の正当性

ゼロ知識証明回路が計算を正しくエンコードしていることを保証する特性。ACL2を用いた検証が行われている

# 安全性特性の詳細：プライバシー特性と計算論的安全性

## プライバシー特性

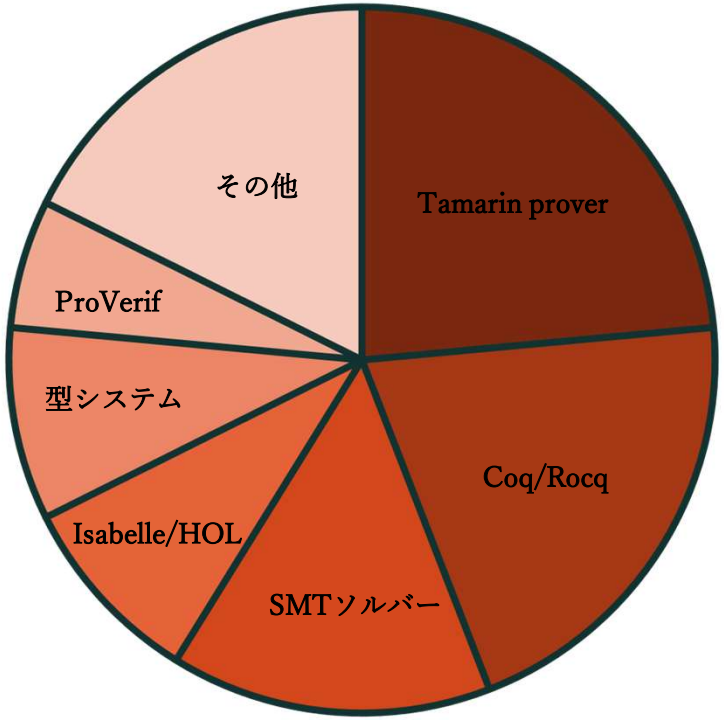
- 差分プライバシー：データベースへのクエリ結果が個々のデータ項目の存在に過度に依存しないことを保証
  Sensitivity：関数の入力の変化に対する出力の変化の度合いを定量化
- 情報フロー：システム内での情報の流れを制御し、機密情報の漏洩を防止

## 計算論的安全性

- 計算量的仮定に基づく安全性：PRF、EUF-CMA安全な署名などの仮定に基づく安全性
- 耐量子性：量子コンピュータによる攻撃に耐性のある安全性
- 構成的証明：より複雑なプロトコルの安全性を基本的な仮定から構成的に証明

# 5.
# 検証ツールの傾向
## どのような形式手法が対象か

# 検証ツールの傾向



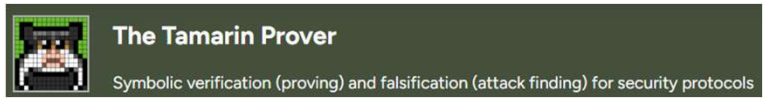

・Tamarin proverが暗号プロトコル検証で最も多く使用されている
・Coq/Rocqは低レベルシステムの検証や理論的証明に利用されている
（SMTソルバーは様々な形式検証ツールのバックエンドとして使われている,また直接使用する論文もある）

# 主要な形式検証ツール：Tamarin prover

特徴

- 暗号プロトコルの形式的検証に特化
- 無限セッション数のプロトコルも検証可能
- 攻撃者モデルを柔軟に記述可能
- プロトコルの状態遷移を記述しやすい

適用例




https://tamarin-prover.com/

- iMessage PQ3の安全性検証
- KEMベースのプロトコル分析
- Aggregate署名を利用したプロトコル

# 主要な形式検証ツール：Coq/Rocq

https://rocq-prover.org/

特徴

- 汎用的な証明支援系
- 依存型を持つ強力な型システム
- 対話的証明と自動証明の組み合わせ
- 高度に表現力のある論理体系

適用例

- SECOMPによるコンパイラの安全性証明
- エンクレーブシステムのタイミング隔離
- StrandsRocqによるプロトコル証明
- FSLHの相対的セキュリティ証明
- 名義状態分離証明(Nominal State-Separating Proofs)

# 主要な形式検証ツール：その他のツール

## SMTソルバー

多くの形式検証ツールのバックエンドとして使用,Z3などが一般的, AuthSaberやSQUIRRELなどで活用

## Isabelle/HOL

高階論理に基づく証明支援, x86-64バイナリの検証やセキュリティポリシーの検証に利用

## ProVerif

暗号プロトコルの検証ツール
OPC UAの検証などに利用

## モデル検査器

SPINなどのモデル検査器がライトニングネットワークの検証などに利用

# 6.
# 研究事例の詳細

全研究論文 (44例)

受賞研究論文の紹介 (2例)

# 各論文毎の詳細一覧（全研究成果）

| 論文名 | 対象 | 安全性 | 貢献 | 手法(ツール等) |
|---|---|---|---|---|
| SECOMP: Formally Secure Compilation of Compartmentalized C Programs | コンパートメント化された(未定義動作を含む)Cプログラムに対するコンパイラ | 動的侵害や相互不信を考慮に入れたRobustly Safe Compilation（低レベルコードで攻撃可能であればコンパイル前のソースコードにおいても攻撃可能だったはず） | 未定義動作の影響範囲が原因のコンパートメントに制限されるという性質がコンパイル後のコードでも保存されることを形式的に保証するコンパイラSECOMPを設計・開発 | CompCert(Coqで検証された現実的なCコンパイラ)を拡張 |
| Testing side-channel security of cryptographic implementations against future microarchitectures | x86プログラム | サイドチャネルに関する非干渉性(秘密情報部分のみ異なる2状態から実行を始めたとき、漏洩するトレースが等しい) | 漏洩モデル用DSLとテスティングツールの開発、最近提案されたマイクロアーキテクチャレベルの最適化に関して主流の暗号実装に対する大規模な漏洩テスト | 1. 漏洩モデルを DSL LmSpec で記述<br>2. 漏洩モデルが指定する漏洩を ランダムテスティングツール LmTest |
| Foundations for Cryptographic Reductions in CCSA Logics | Squirrel向けの暗号公理（PRF、EUF-CMA安全な署名などの計算量的仮定） | 計算量的安全性 | 1. bi-deduction の判定を行うシミュレータSとして確率的PTIMEを許容することで、bi-deductionの概念を改良<br>2. Bi-deductionを確立するための発見的証明アルゴリズムを実装し、いくつかのケーススタディに成功<br>具体的には、ハッシュ・ロック・プロトコルの強秘匿性、EUFとPRFに基づくベーシックハッシュ、秘匿認証プロトコルの匿名性、Needham-Schroeder-Lowe公開鍵プロトコル、CPAゲーム、CCA2ゲーム。 | 独自手法 |
| SpecMon: Modular Black-Box Runtime Verification of Security Protocols | 暗号プロトコルの実装 | プロトコル仕様に準拠していること，認証性，秘匿性 | プロトコルの仕様と実装の検証ギャップを解消するためのブラックボックス型ランタイムモニタリングシステムを提案 | Tamarin prover |
| Block Ciphers in Idealized Models: Automated Proofs and New Security Results | ブロック暗号ベースの構成 | 計算論的安全性 | 理想化された暗号モデルにおけるブロック暗号ベースの構造の安全性を自動証明するフレームワークを提案・実装 | - |
| Compositional Verification of Composite Byzantine Protocols | Byzantine Fault-Tolerant (BFT) プロトコル | Byzantine耐性下でのSafetyとLiveness | BFTプロトコルのSafetyとLivenessの両方を機械的に検証する初めてのフレームワークを提案 | Rocq（旧Coq） |
| $\mu$CFI: Formal Verification of Microarchitectural Control-flow Integrity | 命令セットアーキテクチャ | 定数時間実行への違反，制御フロー整合性への違反など | 定数時間実行と制御フロー整合性などを統合した汎用的な安全性である$\mu$CFIを提案し，その自動検証手法を開発．RISC-V CPUに対して新たな脆弱性を発見 | 独自手法 |
| Keeping Up with the KEMs: Stronger Security Notions for KEMs and automated analysis of KEM-based protocols | シンボリックKEMモデルライブラリ： KEMの核となるバインディング特性を複数のバリエーションと共に形式化したもの | IND-CCAの安全性 + ロバスト性 | 1. KEMの計算結合特性の新しい階層構造の紹介<br>2. プロトコルの安全性を自動的に解析するための記号解析手法を開発し,Tamarin prover に実装<br>3. 著者らが「再カプセル化攻撃」: "Re-encapsulation attacks" と呼ぶ, 新しいタイプのプロトコル攻撃を発見 | Tamarin prover |

# 各論文毎の詳細一覧（全研究成果）

| 論文名 | 対象 | 安全性 | 貢献 | 手法(ツール等) |
|---|---|---|---|---|
| The Privacy-Utility Trade-off in the Topics API | 1. トピックAPI:トピックAPIは，Chromeのサードパーティ Cookie廃止の影響に対処するべく，Googleが提案したPrivacy Sandboxと呼ばれるAPIセットの1つ.<br>3. ユーザのプライバシーを保護しながら，パーソナライズ広告関連の様々な機能を提供. | トピックAPIの利用が（インターネットユーザにとっての）プライバシーと（IBA（Interest-Based Advertising）企業にとっての）有用性に与える影響，それらの関連性，および，トレードオフ. | 1. 定量的情報フロー理論に基づくトピックAPIの新しいモデルの提案<br>2.個人に関する新しい理論的なプライバシー結果の提供<br>3. IBA企業に関する新たな理論的な有用性結果の提供<br>4. インターネットユーザの閲覧履歴と関心のあるトピックの分析用に新たなデータセットを提供<br>5. 理論モデルの予測を検証する実データに関する実験結果の提供 | 1. 定量的情報フロー<br>2. QIF を利用して，トピックAPIとサードパーティ Cookieの両方によって引き起こされる情報漏洩を厳密にモデル化 |
| Specification and Verification of Strong Timing Isolation of Hardware Enclaves | HWエンクレーブ（Intel SGX等）を用いたシステムのレジスタ転送レベル設計 | 同じマシン内の悪意あるエンクレーブによるタイミングサイドチャネル攻撃耐性（強いタイミング隔離） | エンクレーブシステムにおける強いタイミング隔離の形式化と機械証明、マルチコア・パイプラインRISC-Vシステムのケーススタディ | 1. ルールベースのHW記述言語KoikaによるHW動作のハイレベル推論<br>2. Coq-SMTチェインによるモジュール的な隔離証明 |
| Payout Races and Congested Channels: A Formal Analysis of Security in the Lightning Network | ライトニング・ネットワーク（ブロックチェーンのスケーラビリティ問題解決のため、決済をオフチェーンで行う仕組みの1つ） | デッドロック、ライブロック、正しい中間動作に対応する5つの性質（支払いの有効性など） | シングルホップ取引を有限状態マシンとしてモデル化、既知の攻撃（幅輳攻撃：Hashed Timelock Contractsの上限数を超えてしまう）と新規の攻撃（Payout Race攻撃：不公平な取引が生じる）を発見 | モデル検査器SPIN |
| Verifiably Correct Lifting of Position-Independent x86-64 Binaries to Symbolized Assembly | x86-64バイナリ | 機能的同値性 | x86-64バイナリをsymbolized NASMアセンブリに正しさ証明付きで自動変換する手法を提案 | Isabelle/HOL |
| AuthSaber: Automated Safety Verification of OpenID Connect Programs | OpenID Connectプログラムの実装 | 認証性，トークンの秘匿性，リダイレクトURIのSafetyなどのOpenID Connectに対する安全性 | OpenID Connectプログラムの実装から状態遷移モデルを抽出し，安全性を自動検証する手法を提案 | SMTソルバー（Z3） |
| Verifiable Security Policies for Distributed Systems | 分散システムに対するセキュリティーポリシー | 秘匿性(Generalized Non-Interference Modulo Views: GNIV, 分散システム向けにした GNI) | 1. 言語非依存なセキュリティーポリシーを与えるための枠組を提案<br>2. ポリシー自体の妥当性(例: GNIV)検証および実装がポリシーを満たすことを検証するための技術を提示し証明検証系で形式化<br>3. 事例研究による評価でSMTベースの検証の適用可能性を実証 | 1. Observational Determinism,<br>2. SecCSL (OD推論の論理)<br>3. Isabelle/HOL (健全性証明)<br>4. Gobra (Go の自動コード検証器) |
| Towards Automatic Discovery of Denial of Service Weaknesses in Blockchain Resource Models | ブロックチェーンのスマートコントラクト | (資源を過度に使用させられることによる)DoSに対する耐性 | 様々なブロックチェーンのDoS脆弱性を自動的に発見・攻撃を合成するツールを開発、9つの代表的ブロックチェーンを対象とする実験で、12件(うち新規10件)のDoS脆弱性を発見 | 時間付き多重集合書き換え, 大規模言語モデル, 字句解析 |
| PropertyGPT: LLM-driven Formal Verification of Smart Contracts through Retrieval-Augmented Property Generation  | スマートコントラクト | スマートコントラクトに求められる包括的な安全性 | 検証に用いるプロパティを自動生成するPropertyGPTを提案<br>1. コンパイラと静的解析のフィードバックを外部オラクルとして使用し、プロパティを繰り返し修正<br>2. プロパティをランク付けするための複数の類似性を考案<br>3. 生成されたプロパティの正しさをformal verificationするための専用の検証器を設計<br>37件のテストを実行し、26件のCVE付脆弱性の検出に成功、12件のゼロデイ脆弱性を発見し、8,256ドルの報奨金を得た。 | 人間の記述した既存プロパティから、未知のコード用にカスタマイズされたプロパティを自動生成して、生成したプロパティを専用の検証器に与え、検証結果を得る |

# 各論文毎の詳細一覧(全研究成果)

| 論文名 | 対象 | 安全性 | 貢献 | 手法(ツール等) |
|---|---|---|---|---|
| A Formal Approach to Multi-Layered Privileges for Enclaves | Palantir（提案システム） | 初期状態の正当性・実行の一意性，実行の改ざん耐性，秘匿性 | TEEのエンクレーブへの安全な機能拡張を実現する多層権限モデルを提案 | BoogieIVL，Z3など |
| PQConnect: Automated Post-Quantum End-to-End Tunnels | PQConnect（提案プロトコル） | 秘匿性，量子攻撃者に対する秘匿性，前方秘匿性，量子攻撃者に対する秘匿性，認証性 | クライアント-サーバ間の耐量子エンドツーエンド暗号化トンネルを張る仕組みを提案 | Tamarin prover |
| VeriBin: Adaptive Verification of Patches at the Binary Level | ベンダが提供しているパッチ適用済みソフトウェア | 元のバイナリとパッチを適用したバイナリ間の機能的等価性 | 1. バイナリのSafe to Apply(StA)パッチの定義を定式化し、StAプロパティを定義<br>2. StAの特性を検証するための効率的なパスベースの技術を設計,著者らの手法は、標準的な手法と比較して解析時間を75%短縮<br>3. アナリストが、パッチが適用されたバイナリでは得られないドメイン固有の情報を提供することで、VeriBinの分析を補強できるように、著者らの分析を適応性のあるものにしている<br>4. 3つの異なるデータセットから得られた86組のバイナリでVeriBinを評価,結果,VeriBinはパッチの振る舞いを93.0%の精度で、しかも誤検出なしに特定できることがわかった | VeriBin |
| From Control to Chaos: A Comprehensive Formal Analysis of 5G's Access Control | 5Gデバイスおよびサービスの通信，接続，ルーティングを管理する5Gコアネットワークにおけるアクセス制御 | アクセス制御のセキュリティ特性 | 1. 5Gコアネットワークのアクセス制御メカニズムを分析するための包括的な形式分析フレームワークである「CoreScan」を開発<br>2. 間接通信モードと5Gローミングを考慮した5Gコアネットワークのアクセス制御メカニズムに関する最初の包括的な形式モデルを構築<br>3. CoreScanを用いて，3GPP Release 18で規定されているアクセス制御メカニズムをモデル化し，61のセキュリティ特性を検証した結果，5つの新規発見（特権昇格，機密情報の漏えいなど）を含む10種類のアクセス制御攻撃を発見 | Assume-Guarantee型の推論アプローチを活用した構成的検証技術を採用.スプリットアサーションの原理を活用し，システムモデルとセキュリティ特性を独立したコンポーネントに分解することで,スケーラブルで効果的なモデルチェックを実現. |
| Automatic verification of Finite Variant Property beyond convergent equational theories | 等式理論 | 有限変異性（Finite Variant Property：FVP） | 代表的な検証ツール(ProVerif, Tamarin, Maude-NPA, DeepSec, AKiSs) の安全性判定の決定可能性に直結する「有限変異性（Finite Variant Property：FVP）」の新しい準決定的手法（semi-decision procedure）の提案と、複数の等式理論に対してプロトタイプ実装でFVPをもつか否かを判定 | 項書き換え系 |
| Secrecy by typing in the computational model | 共通鍵暗号、公開鍵暗号、および連結を利用する暗号プロトコル | 秘匿性の弱い概念である「非導出可能性（non-deducibility）」 | 1. CCSA (Computationally Complete Symbolic Attacker) フレームワークで記述されたプロトコルを型検査できる型システムを提案し、SQUIRRELツールにおいて型検査手続きを実装<br>2. Needham-SchroederプロトコルやDenning-Saccoプロトコルといった古典プロトコル、鍵管理に関するISO/IEC 11770標準のプロトコルに適用 | 型システム |
| Zero-Knowledge Proofs from Learning Parity with Noise: Optimization, Verification, and Application | ASIACRYPT12で提案されたLPN（Learning Parity with Noise）問題に基づく耐量子ゼロ知識証明（の最適化） | 部品のコミットメントの正当性・束縛性・秘匿性、ZKPの完全性・健全性・ゼロ知識性 | AC12方式の最適化、オープンソース実装の不備発見、部品のコミットメントと最適化ZKP方式の形式手法による証明、応用として電子投票プロトコルの提案 | EasyCrypt |

# 各論文毎の詳細一覧(全研究成果)

| 論文名 | 対象 | 安全性 | 貢献 | 手法(ツール等) |
|---|---|---|---|---|
| SMT-based Automation for Overwhelming Truth | 「圧倒的確率」で起きる事象を含む推論、ステートフルなプロトコル（カウンタを含むもの） | 計算論的検証ツールSQUIRRELで検証可能な性質 | SQUIRRELで従来自動検証できなかった推論を、SQUIRREL論理から演繹的プログラム検証プラットフォームWHY3論理への変換手法を与えSMTソルバにより自動推論できるように拡張 | SQUIRREL、WHY3 |
| Session Types for the Concurrent Composition of Interactive Differential Privacy | interactiveなものを含む差分プライバシメカニズム | 差分プライバシ性 | interactiveな差分プライバシメカニズムを表現するΠ計算ベースの言語を導入し，セッション型による型推論を用いて差分プライバシを満たしていることを自動で検証する手法を開発 | Π計算，Fuzz言語 |
| Gradual Sensitivity Typing | 差分プライバシメカニズム | Sensitivity，差分プライバシ性 | 差分プライバシメカニズムのSensitivityをより柔軟に扱えるような型システムを提案 | Gradual Typing |
| Strands Rocq: Why is a Security Protocol Correct, Mechanically? | セキュリティプロトコル | 認証、秘匿性 | 1. 証明支援系上でストランド空間を機械化実装(StrandsRocq)、合成的な証明を可能とする maximal penetrator の概念を導入<br>2. NSLプロトコルや鍵管理APIの解析に利用し、既存の手書きの証明の問題点と解決策を指摘 | Rocq (旧Coq) |
| FSLH: Flexible Mechanized Speculative Load Hardening  | プロセッサ | 投機実行によるサイドチャネル攻撃への耐性 | 1. Spectre v1に代表される種類のサイドチャネル攻撃から保護するプログラム変換FSLHを導入<br>2. 従来のSelective SLHより適用範囲を広げつつUSLHよりも少ないオーバーヘッドを実現<br>3. 投機実行を原因とする新たな漏洩が発生しないこと(相対的セキュリティ)を証明支援系で検証 | Rocq (旧Coq) |
| One For All: Formally Verifying Protocols which use Aggregate Signatures | Aggregate署名を利用したプロトコル | EUF-CMA安全性、およびEUF-CMA安全性の対象外だが重要な安全性（不正に生成された鍵に対する偽造に対する安全性（Spplitting-Zero攻撃など）） | 1. Aggregate署名の安全性要に記号論的モデルを新たに定義<br>2. Tamarin Prooverで記号論的モデルに基づいた検証 | Tamarin prover |
| Formal Analysis of Random Nonce Misuses in Cryptographic Protocols | Nonceを使用する暗号プロトコル | Nonceが正しく生成されない場合の秘匿性，認証性，前方秘匿性，リプレイ攻撃耐性，完全性など | Nonceの不適切な仕様が起きた場合にプロトコルの安全性に与える影響を体系的に分析するフレームワークを提案 | Tamarin prover |
| Symbolic Parallel Composition for Multi-language Protocol Verification | 複数のプロトコルの合成 | 秘匿性，認証性，完全性，前方秘匿性，プロトコル合成後の安全性 | 複数のシンボリック検証器（ProVerif, Tamarinなど）で検証されたプロトコルを並列合成するための多言語検証フレームワークを提案 | HOL4 |
| Nominal State-Separating Proofs | 名義態分離証明(Nominal State-Separating Proof) | ElGamal暗号システム内で、名義状態分離証明(Nominal State-Separating Proofs)を実証 | 1. 暗黙の変数名再命名に関する非形式的な要件を明確化<br>2. 状態分離証明における状態変数のキャプチャの問題を解決<br>3. 名義状態分離証明の形式的意味論を定義<br>4. Nominal-SSProveを Rocqにおけるデモストレーターとして実装<br>5. ElGamalの公開鍵一回秘密保持からDecisional Diffie-Hellman(DDH)への還元を機械化.<br>6. 結果,SSProveにおける還元機械化に2つの誤りを特定 | 名義状態分離証明のために、SSProveを拡張したライブラリであるNominal-SSProveを実装 |

# 各論文毎の詳細一覧(全研究成果)

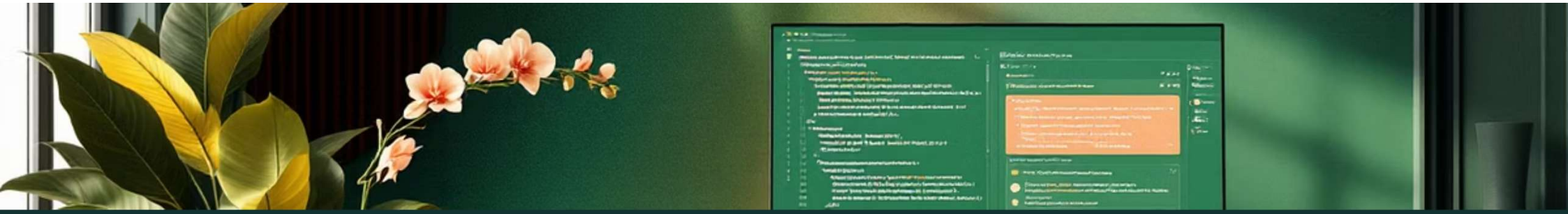
| 論文名 | 対象 | 安全性 | 貢献 | 手法(ツール等) |
|---|---|---|---|---|
| Formal Robustness for Cyber-Physical Systems under Timed Attacks | サイバーフィジカルシステム（CPS） | 特定のタイミングでトリガーされる“時間指定攻撃”に対する安全性 | 1. サイバーフィジカルシステムの時間指定攻撃に対する安全性のロバスト性を定量分析するための形式的なフレームワークを提案<br>2. 時間指定攻撃に対する安全性のロバスト性を特徴付ける一連の時間関連プロパティを導入し，これらのプロパティを確立するために，ダイナミクスのタイミングに焦点を当てた推論手法のシステムを開発. | 連続的及び離散的な挙動が混在する動的システムをモデル化するための形式主義である“ハイブリッドプログラム”．ハイブリッドプログラムの論理であり，安全性プロパティを仕様記述及び検証するために使用される“微分動的論理（dL）”. |
| Automated Verification of Consistency in Zero-Knowledge Proof Circuits | ゼロ知識証明の計算処理をエンコードするゼロ知識回路 | 回路が計算を正しくエンコードしていること（正当性） | 1. 回路の正当性に関する枠組み<br>2. 素体に対するACL2ライブラリとモデル、その検証ツール<br>3. 階層的な回路を表現するモデル、その検証ツール<br>4. 回路の検証と既存のゼロ知識システムのバグを2つ発見 | 定理証明支援システムACL2 |
| Integer Reasoning Modulo Different Constants in SMT | 異なる剰余（拡大体と素体など）の式（整数制約）を含む暗号プロトコル（multimodular systems）とその実装 | モンゴメリ演算やゼロ知識証明実装の正当 | 従来のSMTソルバで扱いが困難だったmultimodular systemsを、剰余ごとにサブシステムに分割し局所的な推論を行った後に情報交換させることによって高速化 | SMTソルバ：cvc5、代数計算：Singular、整数線形計画問題求解：glpk |
| Structural operational semantics for functional and security verification of pipelined processors | パイプライン化されたマイクロプロセッサ（Arm, x86, RISC-Vなどのマイクロアーキテクチャ） | パイプラインハザード(分岐予測、キャッシュ),特権チェック,Spectre/Meltdown脆弱性 | 機能的正当性と安全性（サイドチャネル攻撃耐性）を捉えた構造的操作的意味論の定式化、Haskellベースのモデル検査により Arm/x86/RISC-Vの既存モデルおよび実機挙動との整合性を実験的に評価 | 汎用アセンブリ言語pline |
| Relational Hoare Logic for Realistically Modelled Machine Code | アセンブリレベルのプログラム | Constant-Timeの保証（サイドチャネル攻撃耐性），機能的等価性など | 低レイヤのプログラムを表現するHoare styleの論理を導入し，Relationalな性質を検証できる手法を開発 | Relational Hoare Logic，HOL Light |
| SPROUT: A Verifier for Symbolic Multiparty Protocols | シンボリックマルチパーティプロトコル | 実装可能性 | シンボリックマルチパーティプロトコルの実装可能性を検証するツールSproutを提案 | $\mu$CLP，MuVal |
| A Tale of Two Worlds, a Formal Story of WireGuard Hybridization | PQ-WireGuardおよび、その改良版PQ-WireGuard*とハイブリッド構成のHybrid-WireGuard | PQ-WireGuardに求められる包括的な安全性 | 1. 先行研究の形式検証では、攻撃者の前提条件が弱くUnknown-Key-Share攻撃が見落とされていることを指摘<br>2. Sapic（これ用の形式記述から、TamarinやProverifの検証コードを生成できるやつ）をつかって、メッセージングプロトコルWireGuardの耐量子計算機版PQ-WireGuardの安全性を検証し、いくつかの脆弱性を発見<br>3. 改良版のPQ-Wireguard*とHybrid-WireGuardを提案。SAPICで安全性を検証した<br>4. KEMとしてClassic McElieceとML-KEMを使ったHybrid-WireGuardのRust実装を行った | SAPIC |
| A Formal Analysis of Apple's iMessage PQ3 Protocol | PQ3（AppleのiMassageのプロトコル） | 秘匿性，認証性，前方秘匿性，耐量子前方秘匿性など | 1. PQ3に対して，Tamarin proverを用いて安全性検証を行い，プロトコルが仕様通りの安全性を満たしていることを確認<br>2. いくつかの設計上のトレードオフや改善点を指摘 | Tamarin prover |

# 各論文毎の詳細一覧(全研究成果)

| 論文名 | 対象 | 安全性 | 貢献 | 手法(ツール等) |
|---|---|---|---|---|
| Towards Practical, End-to-End Formally Verified X.509 Certificate Validators with Verdict | X.509証明書 | エンドツーエンド正当性 | カスタマイズ可能な検証ポリシーを備えたエンドツーエンド形式検証済みX.509証明書検証ツールを開発 | Verus |
| PICACHV: Formally Verified Data Use Policy Enforcement for Secure Data Analytics | データ利用ポリシーを自動的に強制する新たなセキュリティモニター「PICACHV」の導入<br>PICACHVと呼ばれるプロトタイプの実行時セキュリティモニターを実装 | 健全性(soundness) | 1. 言語ベースのセマンティクスの複雑さを回避するために,関係演算子を非分類セマンティクスで形式化した.<br>さらに,そのセマンティクスを形式化し,Coqで soundness の機械的証明を提供<br>2. 分析タスクを実行する際,これらのデータ使用ポリシーを透明に強制する動的セキュリティモニターであるPICACHVを設計<br>著者らのフレームワークには複数の最適化が組み込まれており,実験結果から,実行時オーバーヘッドは僅かであることが示された.<br>3. PICACHVを確立されたデータ分析フレームワークであるPolarsに移植し,多様なデータ処理環境における適応性を示すための評価を実施 | Coq |
| OwlC: Compiling Security Protocols to Verified, Secure, High-Performance Libraries | 暗号プロトコル | 計算論的安全性 | 1. 計算論的安全な暗号プロトコルの検証済み，相互運用可能，サイドチャネル耐性のある実装を自動的に生成できる初めてのツール“OwlC”を提案<br>2. プロトコル設計が与えられると，OwlCはその設計のセキュリティプロパティを満たすことが自動的に検証された，高性能かつメモリ安全なRustライブラリを生成<br>3. 14種類の小規模なケーススタディプロトコル（Owlを提案する際に使用されたプロトコル）に加えて，2種類の新しい大規模なケーススタディプロトコル（WireGuard及びHPKE）を通じて，OwlCの有用性とスケーラビリティを実証 | OwlC |
| A Comprehensive Formal Security Analysis of OPC UA | 標準化された産業用制御システム（ICS）プロトコルであるOPC UAのバージョン1.04および1.05 | 認証と秘匿 | 1. RSAベースとDHベースの両方を検証、新たな脆弱性が明らかとなり、それらはOPC Foundationに報告・認知<br>2. 証明可能に安全な修正案を設計・提案し、その多くが次期標準バージョンに反映 | ProVerif |

## 収集対象論文

| 44 | 6 | 2024-2025 |
|---|---|---|
| 論文総数 | 会議数 | 発表年 |
| | CCS, NDSS, S&P, CSF, CAV, USENIX | 2024年後半～2025年前半 |



# 受賞研究論文の紹介 1/2：PropertyGPT

## PropertyGPT: LLM-driven Formal Verification of Smart Contracts through Retrieval-Augmented Property Generation (NDSS2025)

### 研究対象

スマートコントラクト

### 対象安全性

スマートコントラクトに求められる包括的な安全性

### 貢献

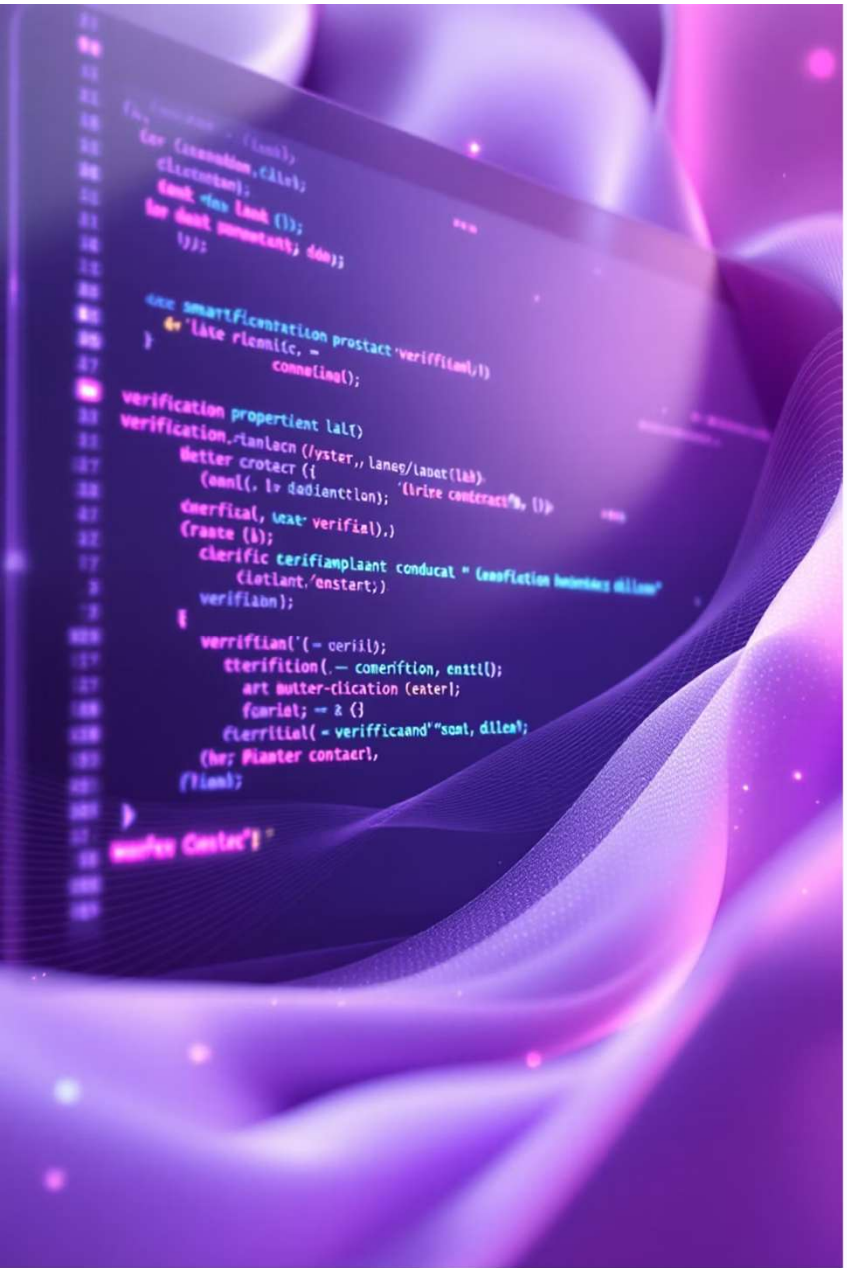
1. 検証に用いるプロパティを自動生成するPropertyGPTを提案
2. コンパイラと静的解析のフィードバックを外部オラクルとして使用し、プロパティを繰り返し修正
3. プロパティをランク付けするための複数の類似性を考案
4. 生成されたプロパティの正しさをformal verificationするための専用の検証器を設計

### 使用手法

人間の記述した既存プロパティから、未知のコード用にカスタマイズされたプロパティを自動生成して、生成したプロパティを専用の検証器に与え、検証結果を得る

### ポイント

LLMを形式検証に応用した挑戦的な研究。スマートコントラクトの包括的な形式検証を推進するために、新しいLLMベースのプロパティ生成ツールPropertyGPTを提案。37件のテストを実行し、26件のCVE付脆弱性の検出に成功、12件のゼロデイ脆弱性を発見し、8,256ドルの報奨金を獲得



# PropertyGPT:
# スマートコントラクト形式検証

## 対象と安全性要件

- スマートコントラクト
- スマートコントラクトに求められる包括的な安全性

### 課題

- 形式検証には、様々なスマートコントラクトに対して、個別に検証用コードを書く必要がある
- LLMで自動生成したプロパティが以下を満たす保証：

① コンパイル可能

② 適切

③ 検証可能

### 貢献

**検証プロパティを自動生成するツールを提案**

- コンパイラと静的解析のフィードバックを活用
- プロパティをランク付けする類似性指標を考案
- 専用の検証器を設計

# PropertyGPTの実証成果

**37**

テスト実行数

様々なスマートコントラクトに対して検証を実施

**26**

CVE脆弱性検出

既知の脆弱性を自動的に検出することに成功

**1**

ゼロディ脆弱性発見

未知の脆弱性を新たに発見

**$8,256**

報奨金獲得

発見した脆弱性に対する報奨金

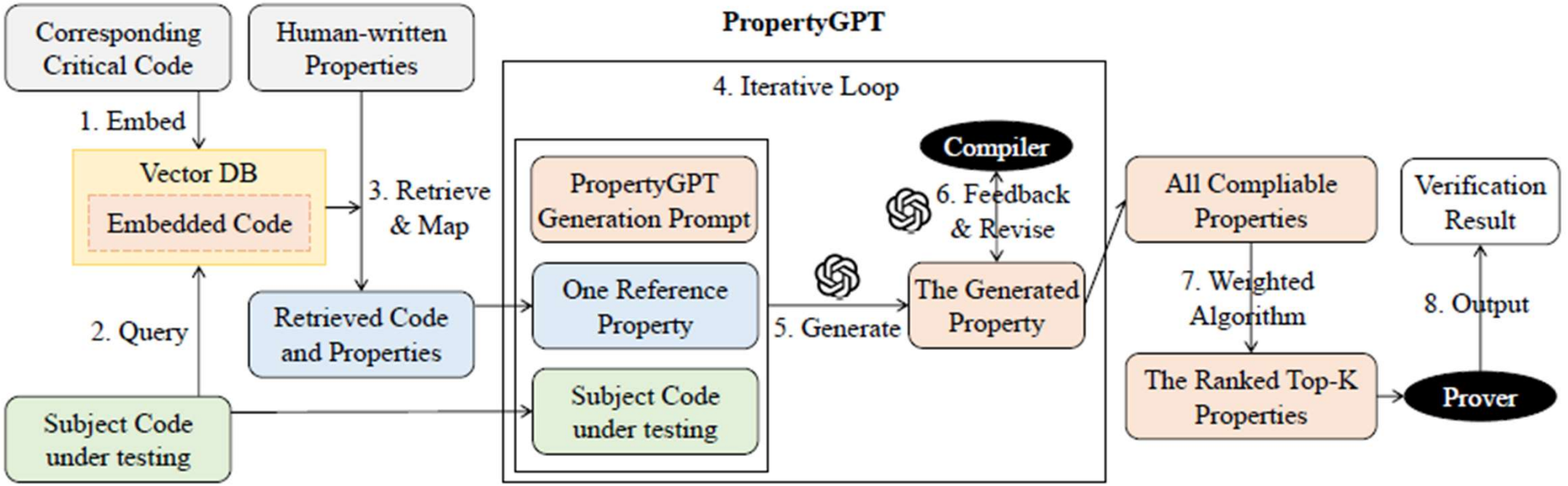



Fig. 1: A high-level workflow of PropertyGPT.

# PropertyGPTのワークフロー

・RAGベースのプロパティ生成機能を搭載したPropertyGPTは、人間の記述した既存プロパティから未知のコード用にカスタマイズされたプロパティを自動生成し、それを検証器に与えて検証結果を得る
・GPT-4を基盤技術として活用している

# 受賞研究論文の紹介 2/2 詳細：FSLH

## FSLH: Flexible Mechanized Speculative Load Hardening（CSF2025）

### 研究対象

プロセッサ

### 対象安全性

投機実行によるサイドチャネル攻撃への耐性

### 貢献

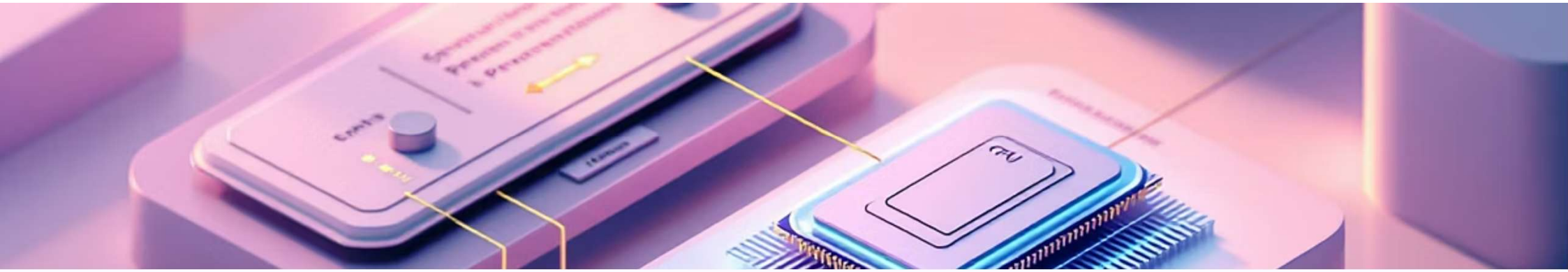
1. Spectre v1に代表される種類のサイドチャネル攻撃から保護するプログラム変換FSLHを導入
2. 従来のSelective SLHより適用範囲を広げつつUSLHよりも少ないオーバーヘッドを実現
3. 投機実行を原因とする新たな漏洩が発生しないこと（相対的セキュリティ）を証明支援系で検証

### 使用手法

Rocq（旧Coq）

### ポイント

従来のSelective SLHはオーバーヘッドは少ないが定数時間暗号プログラムに特化され、USLHは適用範囲が広いがオーバーヘッドが大きい。提案されているFSLH（Flexible SLH）は両者のいいとこ取りになるように、また従来のものの一般化になるように設計。相対的セキュリティという概念を導入し、証明支援系Rocqで検証



# FSLH

## 対象と安全性要件

- 対象：プロセッサ
- 安全性要件：投機実行※によるサイドチャネル攻撃への耐性

## ポイント

- Rocq（旧Coq）で約4300行の証明
- 相対的セキュリティという概念を導入し、形式的に検証

## 貢献

- **Spectre v1**に代表される種類のサイドチャネル攻撃から保護するプログラム変換FSLHを導入
- 従来の**Selective SLH**より適用範囲を広げつつ**USLH**よりも少ないオーバーヘッドを実現
- 投機実行を原因とする新たな漏洩が発生しないこと(相対的セキュリティ)を証明支援系で検証

※**黒太字**は、このあとのスライドで説明します。

※投機実行: 分岐を予測して分岐先を予めパイプラインに読み込む方法

# Spectre v1攻撃の仕組み

## 攻撃コード例

**Listing 1** (Spectre v1 gadget).

```
if i < a1_size then
  j ← a1[i];
  x ← a2[j]
```

$a_1$は公開配列、iは信用されない入力

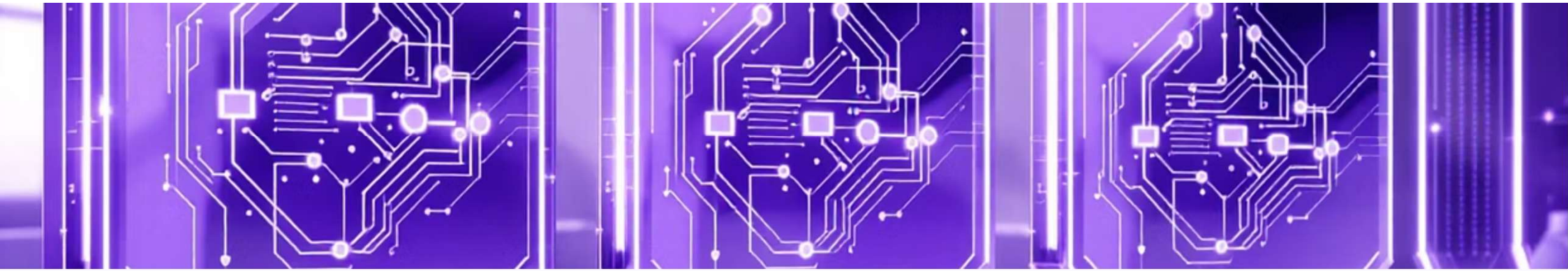
## 攻撃の流れ

1. a2[j]の値がxにロードされる
2. 分岐予測に失敗したと気づいたプロセッサが実行をロールバック
3. プロセッサが実行をロールバックするが、キャッシュに痕跡が残る

## 情報漏洩

タイミング攻撃によりjの値が漏洩する可能性がある

攻撃者が分岐予測を制御し、iの値が範囲外でもthen節を実行させる



# SLH（Speculative Load Hardening）の種類

## Selective SLH

マスクするかどうかをセキュリティレベルによって決定

- オーバーヘッドは少ない
- 定数時間暗号プログラムに特化
- 適用範囲が限定的

## Ultimate SLH

条件部分に秘密の入力を用いることを許可

- 適用範囲が広い
- オーバーヘッドが大きい
- 汎用性が高い

## Flexible SLH（提案手法）

両者のいいとこ取りを実現

- 適用範囲を広げつつ
- オーバーヘッドを削減
- 従来手法の一般化

・FSLHは従来のSLH手法の長所を組み合わせ、短所を克服するように設計されている
・相対的セキュリティという概念を導入し、証明支援系Rocqで形式的に検証されている

# FSLH 形式的証明の流れ

### 基本言語と意味論の定義

単純なwhile言語とその逐次的意味論を定義
意味論はsmall-step操作的意味論を採用

### 投機的意味論の定義

攻撃者の分岐予測制御下(*_FORCE規則)により漏洩の可能性を表す投機的意味論を定義

### 理想論的意味論の定義

攻撃者の分岐予測制御下でも漏洩しないことを表す理想的意味論を定義
分岐条件が秘密ならばelse分岐、添字が秘密ならば0を使用

## 主要補題と定理の証明

FSLH変換後のプログラムが,投機実行を原因とする新たな漏洩が発生しないこと(相対的セキュリティ)を保証することを形式的に証明

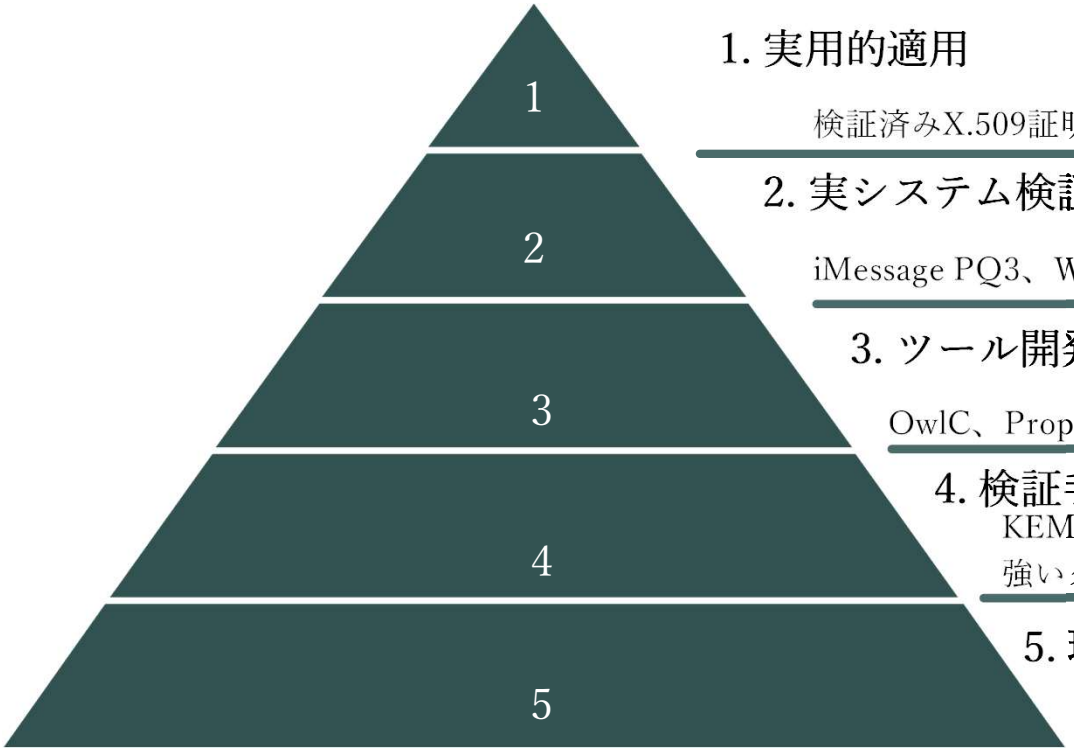
# 7. まとめと今後

適用先の分類

最も活発な研究領域

形式手法の課題と進展

# 適用先の分類




1. 実用的適用

検証済みX.509証明書検証、スマートコントラクト検証、バイナリパッチ検証など

2. 実システム検証

iMessage PQ3、WireGuard、OPC UA、5Gアクセス制御、ライトニングネットワークなど

3. ツール開発

OwlC、PropertyGPT、SAPIC、VeriBin、PICACHV、CoreScanなど

4. 検証手法

KEMモデル、差分プライバシー型システム、Relational Hoare Logic、
強いタイミング隔離など

5. 理論的基礎

暗号理論の形式化、等式理論のFVP、状態分離証明など

形式手法によるセキュリティ研究は、理論的基礎から実用的適用まで幅広い研究が進められている
特に、実システムの検証や実用的適用の研究が増えていて、形式手法の実践的価値が高まっていることがわかる

# 最も活発な研究領域

## 1
### 暗号プロトコル検証

Tamarinなどを用いた暗号プロトコルの形式検証が最も活発。特に耐量子暗号への移行に伴う新たな安全性分析が増加

## 2
### 低レベルセキュリティ

マイクロアーキテクチャセキュリティやバイナリレベルの検証が急速に発展。投機的実行攻撃対策の形式的保証が重要テーマ

## 3
### 実装の検証

理論と実装のギャップを埋める研究が増加。検証済み実装の自動生成や既存システムの検証技術が発展

これらの領域は相互に関連しており、例えば暗号プロトコルの検証結果を実装に反映する技術や、低レベルセキュリティの考慮を暗号プロトコル設計に取り入れる研究などが多く見られた

# 形式手法の課題と進展

## 1. 複雑性の課題

複雑なシステムの形式モデル化は依然として難しいが、構成的検証や階層的モデリングにより改善
5Gコアネットワークの検証やマルチコアプロセッサの検証など、複雑なシステムへの適用が進展

## 2. 自動化の進展

SMTベースの技術やAI活用により自動化が進展
PropertyGPTのようなLLM活用や、SMT-based Automation for Overwhelming Truthのような自動推論技術の発展が見られる

## 3. 理論と実装のギャップ

OwlCのような検証済み実装の自動生成や、VeriBinのようなバイナリレベルの検証技術により、理論と実装のギャップが縮小
検証結果を実際のシステムに反映する技術が発展


ご清聴ありがとうございました