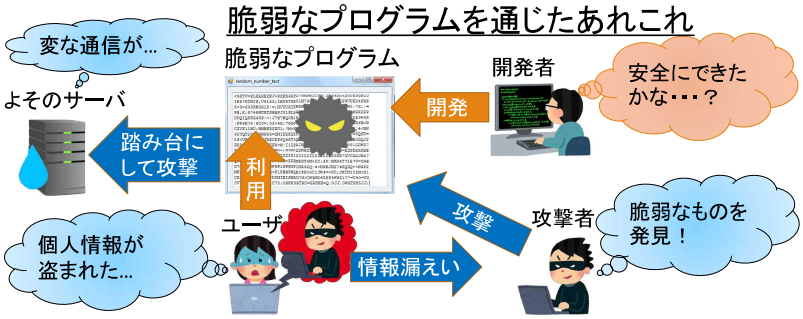
暗号学的期待値分布に基づくソースコードレベルでの汎用的脆弱性検証手法

研究者番号 50126
（大阪大学）矢内直人

セキュリティにおいて脆弱性は重要な課題

脆弱なプログラムを通じたあれこれ



脆弱性の汎用的解析を目指す

- 様々な脆弱性をどう包括・解析するか？
- 誤りを防ぐべく解析の自動化もしたい

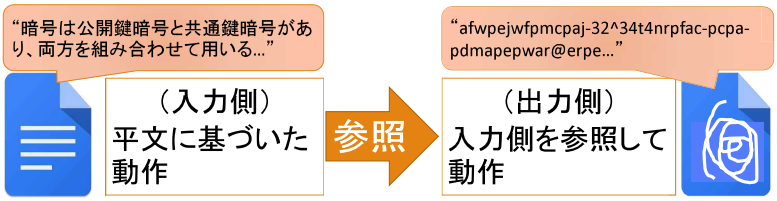
本研究は形式検証と暗号の解析を融合

形式検証：プログラムの“正しさ”を機械的に検証

融合

“入力-出力の依存度”からプログラムの“正しさ”解析

暗号：“入力と出力の依存度（安全性）”を解析



本研究の成果

脆弱性の定式化

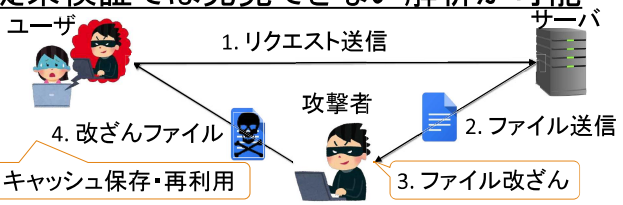
プログラムを二段階状態機械で表現

（入力側）ユーザの状態に基づいて動作 → 参照 → （出力側）入力側を参照して動作

脆弱性：依存度が変化する入力の存在 → 検証 → 二段目の出力候補とその確率が変わるか

事例研究）Web 技術の解析

従来検証では発見できない解析が可能



理論的研究　脆弱性の定式化

- 脆弱性＝“入出力の期待値分布変化”
  - 「脆弱性がない」＝「入力側の分布変化が起きても出力分布が変化しない」
  - 数学的には二段階の状態機械から定義

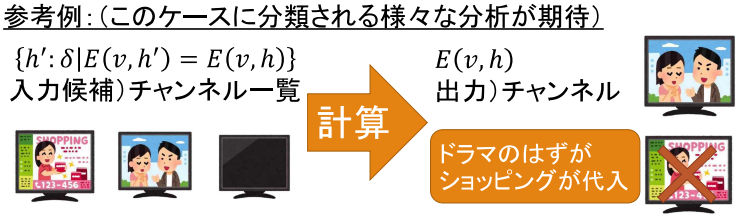
対象動作	プログラムP 上の記述	プログラム構文 $[P](v,\delta)$ … $(v,\delta)\in\mathbb{D}(\mathcal{V}\times\mathbb{D}\mathcal{H})$
出力	return	$\{(v,\delta)\}$
出力変数へ代入	$v:=E(v,h)$	$\{h:\delta\cdot(E(v,h),\{h':\delta\mid E(v,h')=E(v,h)\})\}$
出力変数ランダム選択	$v:\in D(v,h)$	$\{v':(\varepsilon\, h:\delta\cdot D(v,h))\cdot(v',\{h':\delta\mid D(v,h')(v')\})\}$

プログラムの各処理を置き換えることで、検証可能（上の表では一部のみを記載）

脆弱性定義1) 出力変数に違うものが代入

$$\{h:\delta\cdot(E(v,h),\{h':\delta\mid E(v,h')\neq E(v,h)\})\}$$

参考例：（このケースに分類される様々な分析が期待）



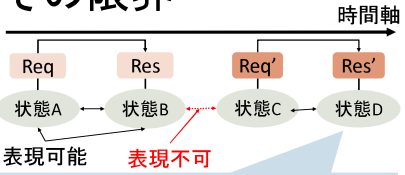
脆弱性定義2) 通常の動作と異なる分布で動作

$$\{v':(\varepsilon\, h:\delta'\cdot D(v,h))\cdot(v',\{h':\delta\mid D(v,h')(v')\})\}$$

応用的研究　Web 技術の解析

既存の解析研究とその限界

- HTTP, HTTPS が対象
- サーバ、クライアントのみ
- 通信の改ざん・盗聴



状態DはRes’ に対するユーザの状態であり、状態B の情報は含まない

三状態以上の状態遷移による変化が表現できない

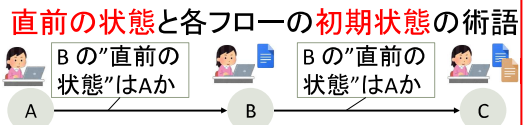
利用ツール：Alloy Analyzer (http://alloytools.org/)

今回）状態同士を紐づけるState クラスとその操作に関する二つの述語を実装

State
- 不変項目 — どの要素か
- 変化項目 — 何が変化するか

→ 同一の遷移フロー特定

直前の状態と各フローの初期状態の術語



A は”初期状態”か

新たに検証可能な内容：
- Web キャッシュ
- キャッシュ毒入れ攻撃（2015年）
- Web 詐称攻撃（2017年）

（コード公開予定）

コードの拡張によりHSTS やHPKP などの検証も期待