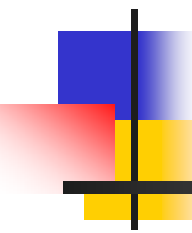
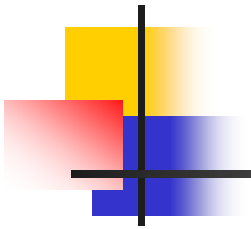


ソフトウェアセキュリティ

奈良先端科学技術大学院大学
情報科学研究科
関　浩之



目次

- セキュリティ技術の基礎
  - 暗号技術
  - アクセス制御技術
- 本講座の研究成果例
  - 匿名性と投票証明を実現する電子投票システム
  - Javaプログラムのセキュリティ検証
- 産学連携への展望

匿名性と投票証明を実現する電子投票システム

[1] 楫, 北川, 岡：特願2002-091392, 平成14年3月28日出願.

[2] T. Kitagawa, H. Oka and K. Kaji: An anonymous questionnaire system for rating faculty courses in universities, Proc. Int'l Symp. on Information Theory and Its Applications, 559-562, Oct. 2002.

授業評価アンケート

- 結果を教官にフィードバックし，講義内容の改善に利用．
- 「無記名アンケート」であることが望ましい．
  - 批判的な意見を積極的に収集したい．

紙ベースで行っていたが，電子的なもので置き換えたい．

⇒ セキュリティ研究で得られた知見を活用し，

- 研究としても価値があり，
- 実際に社会的な役にもたつ．

ような方式の実現を目標として，システムを構築する．

要求される条件

通常のネットワーク投票と共通の要求

- 受講生のみがアンケートに回答可能であること.
- 受講生であっても，2回以上回答できないこと.
- 非受講生はアンケートに回答できないこと.
- 誰がどのような回答を行ったか，本人以外わからないこと.
- 実現にあたり，特殊なハードウェア等が必要でないこと.
- Webベースのインターフェイスを提供すること.

仮定できる前提条件

- 通常の投票方式：
  - 集計者は票の改竄等を行いうる．
- 授業評価アンケート：
  - 大学が票の改竄等を行う動機は希薄．

一方，学生の信頼を得るためには．．．
アンケート回答の無記名性を守る仕組みが必要．

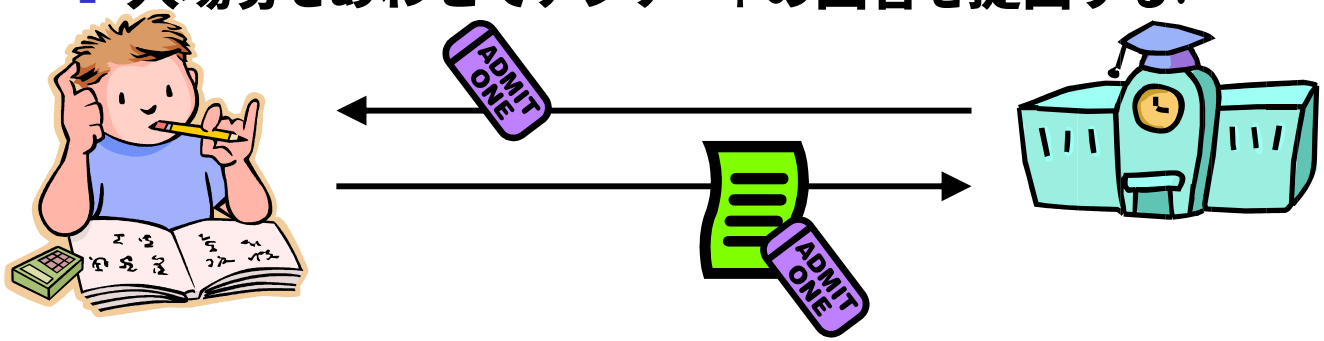
研究の目標

- 授業評価アンケートに特化した**通信手順の開発**
- 開発手順に従ってアンケートを実施する**システムの構築**
- 構築システムを利用した**アンケート実施**

通信手順の開発方針

ブラインド署名を利用した2フェーズ通信手順.

- 第1フェーズ：投票所への入場券発行
  - 受講生ごとに異なる入場券を発行する.
- 第2フェーズ：回答の提出
  - 入場券とあわせてアンケートの回答を提出する.

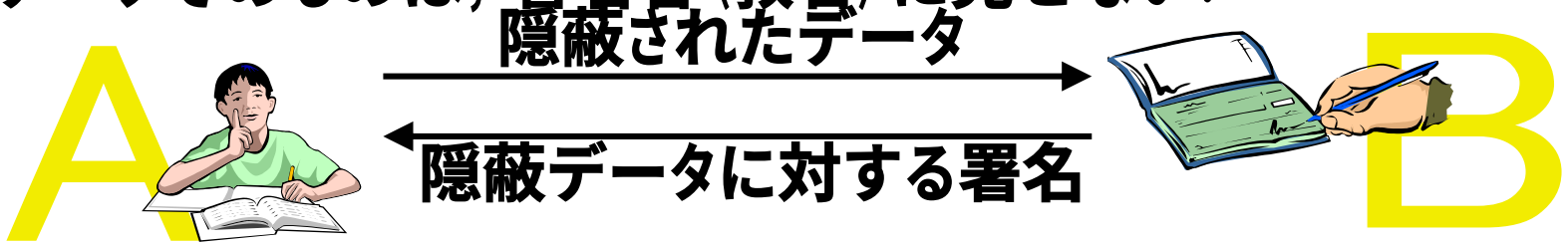


「誰がどの入場券を持っているか」は隠しておく必要あり.

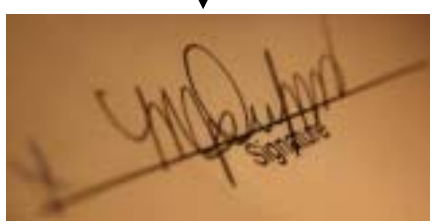
⇒ ブラインド署名の活用

ブラインド署名

- 特殊な電子署名方式.
- 自分が選んだデータに対し，第三者（たとえば教官）から電子署名をもらう.
- データそのものは，署名者（教官）に見せない.



隠蔽の解除



Bの署名

- 目隠しされた署名者に，署名を依頼するのと同じこと

構築したシステム

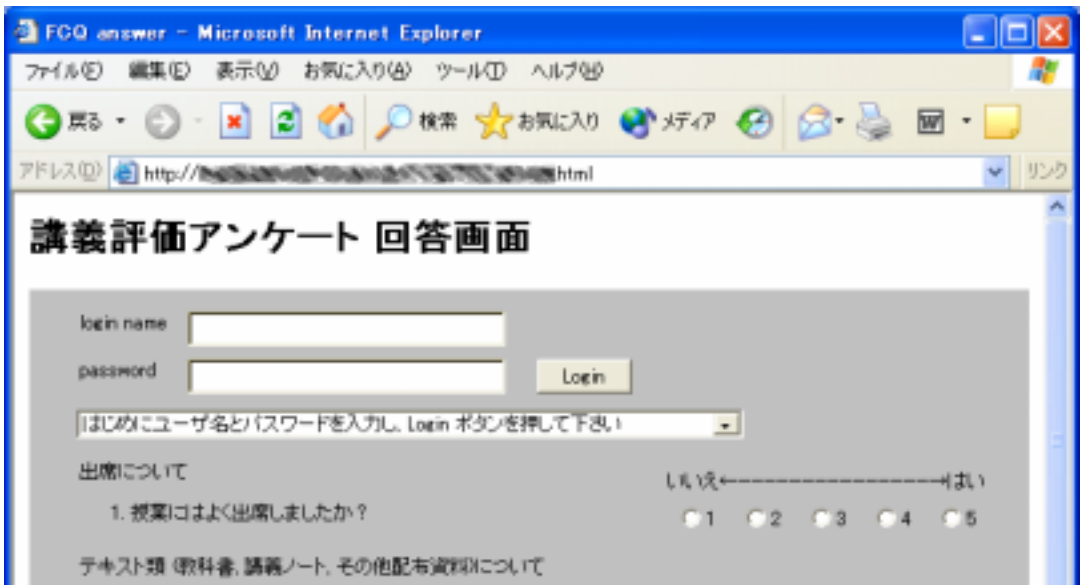
提案手順に従って動作するシステムを構築:

- 学生側クライアント: Javaアプレット
- 大学側サーバ: Javaサーブレット
- 1024ビットRSA型ブラインド署名, MD5ハッシュ関数

⇒ ソフトウェア要素だけで実現可能, WebベースのUI

本学における実証実験

- 第一回実験（02年3月）
  - 受講者11名の小規模な授業で試用
- 第二回実験（02年6月）
  - 20科目，165名の学生を対象に試用
  - のべ680件の回答を収集（回収率49.1%）



アンケート回答画面

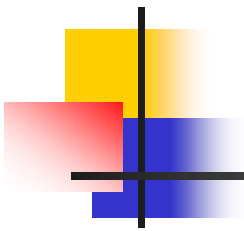
動的アクセス制御を行う Javaプログラムのセキュリティ検証

[1] N. Nitta, Y. Takata and H. Seki: An efficient security verification method for programs with stack inspection, **コンピュータソフトウェア**, Vol.19, No.3, 20-38, May 2002.

ソフトウェアは信頼できるか？

- 金融オンラインシステム
- ２０００年問題
- ロケット制御
  - １９６２年　Mariner 1 (金星探査機)　軌道を外れ，破棄．
    原因：プログラムにハイフン (-) が一つ抜けていた．
  - １９９６年　Ariane 5 打ち上げ失敗．

  原因：６４ビット浮動小数点数 -> １６ビット固定小数点数の変換で誤り発生．バックアップ計算機でも同一の誤り．誤った姿勢データが生成され，ロケットが破壊．



テストと検証

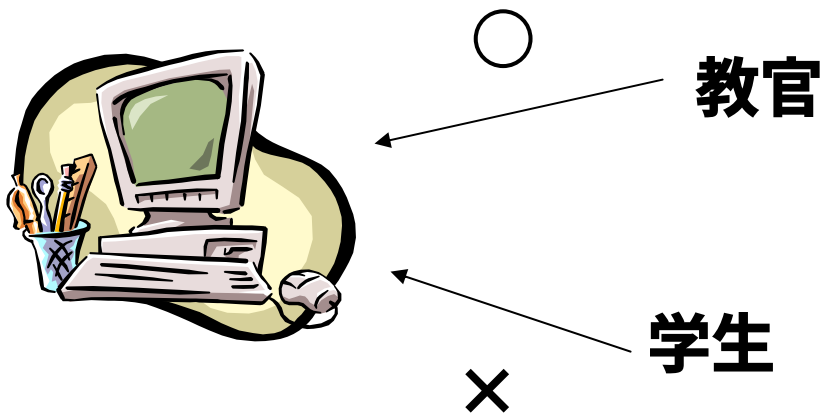
- テスト
  - 開発現場で普通にとられる方法.
  - プログラムを試しに実行してみて正しく動くか調べる．1億回のテストで正しく動いても，1億1回目に誤動作するかもしれない.
- 検証
  - 「プログラムがどんな状況にあろうとも，正しく動作する」かどうかを判定する技術.

検証 (Verification)

- **定理自動証明** (Automated Theorem Proving)
  - **数学の証明に似ている．専門家の助けが必要．**
  - **高度な性質を検証できる．**
- **モデル検査** (Model Checking)
  - **プログラム実行中に起こり得る状況 (到達可能な状態) を列挙して，望ましい性質が成り立つか調べる．**
  - **完全な自動化が可能 (人手不要)．**
  - **到達可能な状態集合が有限で，あまり大きくないことが必要．**
  - **NASA, NASDA, . . .**

アクセス制御 (Access Control)

- アクセス：人や計算機がデータを読み書きすること．
- 許可：「教官は学生の成績データを読んでよい」
  （**アクセス権**をもつ，ともいう）
- 禁止：「学生は自分以外の学生の成績データを読めない」



アクセス制御 (続き)

- パソコンをインターネットに接続,
  外部のプログラム (Javaアプレット) をダウンロード.
- 外部プログラムが悪意をもっていて,
  - パソコン内の重要データを破壊する,
  - 重要データを盗む,
  - そのパソコンを踏み台にして, 他のシステム (携帯電話の交換機システム, 政府のコンピュータ) を攻撃する

  可能性あり.
- それを防ぐための**アクセス制御**機構が提供されている.

アクセス制御機構 (イメージ)

本当にうまく機能するのか？

読み書きを指示

ダウンロードした外部プログラム

アクセス制御

アクセス制御

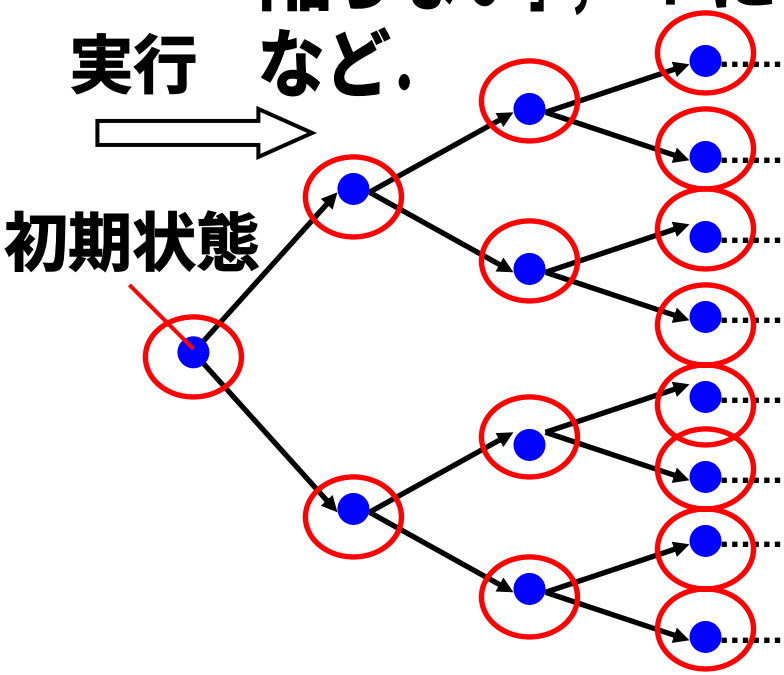
実際の読み書き

重要データ

ローカルメソッド (倉庫番)

モデル検査の原理

- プログラム　P
- 調べたい性質　ψ
  - 「Pはデッドロック（身動きが取れない）状態に陥らない」，「Pにはセキュリティホールはない」など．

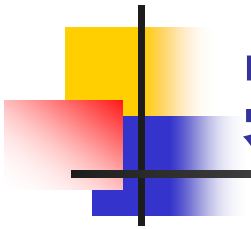


Ψを満たすか？

我々の検証法

- プログラムP　：フローグラフ
  調べたい性質ψ　：正規表現
  「アクセス制御はうまく働くか？」
- 得られた結果
  - 自動検証可能，DEXP-POLY困難[1]．
  - JDK1.2(Java Development kit 1.2) のアクセス制御に対しては，効率良く検証可能[2]．

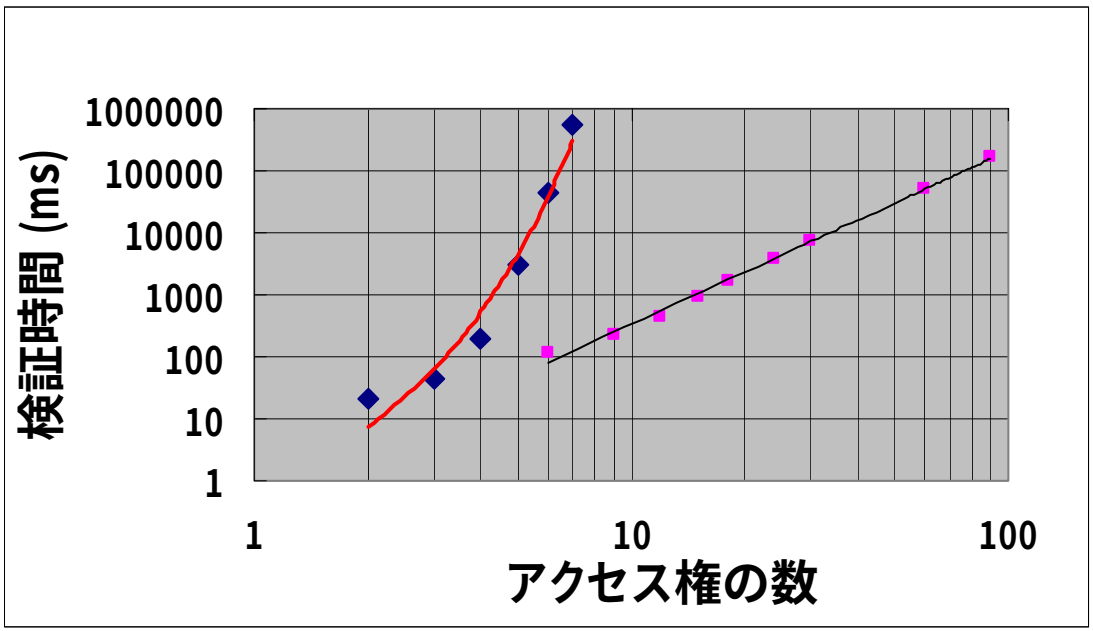
[1] “Security Verification of Programs with Stack Inspection,”
ACM SACMAT 2001.
[2] “An Efficient Security Verification Method for Programs with Stack Inspection,”
ACM CCS-8, May 2001.



実験

- 検証システムを試作.
- 実験1　(現実的な例題)
  - プログラム: k行の銀行を対象としたオンラインバンキングシステム(3×k個のアクセス権)
  - 調べる性質: 不正なアクセスが成功しない.
- 実験2　(最も検証時間がかかるケース)

実験結果



- 現実的なプログラムでは,
  - 検証時間はアクセス権の数に対して多項式オーダ.
  - アクセス権が90以下なら検証時間は3分以内 (使用計算機Alpha21164A　500MHz).

考察

- 1つのシステムにおいてアクセス権の数は多くても数10程度.
  - ドレスデン銀行のシステム[4]
    アクセス権の数 < 20

→本手法は実用規模のプログラムに適用可能.

[4]A.Schaad, J. Moffett and J.Jacob: The role-based access control system of a European Bank: A case study and discussion,
6th ACM Symp. on Access Control Models and Technologies, 3-9, 2001.

産学連携への展望