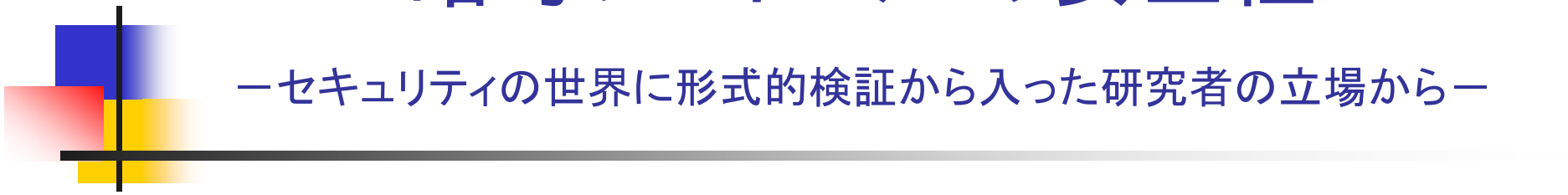
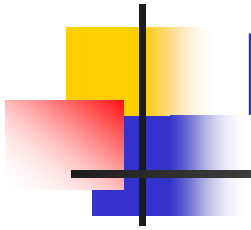


暗号プロトコルの安全性

ーセキュリティの世界に形式的検証から入った研究者の立場からー

藤原　融

大阪大学　大学院情報科学研究科



Dolev-Yaoの研究と嵩研の研究

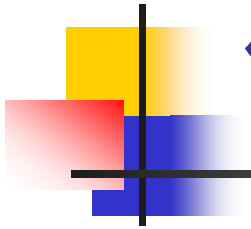
- Dolev-Yao Model（1引数Model）
  - 22nd IEEE FOCS, October 1981
  - Information & Control, December 1982 (Dolev, Even, Karp)
  - IEEE Trans. IT, March 1983
- 嵩研：具体例での検証: Kasami-Yamamura-Mori
  - 信学論, June 1983 (信学技報, May 1980)
- 嵩研：多引数 Model: Fujiwara-Taniguchi-Kasami
  - 信学論, June 1986 (信学技報, January 1983)

- Fujiwara
  - 1981.3 卒論 (具体例)
  - 1983.2 修論 (多引数Model)

Dolev-Yao Modelの対象

- Cascaded Protocol
  - 2者間で、通信がカスケードに行われる
  - 暗号化、復号だけが提供される操作
  - 使える性質：D(E(X))=X, E(D(X))=X
- Name-stamp Protocol
  - 暗号化、復号に加え、IDの付加、削除
  - $i_X(m)$： メッセージmに名前Xを付加
  - d(mX)：末尾からID (X) を削除
  - $d_X(mX)$：末尾がID Xかチェック

いずれも、操作（関数）は1引数



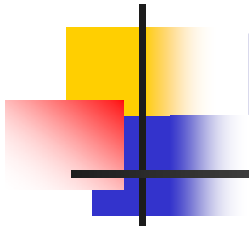
簡単なプロトコル例

- Cascaded Protocol
  - A send B the message (A, $E_B(M)$, B)
  - B answers A with the message (B, $E_A(M)$, A)
- Name-stamp Protocol
  - A send B the message (A, $E_B(MA)$, B)
  - B answers A with the message (B, $E_A(MB)$, A)

Dolev-Yao Model

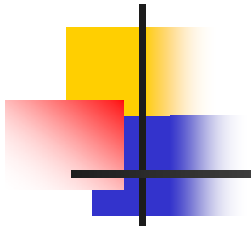
- プロトコルのフォーマルな記述法
- 秘密のメッセージを得ることができないかどうかで、安全性を定義

- Cascaded Protocol
  - A send B the message (A, E_B(M), B)
  - B answers A with the message (B, E_A(M), A)
- Name-stamp Protocol
  - A send B the message (A, E_B(MA), B)
  - B answers A with the message (B, E_A(MB), A)



Dolev-Yao 判定アルゴリズム

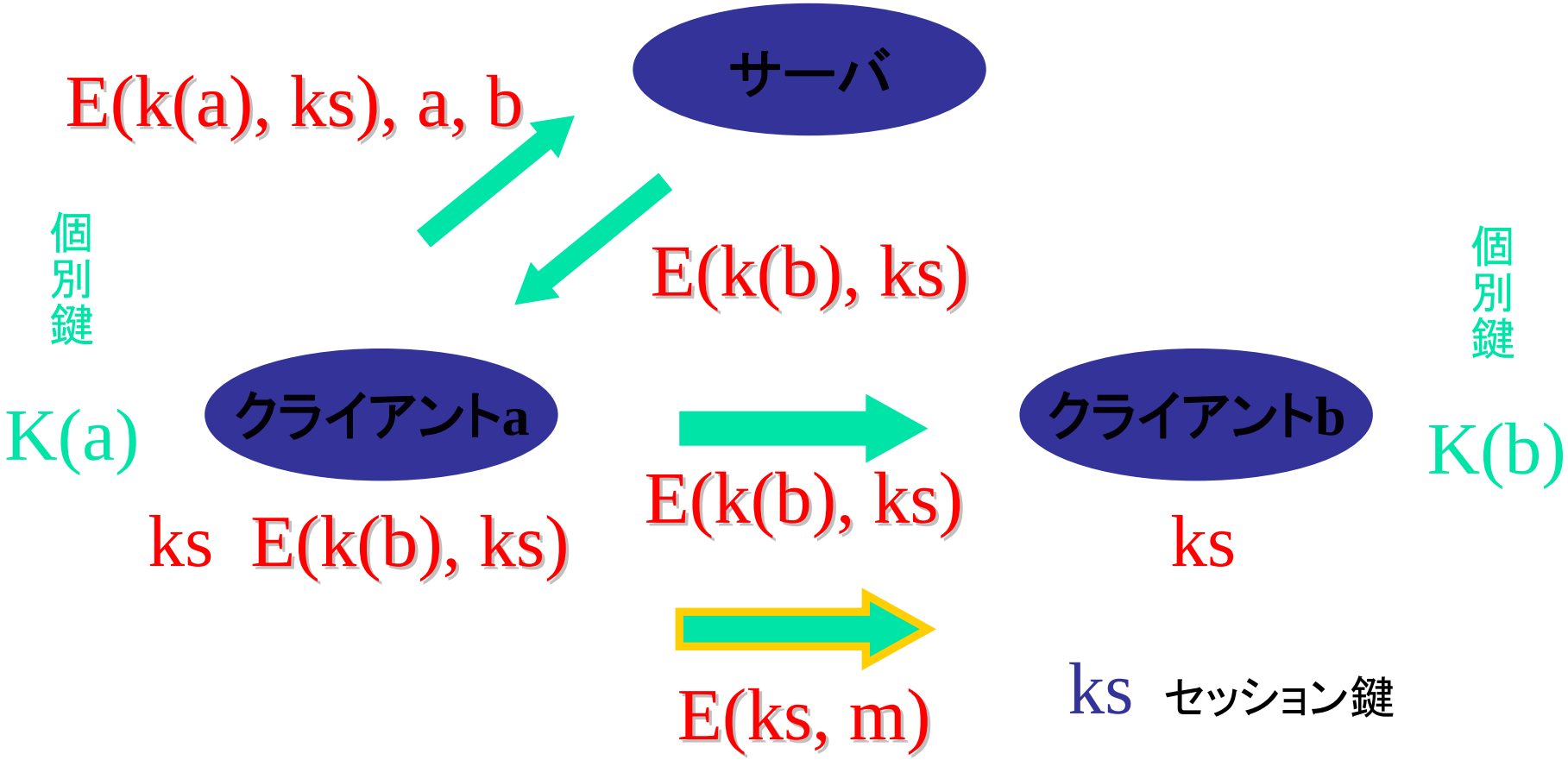
- 入力:プロトコルのフォーマルな記述
- 出力:安全か否か (Secure or Insecure)
- 正規言語の空問題に帰着

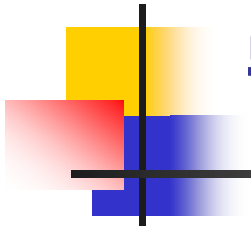


多引数モデルの必要性

- 鍵配送プロトコルでは、配送された鍵を使用いて、メッセージを暗号化
- 暗号化関数は2引数関数とすべき
  - D(X, E(X, Y)) = Y, E(X, D(X, Y)) = Y

鍵配送プロトコル

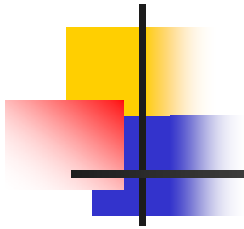

サーバ
E(k(a), ks), a, b
E(k(b), ks)
個別鍵
K(a)
クライアントa
ks
E(k(b), ks)
E(k(b), ks)
E(ks, m)
クライアントb
ks
個別鍵
K(b)
ks
セッション鍵



安全性の問題の記述

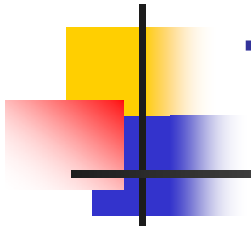
- 鍵配送プロトコルで提供されている操作は関数として，関数の意味や関数間の関係を公理として形式的に記述.
- 安全性も形式的に記述

  項書き換え系の単一化不可能性問題



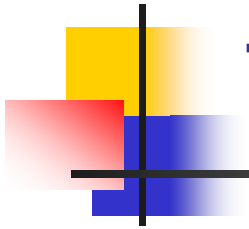
フォーマルな記述

- 攻撃者が得られる情報を表す項の集合を定義
  - もともと知っている情報
  - 盗聴により得られる情報
  - プロトコルの操作を行って得られる情報
  - 公理系を利用して、得られる情報
- 攻撃者が得られる情報のうちに、目的とする情報がなければ、安全



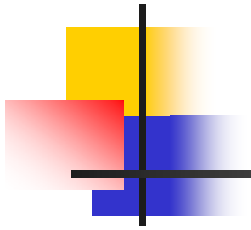
フォーマルな記述

- F: 関数の集合
- A: 公理系
- I:攻撃者が知りうる情報を表す項集合
- O:攻撃者が実行可能な関数
- G: 攻撃者のゴール



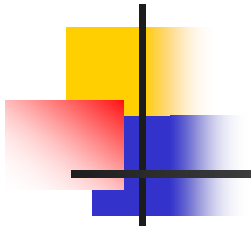
記述例

- 公理系A
  - D(X, E(X, Y)) = Y
  - E(X, D(X, Y)) = Y
  - Req(X,Y,Z) = E(K(Z), D(K(Y),X))
- 攻撃者が得られる情報
  - a, b, c, K(c), E(K(a), ks), E(K(b), ks), E(ks, m)
- 攻撃者が実行できる関数
  - E, D, Req



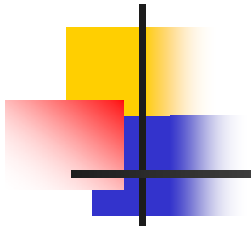
多引数:ゴール設定

- $g(t_1, t_2, \ldots)=g'(t'_1, t'_2, \ldots)$を満たす$t_1, t_2, \ldots, t'_1, t'_2, \ldots$を計算できるか?
  - t_1＝M　　$g(t_1, t_2, \ldots)=t_1$, $g'(t'_1, t'_2, \ldots)$=M
  - VerfySig(t_1)=true: 署名検証



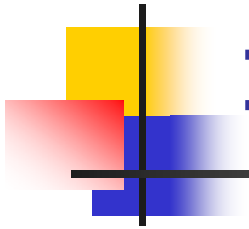
多引数：安全性判定

- 一般には決定不能
- 決定可能条件：項書き換え系として
  - 有限停止性をもち、標準形が一意に定まる
  - 操作を定義する公理は、右辺が線形
  - …
- 右線形を緩めると、決定不能（ポストの対応問題に帰着）



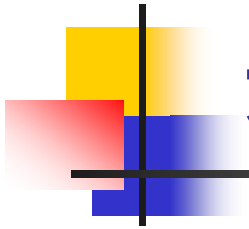
多引数:判定アルゴリズム

- 動的計画法 (dynamic programming)
  - 部分問題は、1引数の安全性問題
- 多項式時間アルゴリズム



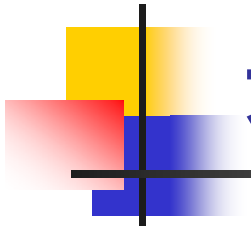
ゴールの設定

- Cascaded Protocol
  - A send B the message (A, $E_B(M)$, B)
  - B answers A with the message (B, $E_A(M)$, A)
- Dolev-Yao モデルでは、このプロトコルの安全性は、攻撃者がMを得られないこと
- もし、Mが得られなければ、それでよいのか？



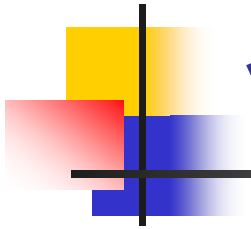
返信偽造不可能性問題

- Cascaded Protocol
  - A send B the message (A, E_B(M), B)
  - B answers A with the message (B, E_A(M), A)
- AはBに(A, E_B(M), B)を送らずに、(B, E_A(M), A)を得たい　⇒　これが返信の偽造



返信偽造不可能性問題

- 返事をもらうという関数は、攻撃者Aは利用してよい。特定の引数での利用が禁止
- 1引数: SCIS 88
- 多引数: ISEC 97
  - 解くのに10年かかったわけではない
  - その間は、検証法の実装など



今後の課題

- ゴールの設定に関する議論
  - プロトコルが与えられたとき、ゴールをどのように設定するか？
  - Needham-Schroeder Protocol への Loweの攻撃なども、設定が易しくはない例のように思われる。
- 本当に安全な暗号の構成