

BAN論理からProtocol Composition Logicへ (セキュリティ・プロトコルの論理的分析法)

長谷部 浩二

産総研システム検証研究センター

k-hasebe@aist.go.jp

岡田 光弘

慶応大学文学部哲学科

本講演の目的

- プロトコルの形式的安全性検証法の紹介
- 特に論理的分析法を紹介
- 近年の計算量的証明手法と形式的(論理的)手法との関係についての研究も(簡単に)紹介

本講演の概要

- プロトコルの形式的分析手法の概観
 - 歴史
 - 特徴による分類
- 論理的分析手法の紹介
 - BAN論理
 - Protocol Composition Logic (PCL)
 - Basic Protocol Logic (BPL)
 - (その他の論理的手法については論文の中で紹介予定)
- 計算量的分析手法との関係についての研究の紹介

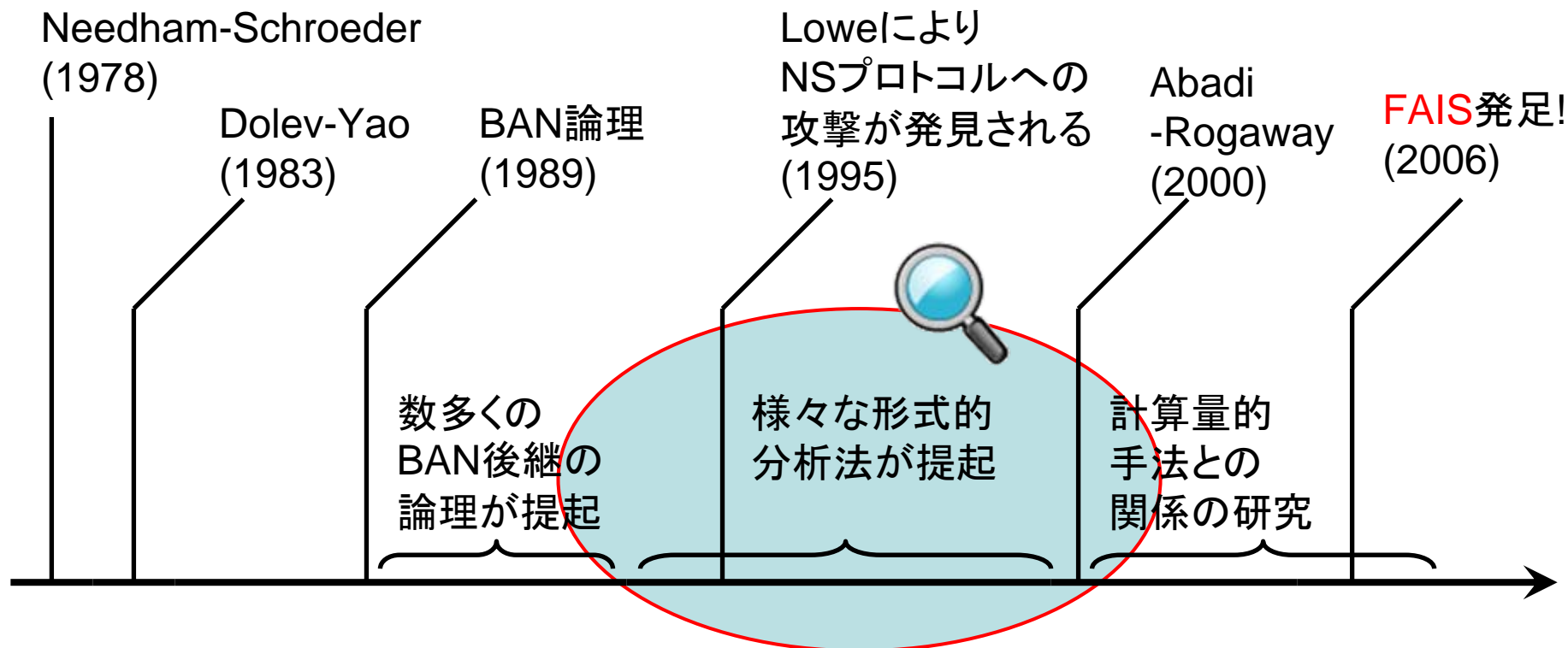
1. 形式的分析法の概観

形式的分析法の歴史(年表)



参考文献

- Clark-Jacob, A Survey of Authentication Protocol Literature, 1997
- Ryan-Schneider, Modelling and Analysis of Security Protocols, 1998
- Boyd-Mathuria, Protocols for Authentication and Key Establishment, 2003



形式的分析法の歴史



論理的アプローチ:

BAN論理

- AT論理
- GNY論理
- SvO論理

Inductive method

Protocol Composition Logic (PCL)

NRL protocol analyzer

Casper+FDR

Strand Spaces

Athena

MultiSet Rewriting (MSR)

意味論的アプロー

チ:

証明論 vs. 意味論

構文論 (syntax):

形式的推論体系

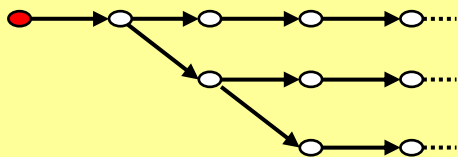
$$\frac{B \text{ believes } PK(A, K_A) \quad [A \rightarrow B : \{T_A, N_A\}_{K_A^{-1}}] \quad B \text{ received } \{T_A, N_A\}_{K_A^{-1}}}{B \text{ believes } A \text{ said } \langle T_A, N_A \rangle \quad B \text{ believes } \text{fresh}(T_A)} \\ B \text{ believes } A \text{ believes } N_B$$

論理的アプローチ:

Traces に関する abstract な推論を行う

意味論

(semantics):
Traces (プロトコルの実行プロセス) の集合



意味論的アプローチ:

効率的なアルゴリズムを用いて、全ての traces の中から攻撃のプロセスを探索

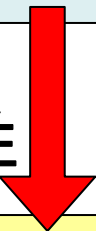
証明論 vs. 意味論

構文論 (syntax):

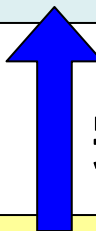
形式的推論体系

$$\frac{B \text{ believes } PK(A, K_A) \quad [A \rightarrow B : \{T_A, N_A\}_{K_A^{-1}}] \quad B \text{ received } \{T_A, N_A\}_{K_A^{-1}}}{B \text{ believes } A \text{ said } \langle T_A, N_A \rangle \quad B \text{ believes } \text{fresh}(T_A)} \quad B \text{ believes } A \text{ believes } N_B$$

健全性



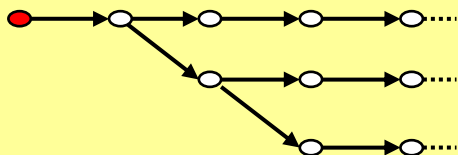
完全性



意味論

(semantics):

Traces (プロトコルの実行プロセス) の集合



両者の対応関係を示すことにより、Syntax での推論が traces の上でも成り立つことを保証

論理的アプローチと意味論的アプローチの特徴

- 論理的(証明論的)アプローチ (Theorem proving)
 - 証明が簡潔
 - 安全性の証明に利用
 - 証明の再利用が可能 (compositionality)
 - Intruder の行動を潜在的に扱う
- 意味論的アプローチ (Model checking)
 - 安全性の証明よりも、危険なプロセス(攻撃)の発見に向く
 - Intruder の行動を顕在的に扱う

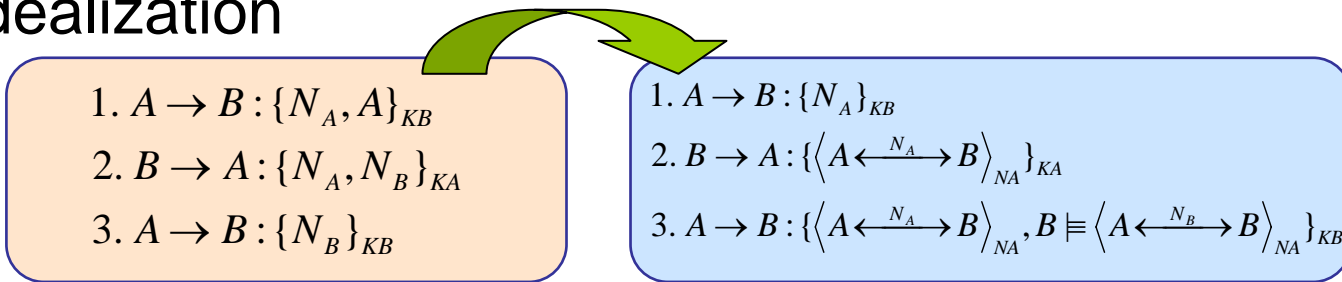
2. BAN論理の紹介

BAN論理

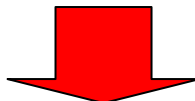
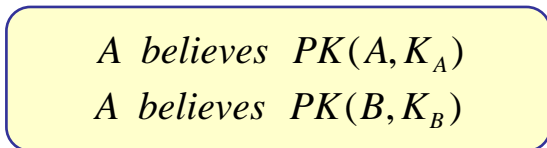
- Burrows-Abadi-Needham により提起
 - A Logic of Authentication (1989)
 - (Cf.) The Logic of Authentication Protocols (Syverson-Cervesato, 1999)
- プロトコル分析のための論理推論体系
- 信念 (belief) の概念を導入 (ただし「信念論理」ではない)
- 数多くの後継の論理に影響

BAN論理による分析の流れ

(1) Idealization



(2) プロトコルの仮定



(3) プロトコルの各ステップごとに推論

(プロトコルの仮定)

$[A \rightarrow B : \{N_A\}_{KB}]$

(実行後に成り立つこと)

論理推論

(結論1)

$[B \rightarrow A : \{\langle A \xleftarrow{N_A} B \rangle_{NA}\}_{KA}]$

(実行後に成り立つこと)

論理推論

(結論: プロトコル実行後に成り立つこと)

BAN論理の言語

- $P \models X$: P は X であることを信じている。
- $P \triangleleft X$: P は X を受け取った。
- $P \vdash X$: P はかつて X を送った。
- $\#(X)$: X は fresh である。
- $P \xleftrightarrow{K} Q$: 鍵 K は P と Q で share されている。
- $\vdash_K P$: 鍵 K は P の公開鍵である(秘密鍵は K^{-1} で表す)。
- $\{X\}_K$: X は鍵 K で暗号化されている。

読みにくいので.....

BAN論理の言語

- $P \text{ believes } X$: P は X であることを信じている。
- $P \text{ received } X$: P は X を受け取った。
- $P \text{ said } X$: P はかつて X を送った。
- $\text{fresh } X$: X は fresh である。
- $P \xleftrightarrow{K} Q$: 鍵 K は P と Q で share されている。
- $PK(P, K)$: 鍵 K は P の公開鍵である(秘密鍵は K^{-1} で表す)。
- $\{X\}_K$: X は鍵 K で暗号化されている。

※ ここで P, Q は名前であるが、 X はメッセージそのものである場合も論理式である場合もあり得る。

BAN論理の推論規則(例)

■ Message Meaning:

$$\frac{P \text{ believes } PK(Q, k) \quad P \text{ received } \{X\}_{K^{-1}}}{P \text{ believes } Q \text{ said } X}$$

(仮定1) k は Q の公開鍵である。
(仮定2) P は $\{X\}_{K^{-1}}$ を受け取った。
(結論) Q はかつて X を送った。

■ Nonce Verification:

$$\frac{P \text{ believes } \text{fresh}(X) \quad P \text{ believes } Q \text{ said } X}{P \text{ believes } Q \text{ believes } X}$$

(仮定1) X は fresh である。
(仮定2) Q はかつて X を送った。
(結論) Q は現在の session で X を送った。

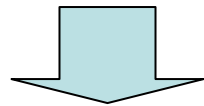
BAN論理による分析例

(CCITT X.509プロトコル, 1987)

(1) プロトコルのidealization

■ オリジナルのプロトコル:

1. $A \rightarrow B : \underline{A}, \{T_A, N_A, \underline{B}\}_{KA^{-1}}$
2. $B \rightarrow A : \underline{B}, \{T_B, N_B, \underline{A}, N_A\}_{KB^{-1}}$
3. $A \rightarrow B : \underline{A}, \{N_B\}_{KA^{-1}}$



Idealization

■ Idealize されたプロトコル:

1. $A \rightarrow B : \{T_A, N_A\}_{KA^{-1}}$
2. $B \rightarrow A : \{T_B, N_B, N_A\}_{KB^{-1}}$
3. $A \rightarrow B : \{N_B\}_{KA^{-1}}$

BAN論理による分析例

(CCITT X.509プロトコル, 1987)

(2) プロトコルの仮定を枚举

■ 暗号鍵についての仮定:

$A \text{ believes } PK(A, K_A)$

$B \text{ believes } PK(B, K_B)$

$A \text{ believes } PK(B, K_B)$

$B \text{ believes } PK(A, K_A)$

■ Freshness についての仮定:

$A \text{ believes } fresh(N_A)$

$A \text{ believes } fresh(N_B)$

$A \text{ believes } fresh(T_B)$

$A \text{ believes } fresh(T_A)$

BAN論理による分析例

(CCITT X.509プロトコル, 1987)

(3) 論理推論によりプロトコルに関して成り立つことを導出

K_A は A の公開鍵 (仮定)

B は最初のメッセージを受け取った。

[A → B : {T_A, N_A}_{KA⁻¹}]

<i>B believes</i> PK(A, K _A)	<i>B received</i> {T _A , N _A } _{KA⁻¹} MM	
<i>B believes</i> A said <T _A , N _A >	<i>B believes</i> fresh(T _A) NV	
<i>B believes</i> A believes N _B		

A はそのメッセージを送った

「B は A が現在のセッションで N_B を送ったと信じている」

[B → A : {T_B, N_B, N_A}_{KB⁻¹}]

<i>A believes</i> PK(B, K _B)	<i>A received</i> {T _B , N _B , N _A } _{KB⁻¹} MM	
<i>A believes</i> B said <T _B , N _B , N _A >	<i>A believes</i> fresh(N _A) NV	
<i>A believes</i> B believes N _A		

※ 論理結合子は ∧ (かつ) だけなので、推論は一本道。

BAN論理による分析例

(CCITT X.509プロトコル, 1987)

- 分析の結果:

1. Authentication の成立が形式的に証明出来た。

BAN論理による分析例

(CCITT X.509プロトコル, 1987)

■ 分析の結果:

1. Authentication の成立が形式的に証明出来た。
2. 「 T_B のチェックはオプションで良い」ことの正しさが説明出来る。

$$\begin{array}{l}
 [B \rightarrow A: \{T_B, N_B, N_A\}_{KB^{-1}}] \\
 \hline
 A \text{ believes } PK(B, K_B) \quad A \text{ received } \{T_B, N_B, N_A\}_{KB^{-1}} \quad \text{MM} \\
 \hline
 A \text{ believes } B \text{ said } \langle T_B, N_B, N_A \rangle \quad A \text{ believes } \text{fresh}(N_A) \quad \text{NV} \\
 \hline
 A \text{ believes } B \text{ believes } N_A
 \end{array}$$

N_A の代わりに T_B でも freshness をチェック出来る。

BAN論理による分析例

(CCITT X.509プロトコル, 1987)

■ 分析の結果:

1. Authentication の成立が形式的に証明出来た。
2. 「 T_B のチェックはオプションで良い」ことの正しさが説明出来る。
3. 「 T_A のチェックはオプションで良い」ことの誤りが説明出来る。

T_A の freshness を使わないと NV が適用出来ない。

$$\begin{array}{l}
 [A \rightarrow B : \{T_A, N_A\}_{K_A^{-1}}] \\
 \hline
 B \text{ believes } PK(A, K_A) \quad B \text{ received } \{T_A, N_A\}_{K_A^{-1}} \quad \text{MM} \\
 \hline
 B \text{ believes } A \text{ said } \langle T_A, N_A \rangle \quad B \text{ believes } \text{fresh}(T_A) \quad \text{NV} \\
 \hline
 B \text{ believes } A \text{ believes } N_B
 \end{array}$$

BAN論理の問題点

- Idealization の問題
 - 手続きが不明確
 - Idealize の過程で有用な情報が消去
- 表現力の弱さ(PCLとの比較において)
- 意味論の問題
- 信念の概念が本当に必要なのか？

BAN後継の論理

- AT 論理 (Abadi-Tuttle, 1990)
- GNY 論理 (Gong-Needham-Yahalom, 1990)
- SvO 論理 (Syverson-van Oorschot, 1994, 96)

3. PCLとその後継の論理

Protocol Composition Logic

- Durgin-Mitchell-Pavlovic らにより考案(1999)
- Datta-Derek-Mitchell-Pavlovic らにより発展・応用
(<http://www.stanford.edu/~danupam/logic-derivation.html>)
- Hoare 論理を基にした推論体系
- プロトコルの拡張・合成に沿って推論を行える。
- 健全(sound)な意味論
- Computational trace model

PCLの基本的なアイデア

- Hoare 論理の枠組を利用
- プロトコル=プログラムと見なす
- プロトコル記述とプロトコルの性質記述のための2種類の言語を導入

プロトコルの性質
(述語論理式にて記述)

$\varphi [\pi] \psi$

プロトコルの実行プロセス

直観的な意味:

「もし φ が成り立っていれば、
プロトコルの実行プロセス π の
終了後に必ず ψ が成り立つ」

PCLの基本的なアイデア

- この論理的枠組を用いると.....

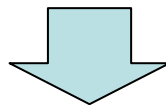
- プロトコルの拡張に関する推論が可能:

$$\frac{[New(n_1); New(n_2); Send\langle n_1, n_2 \rangle]_A Has(A, \langle n_1, n_2 \rangle)}{[New(n_1); New(n_2); Send\langle n_1, n_2 \rangle; Rec(\{n_2\}_{KA})]_A Has(A, \langle n_1, n_2 \rangle)}$$

- 性質に関する推論も可能:

$$\frac{[New(n_1); New(n_2); Send\langle n_1, n_2 \rangle]_A Has(A, \langle n_1, n_2 \rangle) \quad Has(A, \langle n_1, n_2 \rangle) \supset Has(A, n_1) \wedge Has(A, n_2)}{[New(n_1); New(n_2); Send\langle n_1, n_2 \rangle]_A Has(A, n_1) \wedge Has(A, n_2)}$$

$$[New(n_1); New(n_2); Send\langle n_1, n_2 \rangle]_A Has(A, n_1)$$



プロトコルに関する性質を、プロトコルの拡張に沿って証明出来る。

PCLの言語 (Formulas)

- Action formulas:

$$\alpha ::= \text{Send}(P, m) \mid \text{Receive}(P, m) \mid \text{New}(P, n) \\ \mid \text{Decrypt}(P, m) \mid \text{Verify}(P, m)$$

- Formulas:

$$\varphi ::= \alpha \mid \text{Has}(P, m) \mid \text{Fresh}(P, m) \\ \mid \text{Honest}(P) \mid \text{Contains}(m_1, m_2) \\ \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x\varphi \mid o\varphi \mid \diamond\varphi$$

Abbreviation:

$$\alpha_1 < \alpha_2 \equiv \diamond(\alpha_2 \wedge o\diamond\alpha_1)$$

- Modal formulas:

$$\varphi_1[S]_P \varphi_2$$

ただしここで S は $\alpha; \dots; \alpha$ (アクションの列)を表す。

PCLの公理と推論規則の例

- 公理の

例: $Has(P, \{m\}_K) \wedge Has(P, K^{-1}) \rightarrow Has(P, m)$

$Honest(X) \wedge Decrypt(Y, \{m\}_{KX}) \rightarrow X = Y$

- 推論規則の例:

$$\frac{\varphi[S]_P \phi \quad \phi[S']_P \theta}{\varphi[S;S']_P \theta} \text{Composition}$$

$$\frac{[\]_X \varphi \quad \varphi[S]_X \phi}{Honest(X) \rightarrow \phi} \text{Honesty}$$

ただしここで S は、Send で終わる
 X の role の initial segment か、
 あるいは X の role 全体

PCLの意味論と健全性

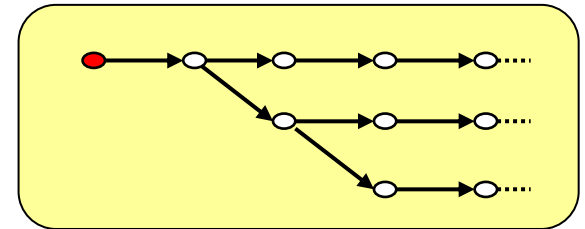
- PCLの意味論 (symbolic model):

Trace (可能なプロトコルの run) の集合

- 論理式が真:

(例) $[Send(m)]_X Has(X, m)$ が真

(どの trace においても、Send(X,t) が起こったら Has(X,t)が成り立つ。)



- PCLの健全性:

$$\vdash \varphi \Rightarrow \models \varphi$$

(PCLで証明可能ならば、全ての可能な traces において真)

Compositionの例

(Cf. DDMP, J. of Comp. Security, 13, 2005)

■ ISO-9798-3 プロトコル:

Diffie-Hellman鍵配送プロトコル

1. $A \rightarrow B : g^a \pmod{p}$
2. $B \rightarrow A : g^b \pmod{p}$

Challenge-response プロトコル

1. $A \rightarrow B : m, A$
2. $B \rightarrow A : n, \{m, n, A\}_{KB^{-1}}$
3. $A \rightarrow B : \{m, n, B\}_{KA^{-1}}$

composition

ISO-9798-3 プロトコル

1. $A \rightarrow B : g^a, A$
2. $B \rightarrow A : g^b, \{g^a, g^b, A\}_{KB^{-1}}$
3. $A \rightarrow B : \{g^a, g^b, B\}_{KA^{-1}}$

ISO-9798-3 プロトコルの secrecy 及び authentication を証明

Compositionの例

Diffie-Hellman鍵配送プロトコル

1. $A \rightarrow B: g^a \pmod{p}$
2. $B \rightarrow A: g^b \pmod{p}$

- g^a, g^b の freshness が PCL で証明出来る。
- g^a, g^b の secrecy を公理として導入。

Challenge-response プロトコル

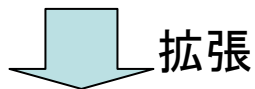
1. $A \rightarrow B: m, A$
2. $B \rightarrow A: n, \{m, n, A\}_{KB^{-1}}$
3. $A \rightarrow B: \{m, n, B\}_{KA^{-1}}$

- m, n の freshness を仮定すると、authentication が PCL で証明出来る。

Compositionの例

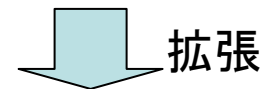
■ 証明のアイデア:

[DH] freshness of g^a, g^b



[DH'] freshness of g^a, g^b

freshness of g^a, g^b [CR] Auth



freshness of g^a, g^b [CR'] Auth

composition

プロトコルを拡張しても
freshness に影響は無い

m, n にそれぞれ g^a, g^b を代入

[DH' U CR'] Auth

DH' U CR' = ISO-9798-3

他のcompositionの応用例

- TSL
- IEEE802.11i
- Contract signing protocol
- Isabelle/HOL による実装

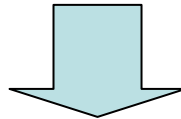
PCLの利点・問題点

- プロトコルの authentication、secrecy を証明するのに十分な言語
- Hoare 論理の枠組を用いることにより、証明の拡張・再利用による flexible なプロトコルの証明が可能
- 多くの property を体系の中に導入したことで、論理が複雑になっている。(Cf. Hasebe-Okada, ISSS'03、FCS'04)

PCLの単純化

(Cf. Hasebe-Okada, *BPL*, Rule'05)

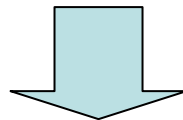
- もし authentication の証明に限定すると、論理的な性質はアクションだけで十分。



- Hoare 論理の枠組が不要となり、一階述語論理だけで十分に形式化出来る。

$$[S] \varphi \longrightarrow S \rightarrow \varphi$$

(S も φ もアクションの列を表す述語)

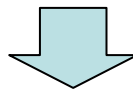


- 単純化により、健全かつ**完全**な意味論を持つ論理体系が得られる。(*Basic Protocol Logic* と呼ぶ。)

BPLの基本的なアイデア(1)

- 完全性定理: $\vdash \varphi \Rightarrow \vdash \varphi$

対偶: もしも Φ が BPL で証明出来なければ、 Φ に対する反例を構成することが出来る。



Φ として authentication を意味する論理式を取ると、 Φ の反例 (具体的な攻撃のプロセス) を構成するアルゴリズムとなる。

プロトコルの authentication を表す論理式

証明検索

(完全性定理)

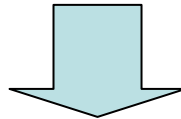
安全性の形式証明

反例モデルの導出

(具体的な攻撃プロセスの生成)

BPLの基本的なアイデア(2)

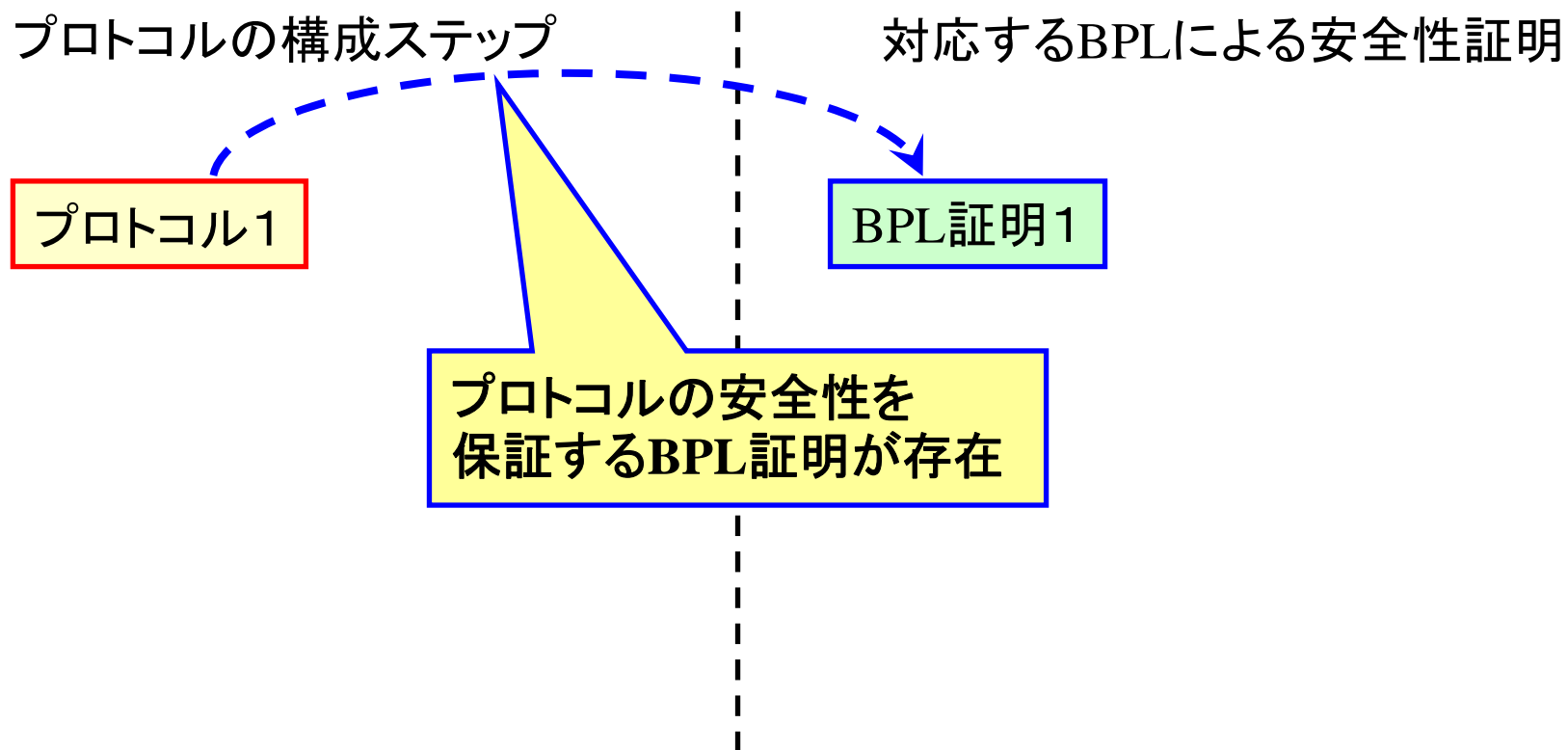
- Hoare 論理の枠組を排除したことで、composition の実現が難しくなる。



- プロトコルの authentication が保たれるような、プロトコルの拡張・合成規則を与える。

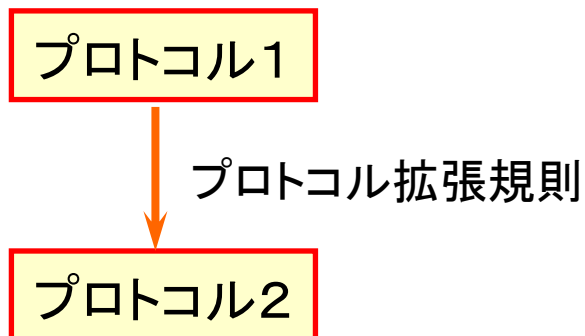
(cf. Cervesato-Meadows-Pavlovic, CSFW'05)

BPLの基本的なアイデア(2)



BPLの基本的なアイデア(2)

プロトコルの構成ステップ



対応するBPLによる安全性証明

BPL証明1

BPLの基本的なアイデア(2)

プロトコルの構成ステ

プロトコル1

1. $A \rightarrow B : \{N_A, A\}_{KB}$

2. $B \rightarrow A : \{N_A, B\}_{KA}$

安全性証明

プロトコル1

プロトコル拡張規則

プロトコル2

BPL証明

BPLの基本的なアイデア(2)

プロトコルの構成ステ

プロトコル2

1. $A \rightarrow B : \{N_A, A\}_{KB}$
2. $B \rightarrow A : \{N_A, N_B, B\}_{KA}$

安全性証明

プロトコル1

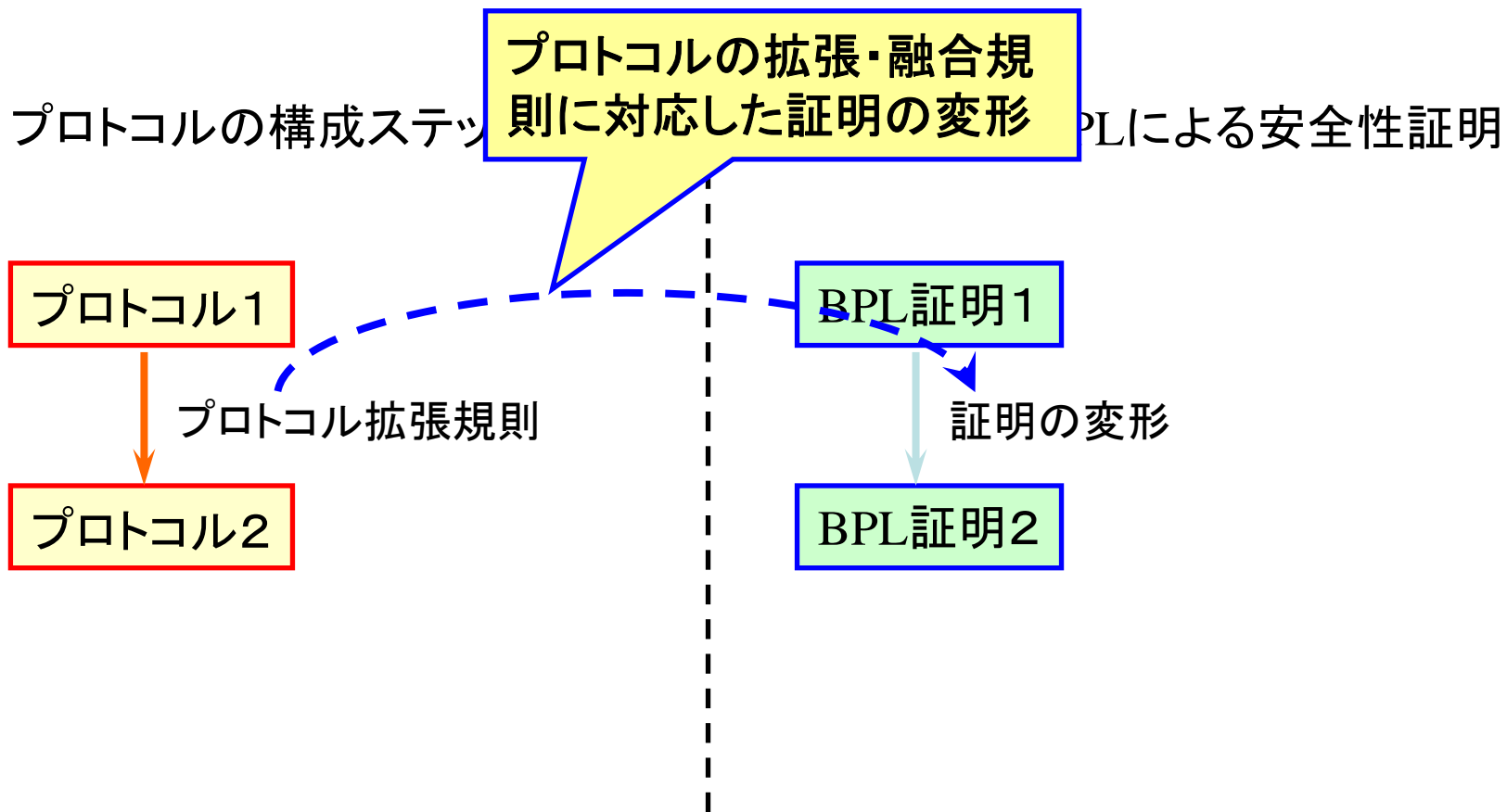


プロトコル拡張規則

プロトコル2

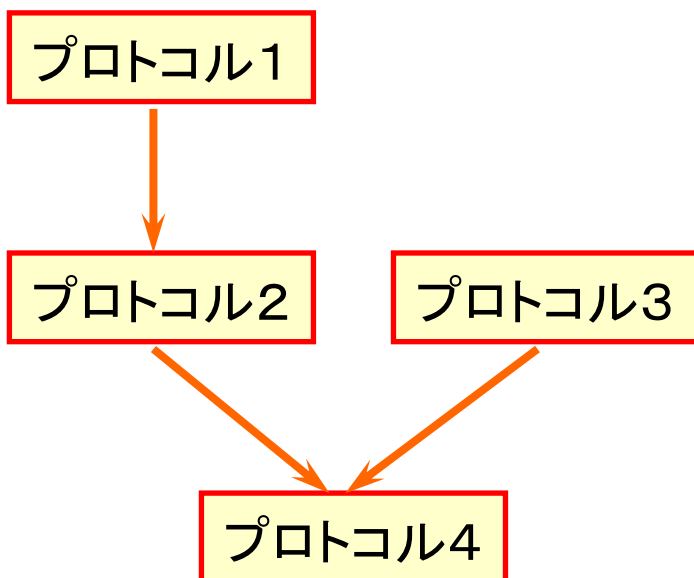
安全性証明

BPLの基本的なアイデア(2)

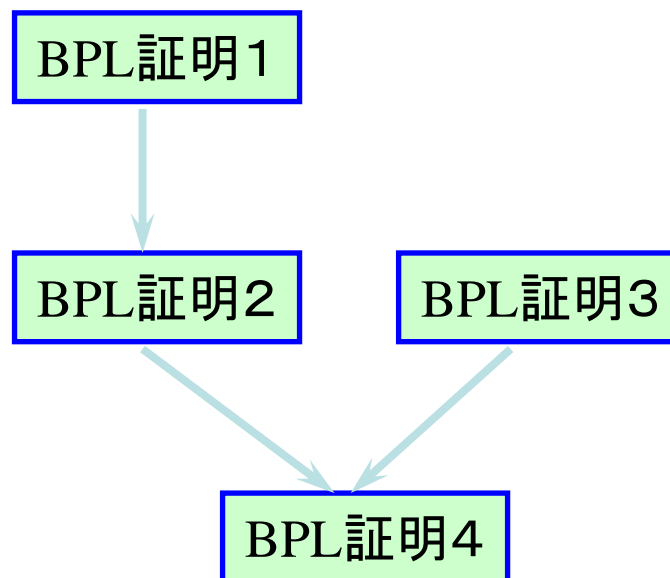


BPLの基本的なアイデア(2)

プロトコルの構成ステップ



対応するBPLによる安全性証明



プロトコルを拡張・融合しながら、BPLによって安全性が保証されたプロトコルを構成することが出来る。

BPLの主要な結果

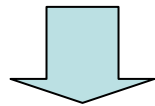
- 完全性定理
- 決定可能性 (セッションの数を制限した上で)
- 反例導出に利用
- Composition rules によるプロトコル生成

- Secrecyの分析 (cf. Meadows-Pavlovic)
- Computational semantics

4. PCLの最近の話題から

PCLの最近の話題

- Computational な意味論をPCLに与える。(間接的方法)



メリット:

- Symbolic な抽象度の高い推論体系で computational なモデルについての推論が可能
- プロトコルの composition も可能
- 異なる暗号スキームに対しても公理を調整することで対応

CPCPLの基本的なアイデア

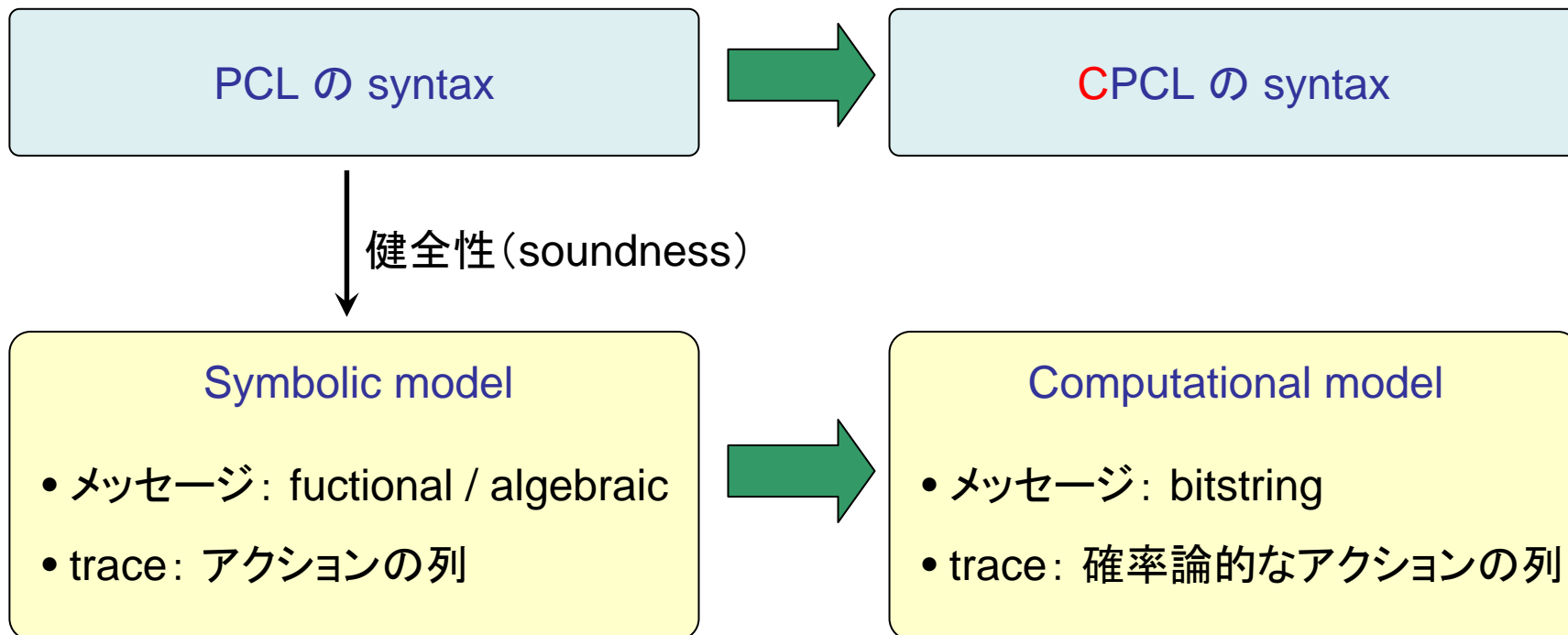
PCL の syntax

健全性 (soundness)

Symbolic model

- メッセージ: functional / algebraic
- trace: アクションの列

CPCPLの基本的なアイデア



CPCLの基本的なアイデア

若干の修正のみ
(基本的に変更しない)

PCL の syntax



CPCL の syntax

健全性 (soundness)

Symbolic model

- メッセージ: functional / algebraic
- trace: アクションの列

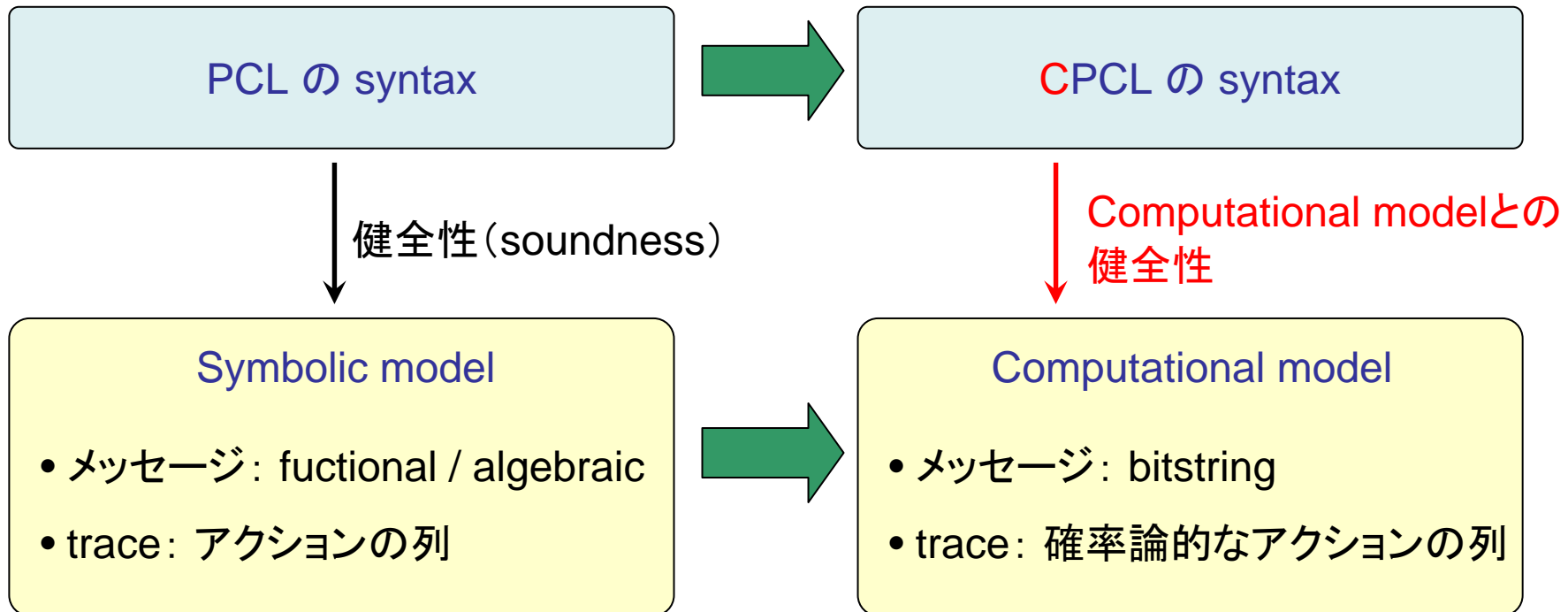


Computational model

- メッセージ: bitstring
- trace: 確率論的なアクションの列

メッセージをbitstringとして
モデル化

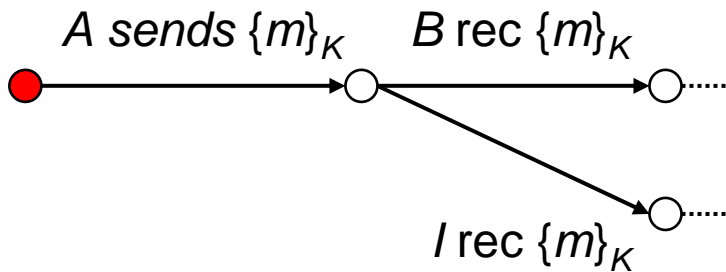
CPCPLの基本的なアイデア



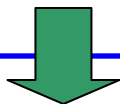
PCLのsyntaxによるabstract levelでの推論が、computational modelの上でも正しい推論になるようにする。

Symbolic trace から Computational trace への変更

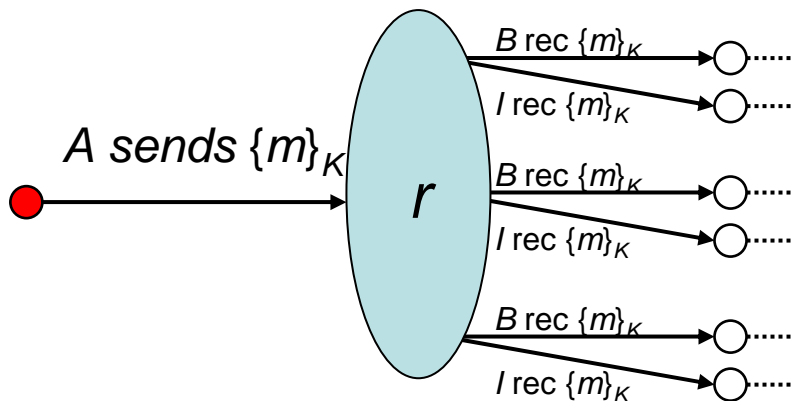
Symbolic traces:



メッセージ $\{m\}_K$ は functional で algebraic なものとして扱う。



Computational traces:



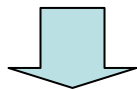
$\{m\}_K$ を bitstring として捉えると trace は確率論的な distribution となる。

CPCPLの言語（主要な変更点）

- $Has(X, m)$ の変更:

PCLではメッセージを functional なものとして形式化

(例) $Has(P, \{m\}_K) \wedge Has(P, K^{-1}) \rightarrow Has(P, m)$



- $Possess(X, m)$: 「X はメッセージ m を ppt で構成出来る」
- $Indist(X, m)$: 「X は、メッセージ m と他のランダムな bitstring との区別を ppt で出来ない」

- 論理結合子“ \Rightarrow ” (ならば) の変更:

$$\phi \Rightarrow \phi \quad (\text{条件付き確率})$$

Computational Semantics

- Computational trace model T :

$$T = T(Q, A, n)$$

- Q : protocol
- A : (ptcの能力を持つ)攻撃者
- n : security parameter

- 論理式 ϕ の意味(解釈) $[\![\phi]\!](T)$:
 ϕ が成り立つような traces の集合 ($\subseteq T$)

- 論理式 ϕ が真である $Q \models \phi$:

$$\frac{|\![\phi]\!(T, f(n))|}{|T|} > 1 - f(n)$$

十分大きな security parameterのもとで、negligible function f に対して高い確率で ϕ が起こる。

まとめ

- セキュリティ・プロトコルの論理的分析法の発展
 - BAN論理
 - PCL、BPL
 - 計算量的分析手法との関係の研究
 - (他のいくつかの手法については論文で紹介予定)