汎用的結合可能な安全性の形式的検証のためのpatternの拡張に関する一考察

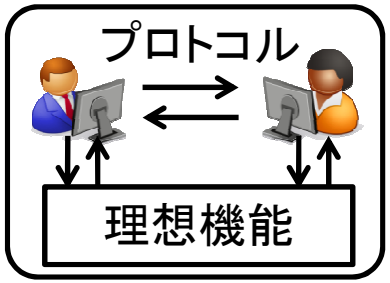
鈴木 斎輝，吉田 真紀，藤原 融
大阪大学 大学院情報科学研究科

背景

- [Canetti, 01] 汎用的結合可能な(UC)安全性の提案
  - プロトコルを組み合わせて実行しても損なわれない安全性
  - プロトコルが実現する暗号機能を他のプロトコルのサブルーチンとして利用
  - プロトコルが実現する暗号機能は理想的なインタフェースをもつとし、理想機能として記述

要求

さまざまな理想機能を利用する
プロトコルの安全性を検証を可能にすること



安全性検証のために考えること

1	2	3	4
理想機能の健全な記号的表現を定義	値の記号的表現(pattern)を定義	記号的安全性(symbolic criterion)を定義	検証法の設計

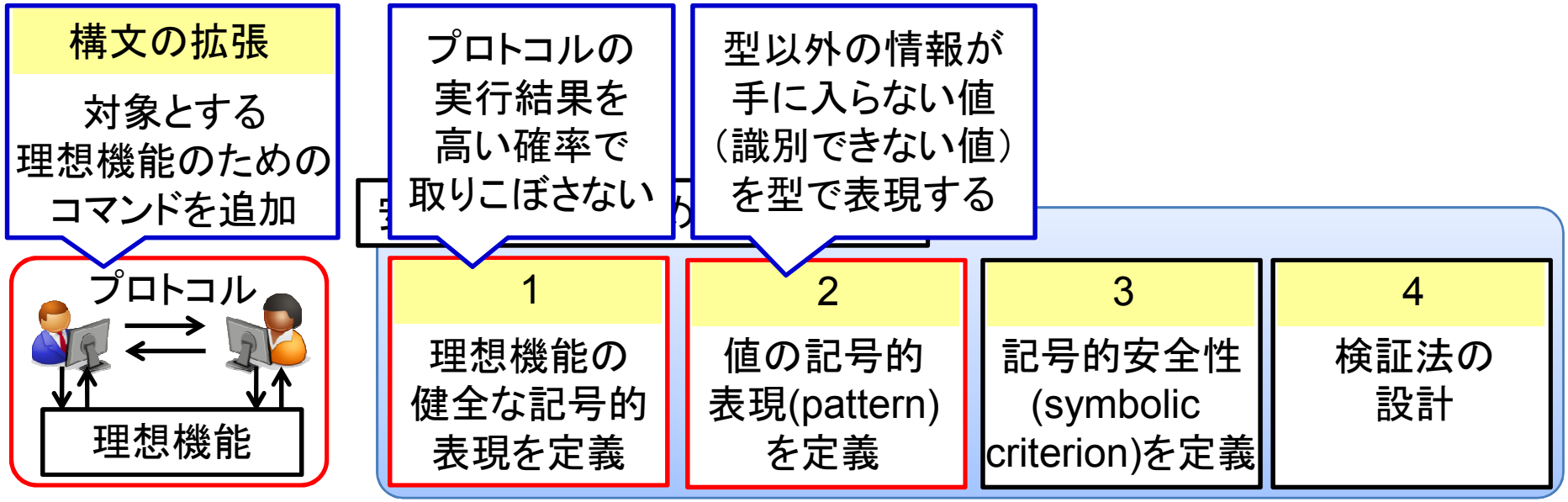
従来研究における形式的検証法

- 公開鍵型の理想機能を利用する相互認証・鍵交換プロトコル
  - 公開鍵型：理想機能が公開鍵と秘密鍵を生成し、パーティに割り当て、処理に用いる
  - [Canetti-Herzog, 04] 公開鍵暗号の理想機能
  - [Patil, 05] 公開鍵暗号とディジタル署名の理想機能
  - [村谷-花谷, 06] 任意の公開鍵型の理想機能

さらに、公開鍵型以外の有用な理想機能
（コミットメントや紛失通信）まで
拡張できれば、より望ましい

本発表の内容

- 公開鍵型以外の理想機能に対して、健全な記号的表現とpatternを定義
  - [Canetti, 01]で定義された理想機能のうち、通信路の理想機能を除く2者以下で実行する24種類（公開鍵型はこのうち5種類）の理想機能を対象



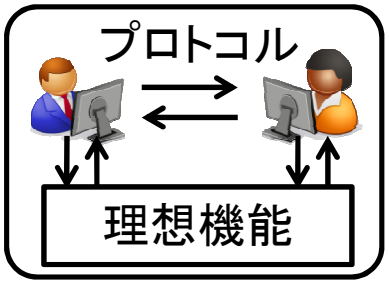
pattern の定義の比較

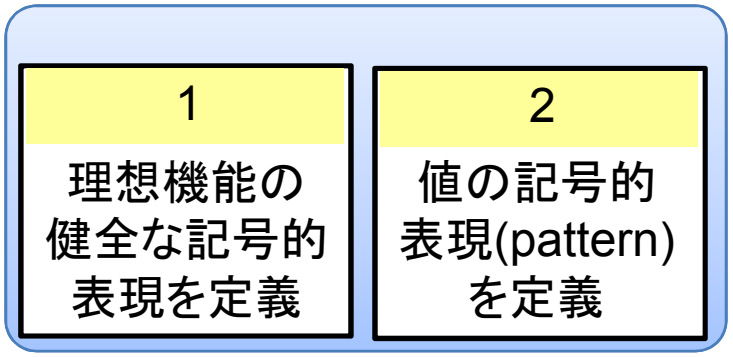
- 共通点
  - 定義の基本方針：型以外の情報が手に入らない（識別できない）値を表す項を型で表現
- 異なる点
  - どのような値を識別できない値とみなすか
  - [村谷-花谷, 06]：秘密鍵を用いて処理された値
  - 本発表：以下の2つの値
    - 理想機能によって秘匿される値
    - 理想機能がランダムに生成する値

これらの値を型で表現するようにpatternを定義

以降の発表の流れ

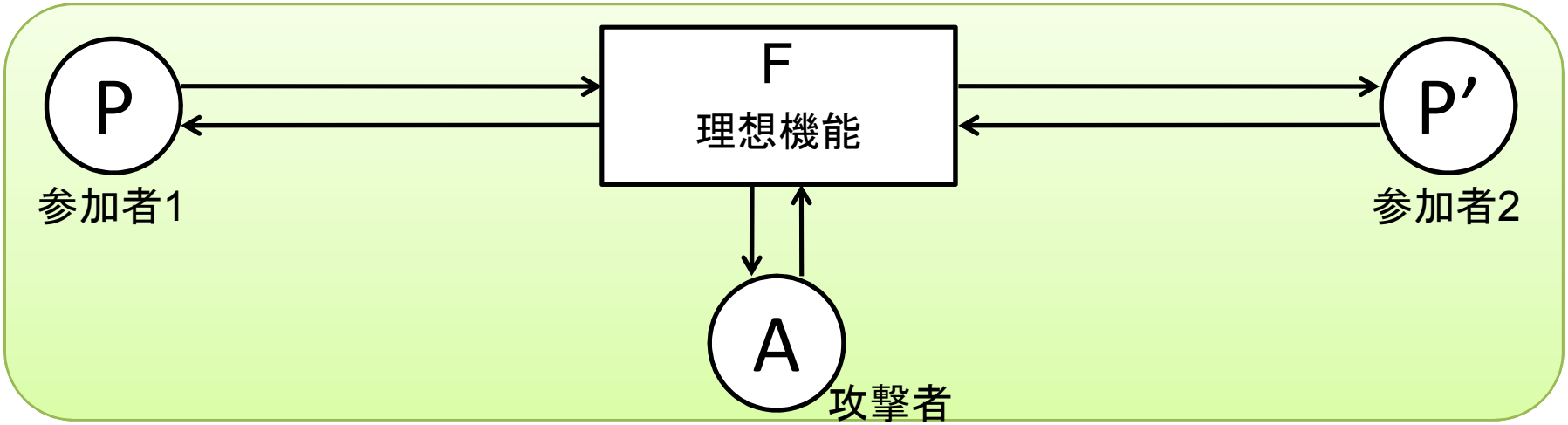
- 対象とする理想機能と例（コミットメント）
- 準備
- プロトコル構文の拡張
- 理想機能の記号的表現の定義
- 値の記号的表現（pattern）の定義





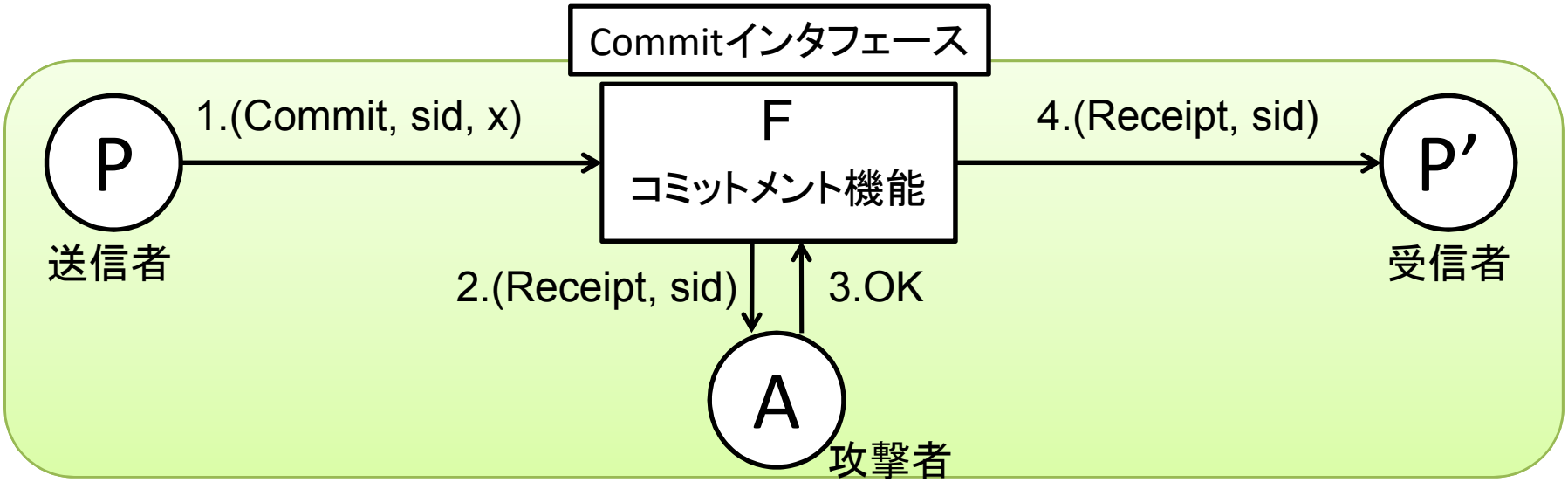
対象とする理想機能

- 初期化インタフェースでその理想機能内で用いる関数や、静的な値を定義
  - 公開鍵型の理想機能であれば、鍵の生成、割り当て
- パーティまたは攻撃者からの入力で動作
  - パーティからの各入力はクエリ名を含む
  - 攻撃者の入力は必ずしもクエリ名を含む必要はない



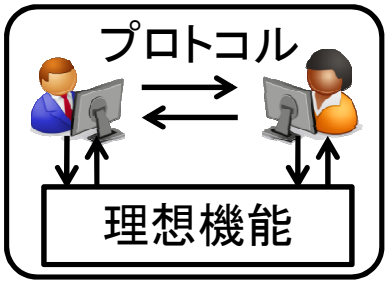
理想機能の例

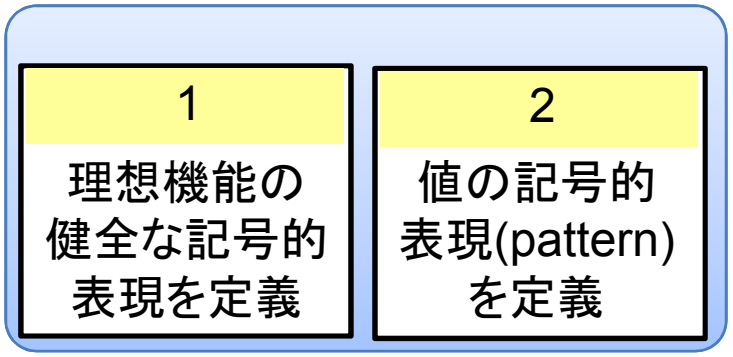
- コミットメント機能
  - 初期化インタフェース：なし
  - Commit インタフェース：送信者からCommitする値を受け取り、受信者へ値を受け取ったことを通知
  - Open インタフェース：送信者の開示の要求を受け、受信者へ値を送信
  - Corrupt インタフェース：送信者がコラプトされている場合に、攻撃者から値を受け取り、Commitしている値をその値に変更



以降の発表の流れ

- 対象とする理想機能と例
- 準備
- プロトコル構文の拡張
- 理想機能の記号的表現の定義
- 値の記号的表現(pattern)の定義



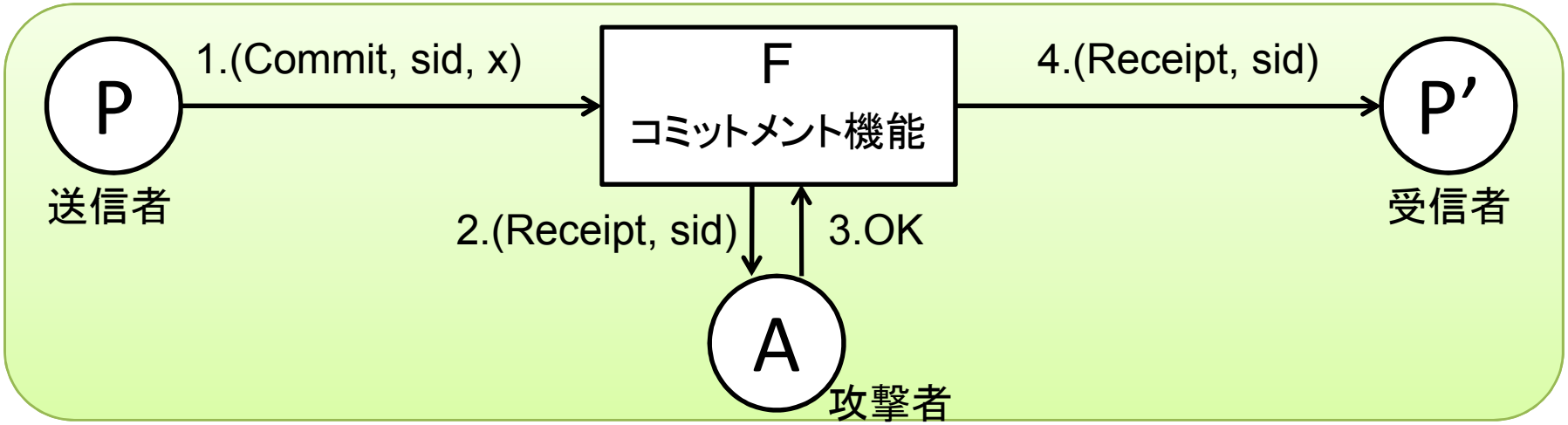


準備

- 理想機能の各インタフェースを分類・分割

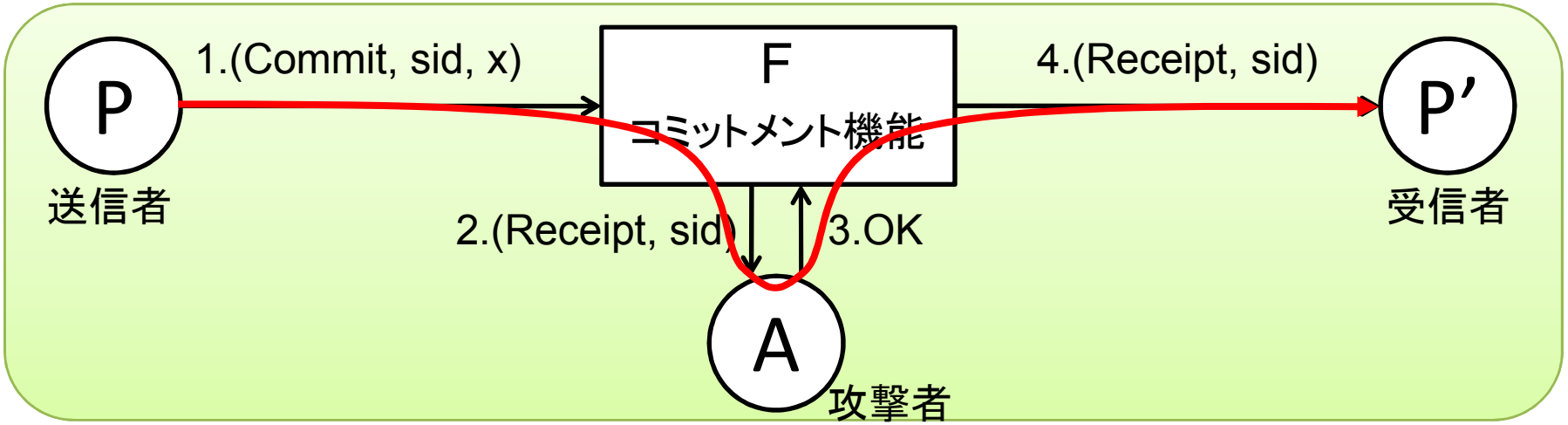
分割結果をもとに・・・

- プロトコル構文の拡張：コマンドの追加
- 理想機能の記号的表現：関数記号の用意



インタフェースの分類

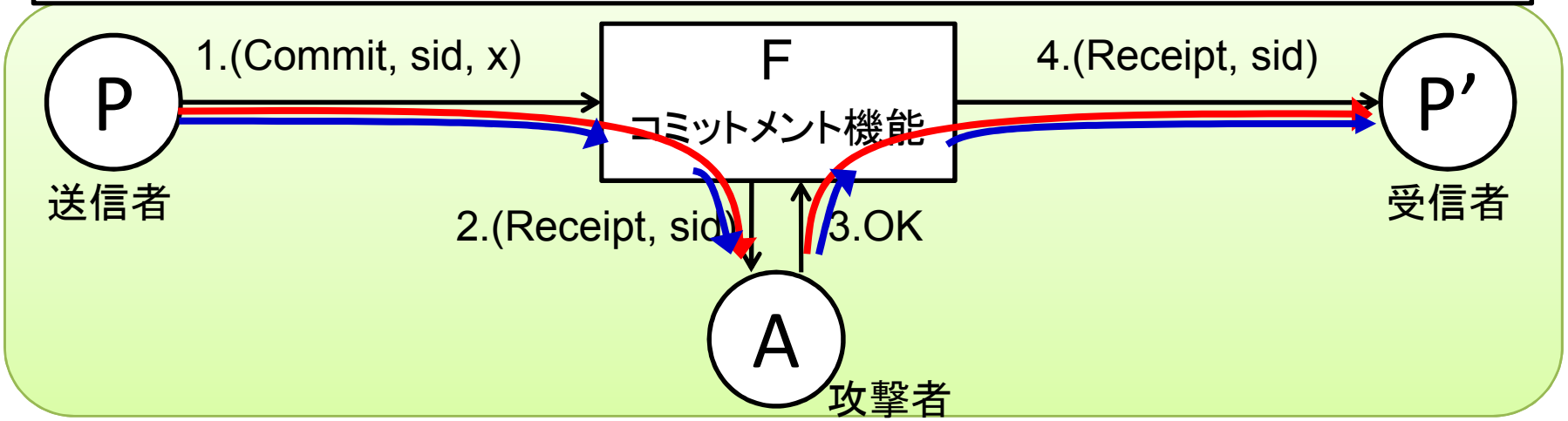
- 分類基準は[村谷-花谷, 06]と同じ
  - エンティティ(パーティ、攻撃者、理想機能)間のメッセージ授受の流れで分類
  - メッセージ授受の流れはエンティティの頭文字で表現
  - ex. Commit インタフェースはPFAFP’



インタフェースの分割

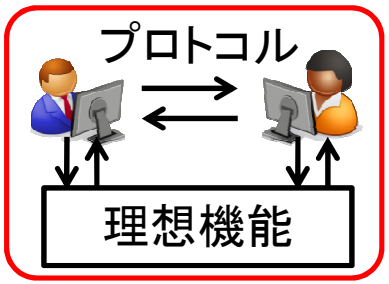
- 分割単位は[村谷-花谷, 06]と異なる
  - [村谷-花谷, 06]：理想機能への入力からそれに対する理想機能の出力までを1つのまとまりとして分割
    - ex. PFAFP'の場合は(PFA, AFP')と分割
  - 本発表：理想機能への入力と理想機能の出力を別々に分割
    - ex. PFAFP'の場合は(PF, FA, AF, FP')と分割

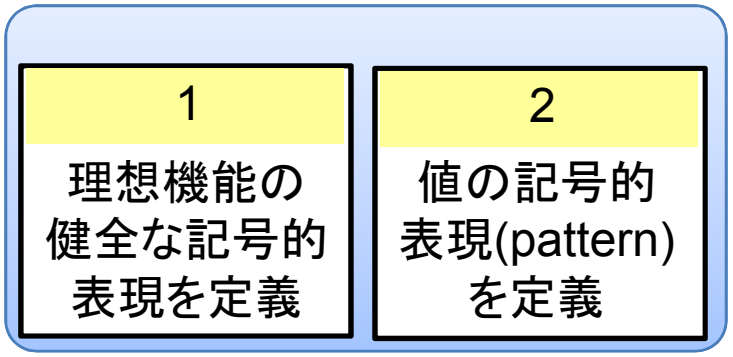
[村谷-花谷, 06]の分割で十分だが、コマンドや理想機能の記号的表現への対応付けを直観的にするために、一番細かく分割



以降の発表の流れ

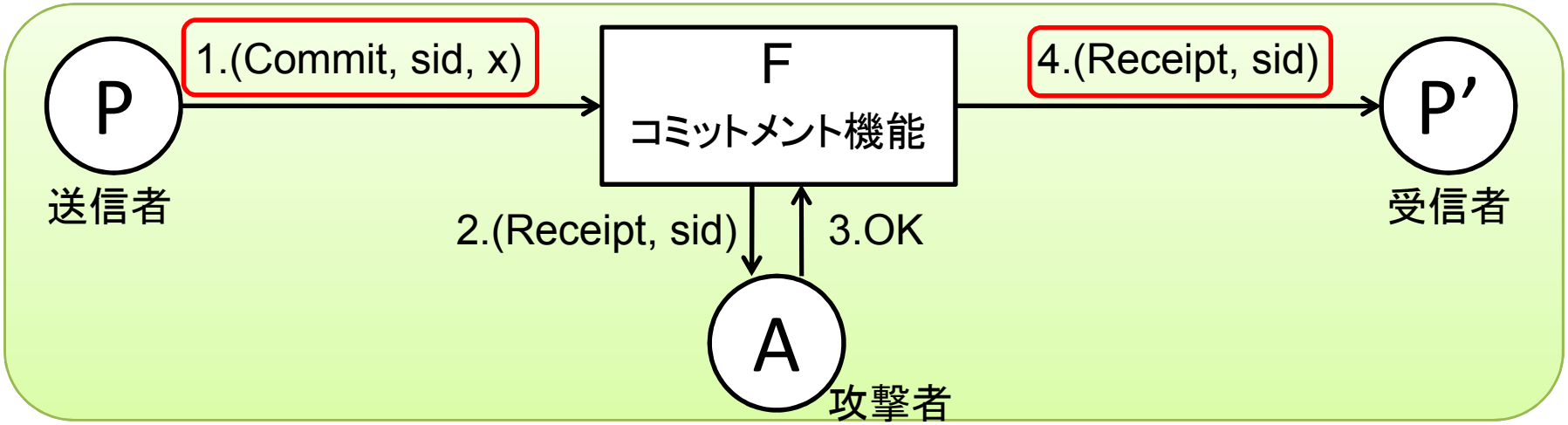
- 対象とする理想機能と例
- 準備
- プロトコル構文の拡張
- 理想機能の記号的表現の定義
- 値の記号的表現(pattern)の定義





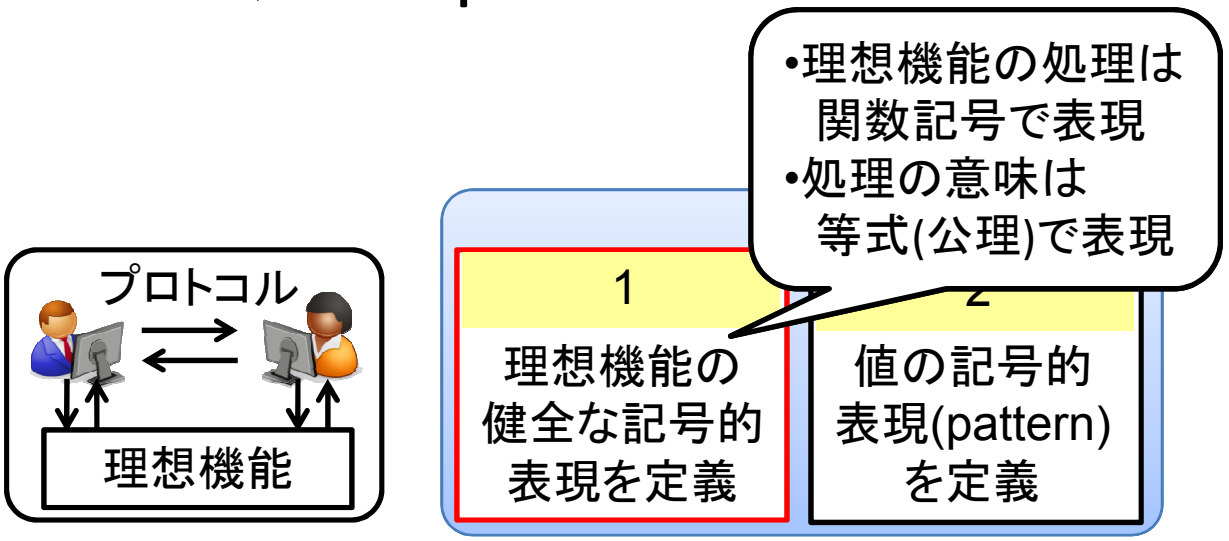
プロトコル構文の拡張

- パーティと理想機能との間のインタフェースPF, FP'に対するコマンドをプロトコルの構文へ追加
  - PFに対するコマンド：パーティが理想機能へ入力することを表す（commitコマンド）
  - FP’に対するコマンド：理想機能からの出力を受信することを表す（receiptコマンド）



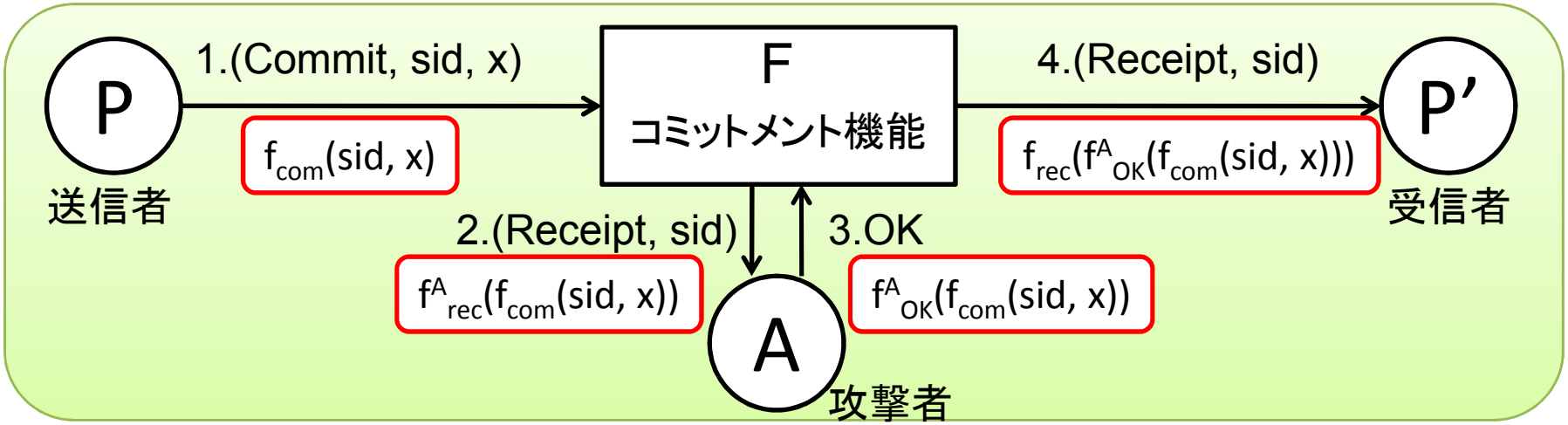
以降の発表の流れ

- 対象とする理想機能と例
- 準備
- プロトコル構文の拡張
- 理想機能の記号的表現の定義
- 値の記号的表現(pattern)の定義



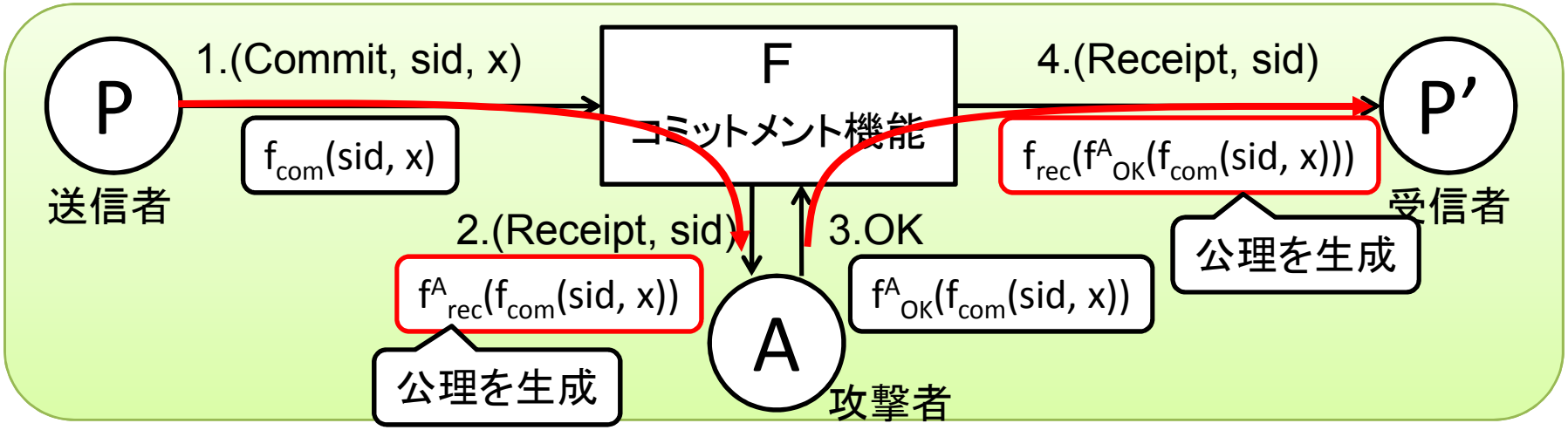
関数記号の定義

- 初期化インタフェースで定義される関数と静的な値に対応する関数記号を定義
- 分割結果に対応する関数記号を定義
  - 理想機能へ入力を与えるPF、AFを表す関数記号
    - 引数 = そこで入力する値
      ＋それまでの履歴（理想機能への入出力）
  - 理想機能から出力を受け取るFP’、FAを表す関数記号
    - 引数は理想機能への対応する入力（1つ）



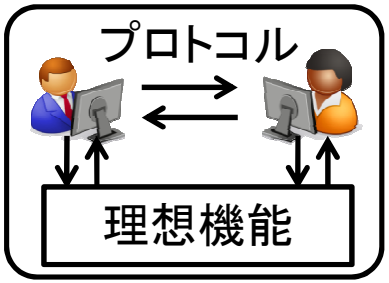
公理の定義

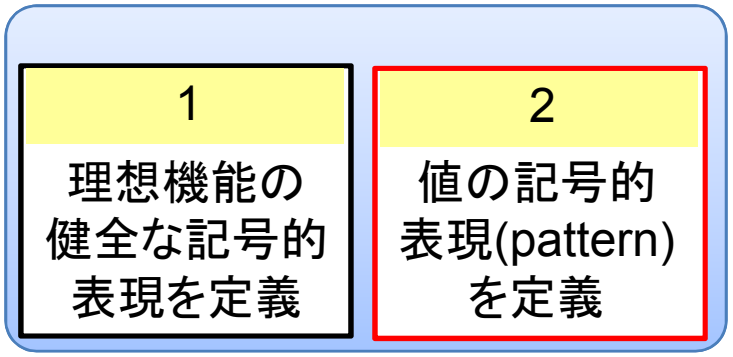
- 各インタフェースの各出力ごとに簡約公理を定義
  - コマンドや攻撃者の処理の流れがインタフェースに合っていれば，出力が得られることを表す
    - ex. $f^{A}_{rec}(f_{com}(sid, x)) == sid$
      $f_{rec}(f^{A}_{OK}(f_{com}(sid, x))) == sid$



以降の発表の流れ

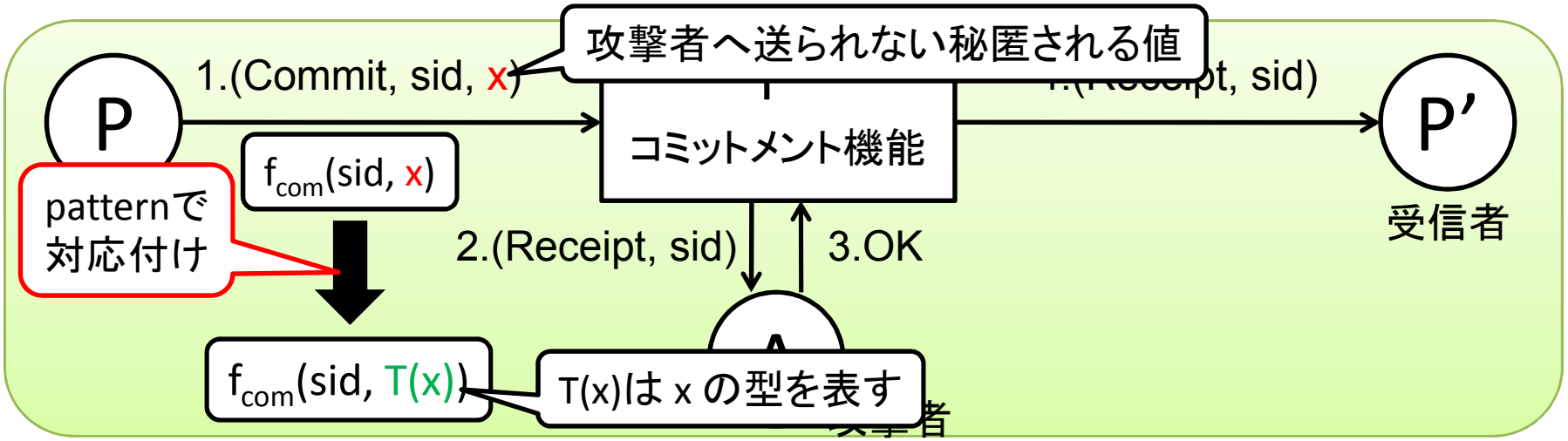
- 対象とする理想機能と例
- 準備
- プロトコル構文の拡張
- 理想機能の記号的表現の定義
- 値の記号的表現(pattern)の定義





pattern の定義

- pattern定義の基本方針：型以外の情報が手に入らない（識別できない）値を表す項を型表現へ対応付ける
- 本発表のpatternの定義：以下の値を型で表現
  - 理想機能によって秘匿される値
    - パーティの入力のうち、攻撃者へ出力されない値
    - ex. 送信者がコミットする値
  - 理想機能がランダムに生成する値
    - 初期化インタフェースで定義した関数でランダムに生成される値



まとめと今後の課題

- まとめ
  - [Canetti, 01]で定義された理想機能のうち、通信路の理想機能を除く2者以下で実行する理想機能を対象
    - プロトコル構文の拡張
    - 理想機能の健全な記号的表現を提案
    - 提案した記号的表現に対するpatternを定義
- 今後の課題
  - 記号的安全性の定義
  - 検証法の設計