計算論的に完全な記号的攻撃者と健全なプロトコル検証法

バナ・ゲルゲイ

ウベール・コモン-ルンド

櫻田英樹

発表内容

- 記号的な定式化と計算論的な定式化
- 従来の健全性定理の弱点
- 我々の提案：完全な記号的攻撃者の手法と健全性
- 我々の手法でNSLプロトコルの検証

計算論的、記号的な定式化

- 計算論的な定式化
  - 現代の暗号プリミティブは確率的なアルゴリズムに基づく。
  - 暗号プロトコルの実行：確率過程。
  - 安全なプロトコル：攻撃者が成功する確率は十分低い
  - 安全性は限定された計算能力に依存する (例えば、Diffie-Hellman仮定)
- 記号的な定式化
  - 暗号プロトコルの自動検証の為
  - プロトコルの動作を抽象化して参加者はビット列の代わりに項を操作する
  - 暗号プロトコルの実行：非決定プロセス
  - 確率、計算能力の限定を含まない

計算論的定式化

- 送信されるメッセージはビット列。
- 確率は低いか高いか、セキュリティ・パラメーターというサイズ・パラメターに対する関数として定式化される。
- セキュリティ・パラメータは暗号化に用いられる鍵の長さと考えてよい
- 無視できる関数 f(n): 任意の多項式p(n)について、nが十分大きくなるに連れて1/p(n)よりf(n)の方が0に速く近づく。すなわち、$\lim_{n\to\infty} f(n)p(n) = 0$
- 「攻撃者は成功できない」という事は、成功の確率がセキュリティ・パラメーターに対して無視できる関数である事で定義される。
- 攻撃者と参加者が用いるアルゴリズムは全て確率多項式時間アルゴリズムとする。(計算能力が限定される事)

記号的な定式化

- 送信されるメッセージは項。例： $\{N\}_B$
- 確率論的・計算量理論的な側面を無視する。
- 攻撃者に許される動作を全て列挙する。例えば、攻撃者は鍵Kおよび暗号文$\{x\}_K$を持っていると暗号文を復号して平文xを計算できる。
- 列挙されていない動作はできない。
- 検証ツールで全ての記号的実行を生成して成功する攻撃を探す。

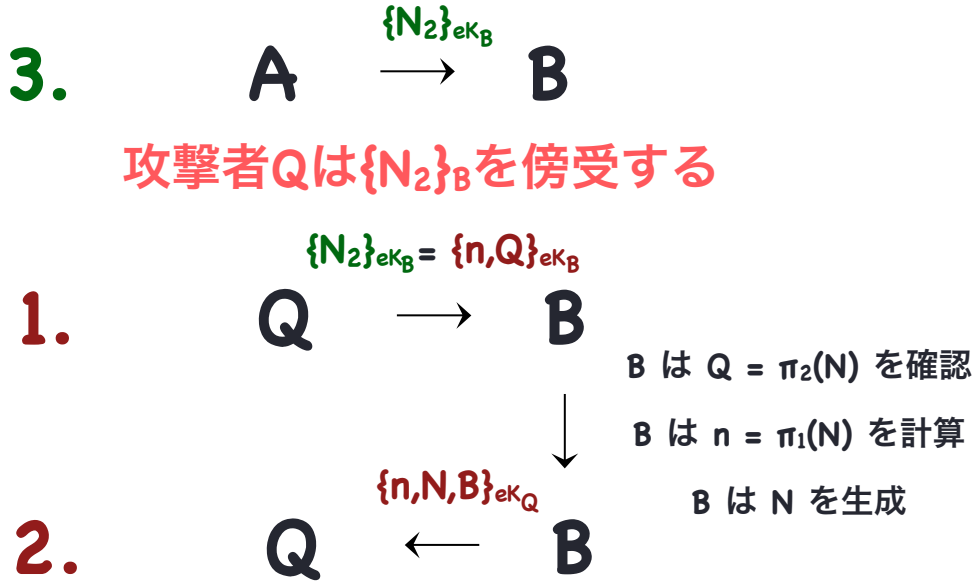
理想化された健全性定理

- **記号的な定式化の計算論的健全性**
  - **成功する記号的な攻撃が存在しないならば成功する計算論的な攻撃も存在しない。**
- **理想化**
  - **健全性定理は多くの文献で証明されている。**
  - **今までの文献の健全性定理は全て何らかの標準的でない仮定を用いる。**
  - **理由：記号的な攻撃者の動作を全て列挙しなくてはならない。計算論的な攻撃者は記号的な動作に対応しない動作をしない為に仮定が必要。**
  - **すなわち、計算論的な定式化の標準的な仮定を抽象化して記号的な定式化を構成できないから、計算論的な仮定を記号的な定式化に合わせる必要がある。**
  - **暗号プロトコルの実装について非標準的な仮定の例：鍵のビット列の半分を暗号化で使って、残りの半分を暗号文と連結する。**

NSLプロトコルへの攻撃

記号的に検証されているが、様々な計算論的攻撃が可能。

- **NSL:** $1.\ A \to B:\ \{N_1, A\}_{eK_B}$ $2.\ B \to A:\ \{N_1, N_2, B\}_{eK_A}$ $3.\ A \to B:\ \{N_2\}_{eK_B}$
- **〈n,Q〉 = Nと二つの並列のセッションがあるとする**
  - ある名前Qに対して、任意の正規に生成されたNについて$\pi_2(N)=Q$の確率は無視できないとする。



- **Qは正規の参加者でない**

- **このような攻撃を記号的手法で見つける為には、Dolev-Yao手法では、このような可能性を全て列挙せねばならない。しかし、どのようにすれば良いか分からない。**

アプローチ

- 1. 記号的な実行の代わりに計算論的モデルを直接用いて検証を行う方法（記号的な攻撃者を用いない）
  - Cryptoverif - ゲーム変換に基づく検証ツール
  - 論理的なアプローチ - 推論体系と計算論的な意味論を定義してセキュリティ要求を計算論的に妥当な公理から導く
- 2. 既に記号的な攻撃者の手法を用いる多くの研究の蓄積があるが、以上のアプローチでは使えない。
- 本研究
  - 2.の蓄積を利用可能にするため、1.とは異なる方法を提示する
  - 記号的な攻撃者の手法を修正して、強い健全性を得る新しい方法を提案する

アイデア

- 記号的攻撃者に許される動作を列挙する代わりに、記号的攻撃者が違反できないルール（公理）を列挙する。このルール以外、どのような動作をも可能であるとする。
- このような公理は暗号プリミティブと確率多項式時間アルゴリズムについての標準的な計算論的仮定から導かれる。
- たとえば暗号スキームの IND-CCA 安全性に対応するルールがある。
  - 「攻撃者は鍵および暗号文を持っていると平文を計算できる」代わりに
  - 鍵についての情報を全く持っていないと暗号文を持っていても平文を計算できない。
- このような記号的攻撃者は計算論的な攻撃者ができる動作を全てできるため、**計算論的に完全**な記号的な攻撃者である。

プロトコルの実行

- 参加者がお互いに送信するメッセージは全て攻撃者を経由する
- ラウンド：
  - 参加者は攻撃者からメッセージを受け取る
  - 参加者は受け取ったメッセージに関して様々な条件をチェックする。
    - 計算論的：受け取ったビット列は条件を充足すれば実行を続ける
    - 記号的：受け取ったメッセージをハンドルhで表して、チェックをhについての条件として記録する
  - 参加者はメッセージを返信するとともにフレームというデータに追加する
    - 計算論的実行の場合：ビット列の列。
    - 記号的実行の場合：項の列

計算論的なラウンド

1. $A \to B : \{N_1, A\}_{eK_B}$　　2. $B \to A : \{N_1, N_2, B\}_{eK_A}$　　3. $A \to B : \{N_2\}_{eK_B}$

攻撃者

ビット列bを確率多項式時間アルゴリズムで過去の実行から計算する

ビット列　b

ビット列　b′

正規参加者

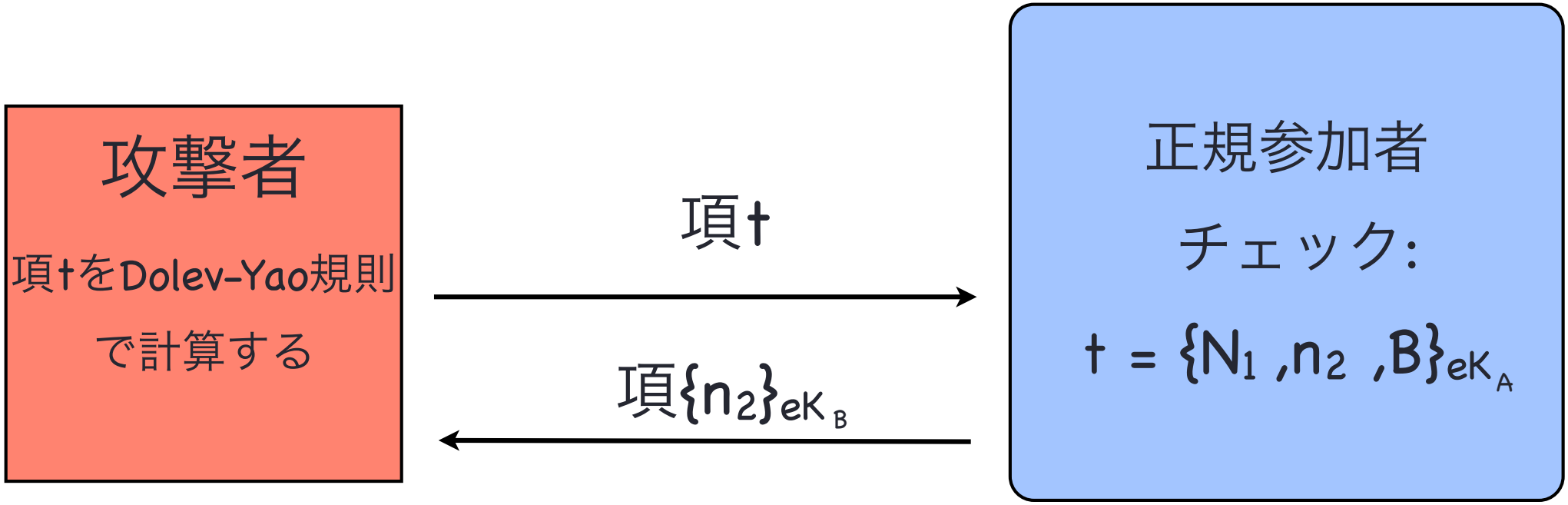
チェック:
$\pi_3(Dec(b,dK_A))=B$
$\pi_1(Dec(b,dK_A))=N_1$
計算：
$b'=Enc(\pi_2(Dec(b,dK_A)), eK_B)$

ビット列b′はフレームに追加される

条件が成立したら参加者は実行を続ける

従来の手法の記号的なラウンド

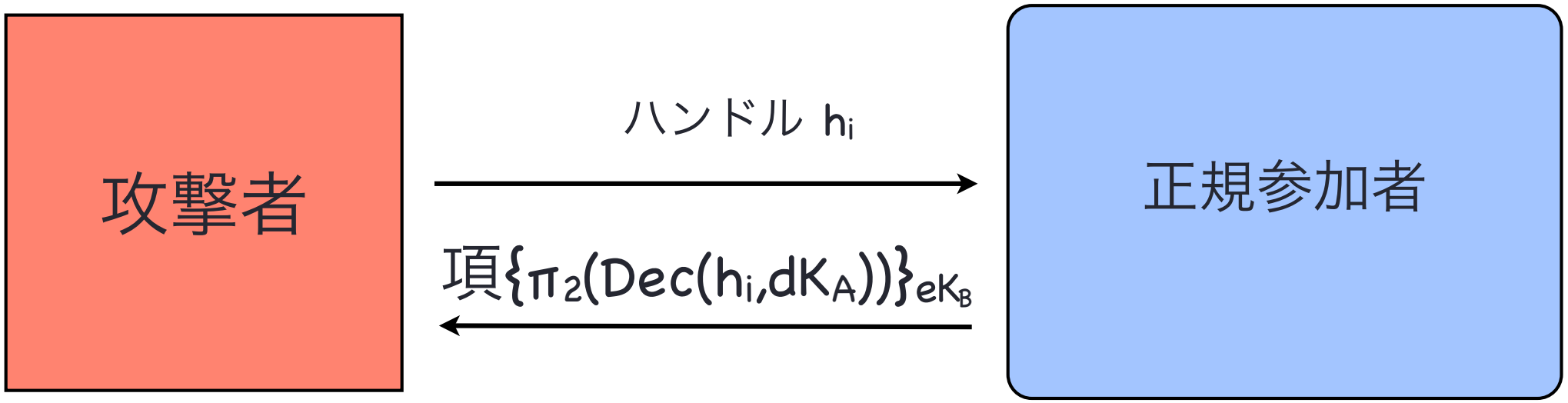
1. $A \to B : \{N_1, A\}_{eK_B}$　　2. $B \to A : \{N_1, N_2, B\}_{eK_A}$　　3. $A \to B : \{N_2\}_{eK_B}$



条件が成立したら参加者は実行を続ける

我々の記号的なラウンド

1. $A \to B : \{N_1, A\}_{eK_B}$　　2. $B \to A : \{N_1, N_2, B\}_{eK_A}$　　3. $A \to B : \{N_2\}_{eK_B}$



項$\{N_2\}_{eK_B}$はフレームに追加される

次の論理式をh_iの条件として記録する

$\pi_3(Dec(h_i,dK_A))=B$
$\pi_1(Dec(h_i,dK_A))=N_1$
$\varphi_i \vdash h_i$

それまでに記録された条件は公理に矛盾する事ができない。矛盾するトレースは捨てられる。

性質

- セキュリティ要求やチェックされた性質を形式化するため一階述語論理的な言語を定義する。
- 例えば、述語：
  - x = y：等式
  - φ ⊢ x：攻撃者はフレームφからxを計算できる
- 計算論的な意味論を定義する。計算論的なモデルは実行の一つのトレースではなく、トレース全体の無視できない確率の部分集合。
  - x = y：xとy（ビット列の確率変数）は無視できる確率以外、値が等しい
  - φ ⊢ x：攻撃者は無視できる確率以外φのビット列からxのビット列を多項式時間アルゴリズムで計算できる。
  - ∨、∧、∀、∃、¬　の解釈は難しい（通常の一階述語論理とは異なる解釈が必要）。
- 計算論的に妥当な性質を見つけて公理とする。

公理の例

- **自明なもの:**
  - Increasing capabilities: $\hat{\phi} \vdash y \rightarrow \hat{\phi}, x \vdash y$
  - No telepathy: $\mathsf{fresh}(x) \rightarrow \hat{\phi} \nvdash x$
  - Function of derivable items

    $$\hat{\phi} \vdash t_1 \wedge \hat{\phi} \vdash t_2 \wedge \ldots \wedge \hat{\phi} \vdash t_n \rightarrow \hat{\phi} \vdash f(t_1, t_2, \ldots, t_n)$$

  - Self derivability: $\hat{\phi}, t \vdash t$
- **暗号の秘匿性（IND−CCA安全性から導かれる）:**

$$\mathit{fresh}(R) \wedge \hat{\phi}, \{t\}^{R}_{eK} \vdash t \longrightarrow dK \sqsubseteq \hat{\phi} \vee \hat{\phi} \vdash t$$

記号的な意味論と健全性

- 論理式の記号的な解釈、意味論を定義しない。
  - 正規参加者のチェックの論理式が公理と矛盾するトレースを捨てる。
  - 記号的なトレースに対して、公理とチェックに矛盾しない論理式を可能な性質と呼ぶ。
- 健全性定理：計算論的な意味論で充足可能な論理式は、記号的なあるトレースに対して可能な性質である。
- すなわち、計算論的な攻撃が存在すると、記号的な攻撃も存在する。

計算論的な攻撃の存在↔セキュリティ要求の否定は充足可能である→セキュリティ要求の否定が可能な性質のトレースがある↔記号的な攻撃も存在する

十分な公理

- **計算論的に妥当な性質を全て公理化する必要はない。**
- **ある公理集合で攻撃が成功しないならば、これらの公理を満たすプリミティブを用いる場合にプロトコルは安全である**

プロトコルの自動検証

- 究極の目的は、このような記号的な攻撃が存在するかどうか自動ツールで決定する事。
- この問題は決定可能かどうかまだ分からない。
- NSLプロトコルの検証の為に十分な公理を発見した。この公理からなる集合と「〈n,Q〉 ≠ N 」条件が、記号的な攻撃がない事を含意する事を手で証明した。
- この公理集合は次のような公理を含む：
  - IND-CCA安全性から導かれる
    - 以上の秘匿性の公理
    - 知らない平文の暗号文を構成できない事を表す公理(Non-malleability)
  - 少数の自明の公理
- 他のプロトコルの為にも十分かどうかまだ分からない。

まとめ

- 計算論的に完全な記号的な攻撃者を定義した。
- 理想化なしでも、健全性定理を証明できるようになった。
- NSLプロトコルの為に十分な公理を発見して、記号的な攻撃が存在しないと証明した。さらに、我々の健全性定理によって「IND-CCA安全性を満たす暗号化を使うなら、計算論的な攻撃も存在しない」事が言える。