計算論的に完全な記号的攻撃者と鍵交換に関する分析手法

バナ・ゲルゲイ (INRIA, Paris)

長谷部浩二 (筑波大学)

岡田光弘 (慶應大学)

※ 本研究の内容はCCS'13にも採択済

発表内容

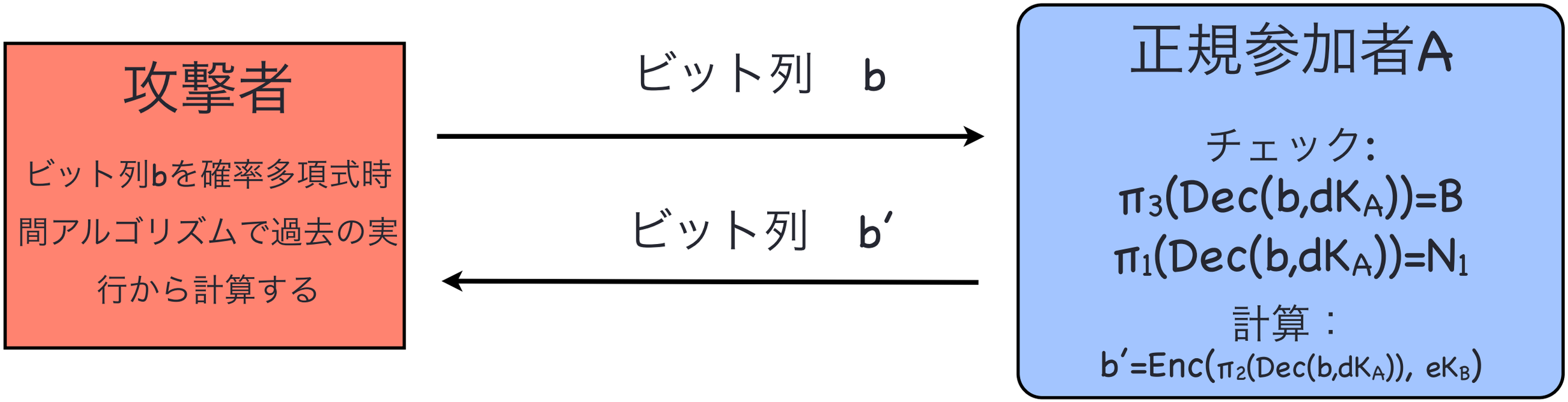
- POST'12でのBanaとComon (BC)の提案:
  - 計算論的に完全な記号的攻撃者の提案
  - 条件を伴わない計算論的健全性の定式化
- 発表内容
  - 記号的な定式化と計算論的な定式化
  - 従来の健全性定理の弱点
  - BCの提案：完全な記号的攻撃者の手法と健全性
  - BCの手法によるプロトコルの検証

計算論的な定式化と記号的な定式化

- 計算論的な定式化
  - 暗号プリミティブはビット列を入出力する確率的なアルゴリズム。
  - 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。
  - 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)
  - このため、プロトコルの安全性を自動的に検証することは難しい。
- 記号的な定式化
  - 暗号プロトコルの自動検証の為
  - 攻撃者と参加者はビット列の代わりに項を操作する
  - 暗号や電子署名などのプリミティブはブラックボックスとして扱う。
  - 確率、計算能力の限定を含まない

計算論的な実行

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



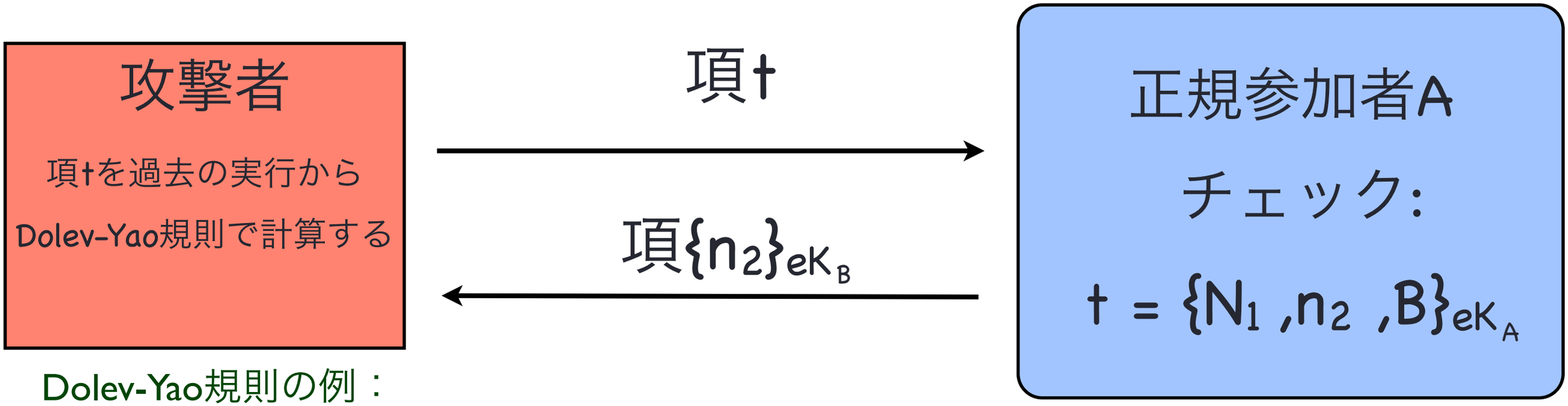
攻撃者と参加者が用いるアルゴリズムは全て確率多項式時間アルゴリズムとする。(計算能力が限定される事)

条件が成立したら参加者は実行を続ける

- プロトコルが安全であるとは、攻撃者が成功する確率が非常に低いこと「確率はセキュリティーパラメーターに対する無視できる関数である」。
- プロトコルの安全性（秘匿性、認証など）は暗号プリミティブの安全性（CCA安全性など）から導かれる。

既存の(Dolev-Yao)記号的な実行

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



Dolev-Yao規則の例：

$\{x\}_{eK}, dK \vdash x$

$x, y \vdash \langle x, y \rangle$

条件が成立したら参加者は実行を続ける

- 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（Dolev-Yaoルール）。列挙されていない動作はできない。
- 自動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（例えば、過去の実行 $\vdash N_1$ ）
- 単純だが、多くの攻撃を見つけることができる。
- 計算論的な攻撃をすべて見つけられると嬉しい。

理想化された健全性定理

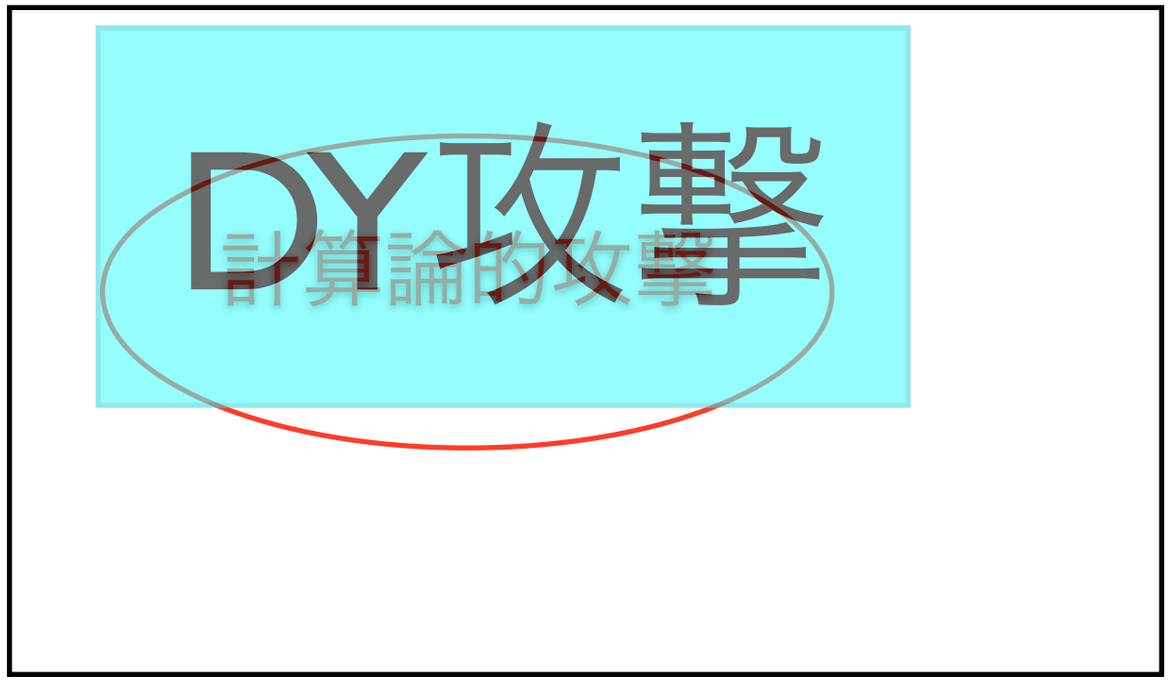
- 記号的な定式化の計算論的健全性
  - 成功する記号的な攻撃が存在しないならば成功する計算論的な攻撃も存在しない。
  - すなわち、記号的な攻撃者は計算論的攻撃者より強い。
- 計算論的実装についての非常に強い仮定
  - 健全性定理は多くの文献で証明されている。
  - 今までの文献の健全性定理は全てプロトコルの安全性に必要でない不自然な仮定を用いる。
  - 暗号プロトコルの実装についてのこのような仮定の例：
    - ビット列が一意に記号列にパースできる
      - 鍵のビット列の半分を暗号化で使って、残りの半分を暗号文と連結する。
    - 攻撃者は参加者をプロトコル実行中に買収しない

不自然な仮定の理由

- Dolev-Yaoの手法では、攻撃者が許されるすべての振る舞いを、Dolev-Yaoルールという小さな集合によって表す

不自然な仮定の理由

- Dolev-Yaoの手法では、攻撃者が許されるすべての振る舞いを、Dolev-Yaoルールという小さな集合によって表す
- 記号的攻撃者を十分強くするためにDolev-Yaoルールを拡張しても、記号的攻撃に対応しない計算論的攻撃が残ってしまう。



不自然な仮定の理由

- Dolev-Yaoの手法では、攻撃者が許されるすべての振る舞いを、Dolev-Yaoルールという小さな集合によって表す
- 記号的攻撃者を十分強くするためにDolev-Yaoルールを拡張しても、記号的攻撃に対応しない計算論的攻撃が残ってしまう。
- このため、計算論的な攻撃者を制限して、残ってしまう攻撃ができないようにする必要がある。

すなわち、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。

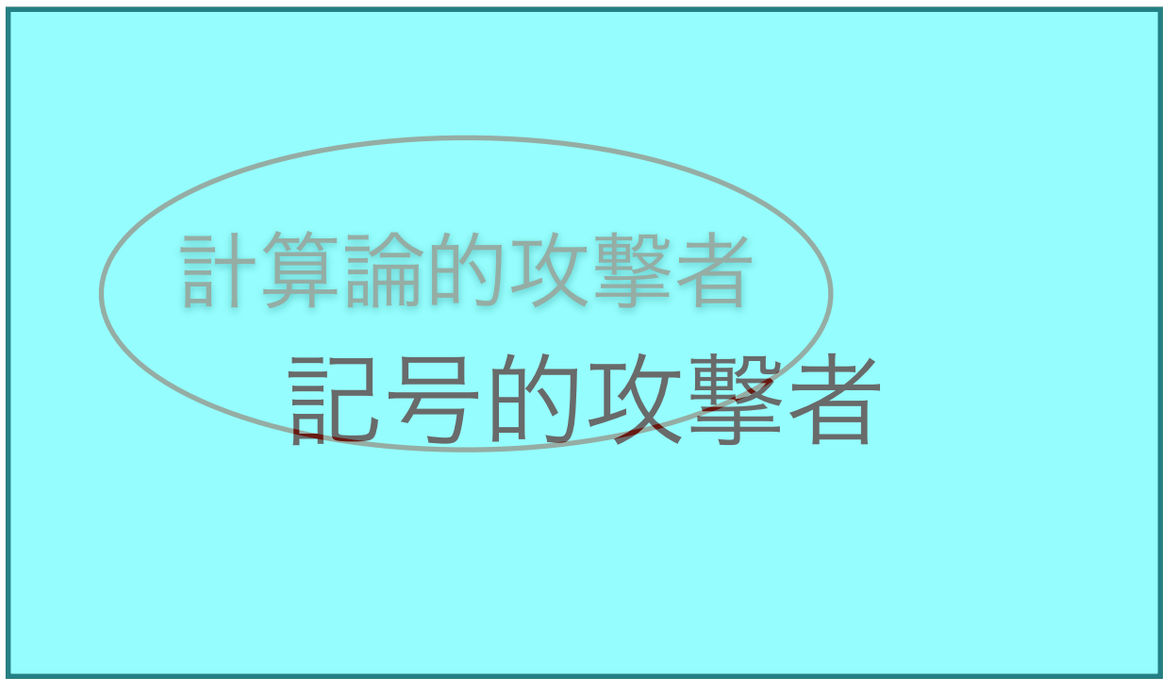
Bana-Comon（POST'12）アイディア

- 計算論的実装についての典型的な仮定（CCA安全性など）は、攻撃者が破り得ない事を指定するにも関わらず、
- 従来の記号的方法では攻撃者に許される事を指定する。

- 我々は記号的な定式化を計算論的な仮定に合わせる:
- 記号的攻撃者も破り得ない事を指定する記号検証方法を構成しよう。

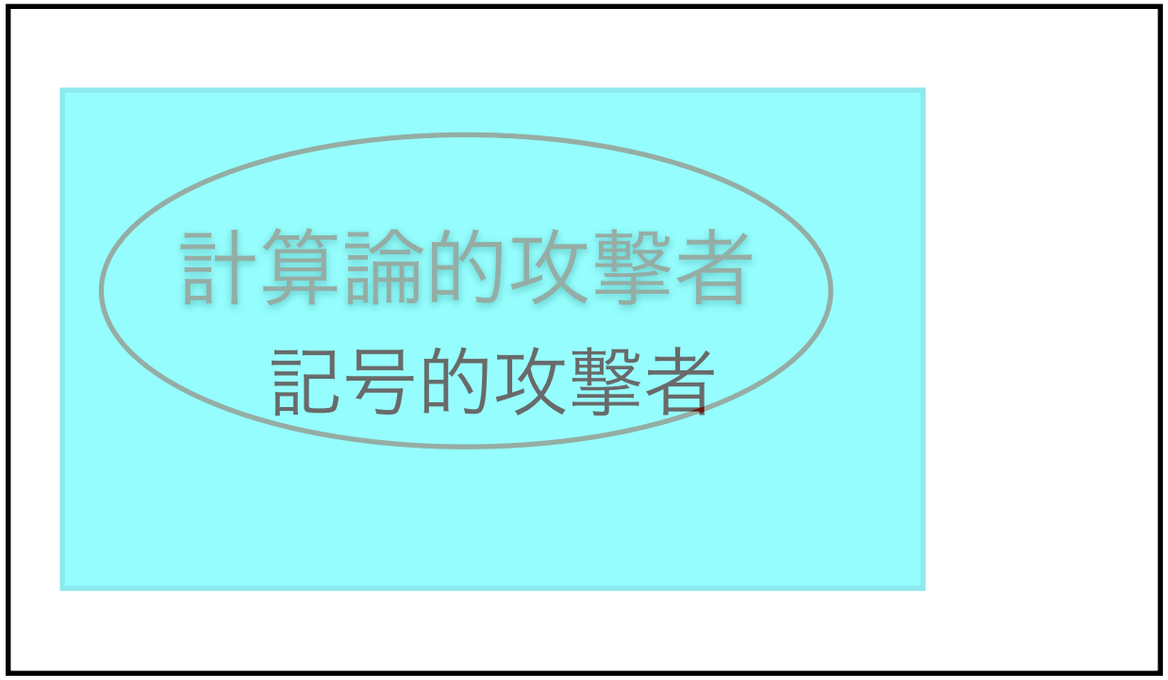
BC記号的検証方法

- 記号的攻撃者が破り得ないことを公理として列挙する。
- この公理に矛盾しないいかなる動作も可能であるとする。
- 公理
  - 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計算論的仮定（CCA安全性など）から導かれる。



BC記号的検証方法

- 記号的攻撃者が破り得ないことを公理として列挙する。
- この公理に矛盾しないいかなる動作も可能であるとする。
- 公理
  - 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計算論的仮定（CCA安全性など）から導かれる。



Bana-Comonの記号的な実行

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

ハンドル h_i

項$\{\pi_2(Dec(h_i,dK_A))\}_{eK_B}$

項 $\{\pi_2(Dec(h_i,dK_A))\}_{eK_B}$

は攻撃者の知識φに追加される

正規参加者A

次の論理式をh_iの条件として記録する

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

- 記号的攻撃の成功：プロトコルの安全性を表す論理式の否定、公理、正規参加者のチェックの論理式が互いに矛盾しない。
- 例えば、$\varphi \triangleright N_1$、公理、正規参加者のチェックの論理式が互いに矛盾しない。

Bana-Comonの記号検証の健全性

- 健全性定理：
  - 公理が計算論的に正しければ、 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。
- 従来方法と異なり、この健全性定理のため、不自然な仮定は不要。
- 健全性定理はすばらしいが、提案方法で本当にプロトコルが検証できるのか？
- Bana, Adao, Sakurada (FSTTCS'12): 公理系およびそれによるNeedham-Schroeder-Loweプロトコルの安全性証明
  - 秘密鍵がネットワークに送られないことが前提
- 本研究：秘密鍵の送信についての前提が不要

導出可能性と鍵の部分情報の漏洩

- 以下の3つの述語を基にする
  - $t_1 = t_2$： t_1 と t_2 は等しい (ただし、無視できる確率を除いて)
  - $\varphi, t_1, \ldots, t_n \triangleright t$：攻撃者は，過去に実行したプロトコルから得られる情報　φ　と $t_1, \ldots, t_n$ から t を導出できる．
  - $\varphi, t_1, \ldots, t_n \blacktriangleright K$：$\varphi, t_1, \ldots, t_n$ に現れる情報を攻撃者が知ると，鍵 K によって暗号化されたメッセージの秘匿性が保証されなくなる.
- 以下の２の公理がCCA2に対して健全となるように，述語 ▸ の計算論的意味論を与えたい：

$$\varphi, \vec{x}, \{x\}^R_{eK} \triangleright y \longrightarrow \varphi, \vec{x}, x \blacktriangleright K \vee \varphi, \vec{x} \triangleright y$$

$$\mathsf{keyfresh}(K; \varphi) \longrightarrow \varphi \not\blacktriangleright K$$

- しかし，それは不可能である．

オラクルの呼び出し

- 「オラクルの呼び出し」の概念を伴った導出可能性や鍵の漏洩の定義が必要：
  - $\varphi, t_1, \ldots, t_n \triangleright^{\mathcal{O}} t$ ：攻撃者は，暗号化と復号化のオラクルを用いて，φ と $t_1, \ldots, t_n$ から得られる情報から t を導出することができる．
  - $\varphi, t_1, \ldots, t_n \blacktriangleright^{\mathcal{O}} K$ ：攻撃者が，$\varphi, t_1, \ldots, t_n$ に現れる情報とオラクルを用いると，鍵 K によって暗号化されたメッセージの秘匿性が保証されなくなる.
- 以上の述語を導入すると，以下の公理の健全性を示すことができる:

$$\varphi, \vec{x}, \{x\}^R_{eK} \triangleright^{\mathcal{O}} y \longrightarrow \varphi, \vec{x}, x \blacktriangleright^{\mathcal{O}} K \vee \varphi, \vec{x} \triangleright^{\mathcal{O}} y$$

$$\mathsf{keyfresh}(K; \varphi) \longrightarrow \varphi \not\blacktriangleright^{\mathcal{O}} K$$

- ここでは，暗号化の際の乱数の入力が新鮮（fresh）であることを仮定している：

$$\mathsf{RandGen}(K) \wedge \mathsf{fresh}(R; \varphi, \vec{x}, x, y, K) \wedge \vec{x}, x, y \preccurlyeq \varphi \wedge \varphi, \vec{x}, \{x\}^R_{eK} \triangleright^{\mathcal{O}} y \longrightarrow \varphi, \vec{x}, x \blacktriangleright^{\mathcal{O}} K \vee \varphi, \vec{x} \triangleright^{\mathcal{O}} y$$

その他の公理 1

- 鍵の漏洩に関する同様の公理を追加:

$$\varphi, \vec{x}, \{x\}^R_{eK} \blacktriangleright^{\mathcal{O}} K' \longrightarrow \varphi, \vec{x}, x \blacktriangleright^{\mathcal{O}} K \vee \varphi, \vec{x} \blacktriangleright^{\mathcal{O}} K'$$

- 復号化のオラクル呼び出しについての公理を追加:

$$\varphi, \vec{x} \rhd^{\mathcal{O}} y \wedge \forall x R(y = \{x\}^R_{eK} \rightarrow \{x\}^R_{eK} \not\sqsubseteq \varphi, \vec{x})$$
$$\longrightarrow \varphi, \vec{x} \rhd^{\mathcal{O}} dec(y, dK)$$

その他の公理２

- No telepathy

$$\mathsf{fresh}(x;\varphi) \longrightarrow \varphi \not\rhd^{\mathcal{O}} x$$

- Freshly generated items don’t help

$$\mathsf{fresh}(x;\varphi,\vec{x},y) \wedge \varphi,\vec{x},x \rhd y \longrightarrow \varphi,\vec{x} \rhd y$$

- いくつかの自明な公理

Self derivability: $\varphi,\vec{x},x \rhd x$

Increasing capabilities: $\varphi,\vec{x} \rhd y \longrightarrow \varphi,\vec{x},x \rhd y$

Commutativity: If $\vec{x}'$ is a permutation of $\vec{x}$, then $\varphi,\vec{x} \rhd y \longrightarrow \varphi,\vec{x}' \rhd y$

Transitivity of derivability: $\varphi,\vec{x} \rhd \vec{y} \wedge \varphi,\vec{x},\vec{y} \rhd \vec{z} \longrightarrow \varphi,\vec{x} \rhd \vec{z}$

Functions are derivable: $\varphi,\vec{x} \rhd f(\vec{x})$
This axiom is sound as long as functions are interpreted as PT computable algorithms.

公理のライブラリー

- 対称鍵Needham-Schroeder, Needham-Schroeder-Lowe, Otway-Reesプロトコルの安全性証明のための公理

- $\varphi, \vec{x}, x \rhd x$
- $\varphi, \vec{x} \rhd y \longrightarrow \varphi, \vec{x}, x \rhd y$
- $\vec{x}'$ permut. of $\vec{x}$: $\varphi, \vec{x} \rhd y \longrightarrow \varphi, \vec{x}' \rhd y$
- $\varphi, \vec{x} \rhd \vec{y} \ \wedge \ \varphi, \vec{x}, \vec{y} \rhd \vec{z} \longrightarrow \varphi, \vec{x} \rhd \vec{z}$
- $\varphi, \vec{x} \rhd f(\vec{x})$
- fresh$(x; \varphi, \vec{x}, y) \wedge \vec{x}, y \preccurlyeq \varphi \wedge \varphi, \vec{x}, x \rhd y \longrightarrow \varphi, \vec{x} \rhd y$

- $\varphi, \vec{x} \rhd x \longrightarrow \varphi, \vec{x} \rhd^{\mathcal{O}} x.$
- $\varphi, \vec{x} \rhd^{\mathcal{O}} y \longrightarrow \varphi, \vec{x}, x \rhd^{\mathcal{O}} y$
- $\vec{x}'$ permut. of $\vec{x}$: $\varphi, \vec{x} \rhd^{\mathcal{O}} y \longrightarrow \varphi, \vec{x}' \rhd^{\mathcal{O}} y$
- $\varphi, \vec{x} \rhd^{\mathcal{O}} \vec{y} \ \wedge \ \varphi, \vec{x}, \vec{y} \rhd^{\mathcal{O}} \vec{z} \longrightarrow \varphi, \vec{x} \rhd^{\mathcal{O}} \vec{z}$
- fresh$(x; \varphi, \vec{x}, y) \wedge \vec{x}, y \preccurlyeq \varphi \wedge \varphi, \vec{x}, x \rhd^{\mathcal{O}} y \longrightarrow \varphi, \vec{x} \rhd^{\mathcal{O}} y$

- $\varphi, \vec{x} \rhd^{\mathcal{O}} K \longrightarrow \varphi, \vec{x} \blacktriangleright^{\mathcal{O}} K$
- $\varphi, \vec{x} \blacktriangleright^{\mathcal{O}} K \longrightarrow \varphi, \vec{x}, x \blacktriangleright^{\mathcal{O}} K$
- $\vec{x}'$ a permut. of $\vec{x}$: $\varphi, \vec{x} \blacktriangleright^{\mathcal{O}} K \longrightarrow \varphi, \vec{x}' \blacktriangleright^{\mathcal{O}} K$
- $\varphi, \vec{x} \rhd^{\mathcal{O}} \vec{y} \wedge \varphi, \vec{x}, \vec{y} \blacktriangleright^{\mathcal{O}} K \longrightarrow \varphi, \vec{x} \blacktriangleright^{\mathcal{O}} K$
- fresh$(x; \varphi, \vec{x}, y) \wedge \vec{x}, y \preccurlyeq \varphi \wedge \varphi, \vec{x}, x \blacktriangleright^{\mathcal{O}} y \longrightarrow \varphi, \vec{x} \blacktriangleright^{\mathcal{O}} y$

- keyfresh$(K; \varphi) \longrightarrow \varphi \not\blacktriangleright^{\mathcal{O}} K$
- fresh$(x; \varphi) \longrightarrow \varphi \not\rhd^{\mathcal{O}} x$
- RandGen$(K) \ \wedge$ fresh$(R; \varphi, \vec{x}, x, y, K) \ \wedge \ \vec{x}, x, y \preccurlyeq \varphi \ \wedge \ \varphi, \vec{x}, \{x\}^R_{eK} \rhd^{\mathcal{O}} y \longrightarrow \varphi, \vec{x}, x \blacktriangleright^{\mathcal{O}} K \vee \varphi, \vec{x} \rhd^{\mathcal{O}} y$
- RandGen$(K) \wedge$ RandGen$(K') \wedge$ fresh$(R; \varphi, \vec{x}, x, K, K') \ \wedge \ \vec{x}, x \preccurlyeq \varphi \ \wedge \ \varphi, \vec{x}, \{x\}^R_{eK'} \blacktriangleright^{\mathcal{O}} K \longrightarrow \varphi, \vec{x}, x \blacktriangleright^{\mathcal{O}} K' \vee \varphi, \vec{x} \blacktriangleright^{\mathcal{O}} K$
- $\varphi, \vec{x} \rhd^{\mathcal{O}} y \ \wedge \ \forall x R(y = \{x\}^R_{eK} \rightarrow \{x\}^R_{eK} \not\sqsubseteq \varphi, \vec{x}) \longrightarrow \varphi, \vec{x} \rhd^{\mathcal{O}} dec(y, dK)$

対称鍵Needham-Schroederプロトコル

1. $A \rightarrow B : A$
2. $B \rightarrow A : \{A, N_1\}_{K_{BT}}$
3. $A \rightarrow T : \langle A, B, N_2, \{A, N_1\}_{K_{BT}} \rangle$
4. $T \rightarrow A : \{N_2, B, K, \{K, N_1, A\}_{K_{BT}}\}_{K_{AT}}$
5. $A \rightarrow B : \{K, N_1, A\}_{K_{BT}}$
6. $B \rightarrow A : \{N_3\}_K$
7. $A \rightarrow B : \{N_3 - 1\}_K$

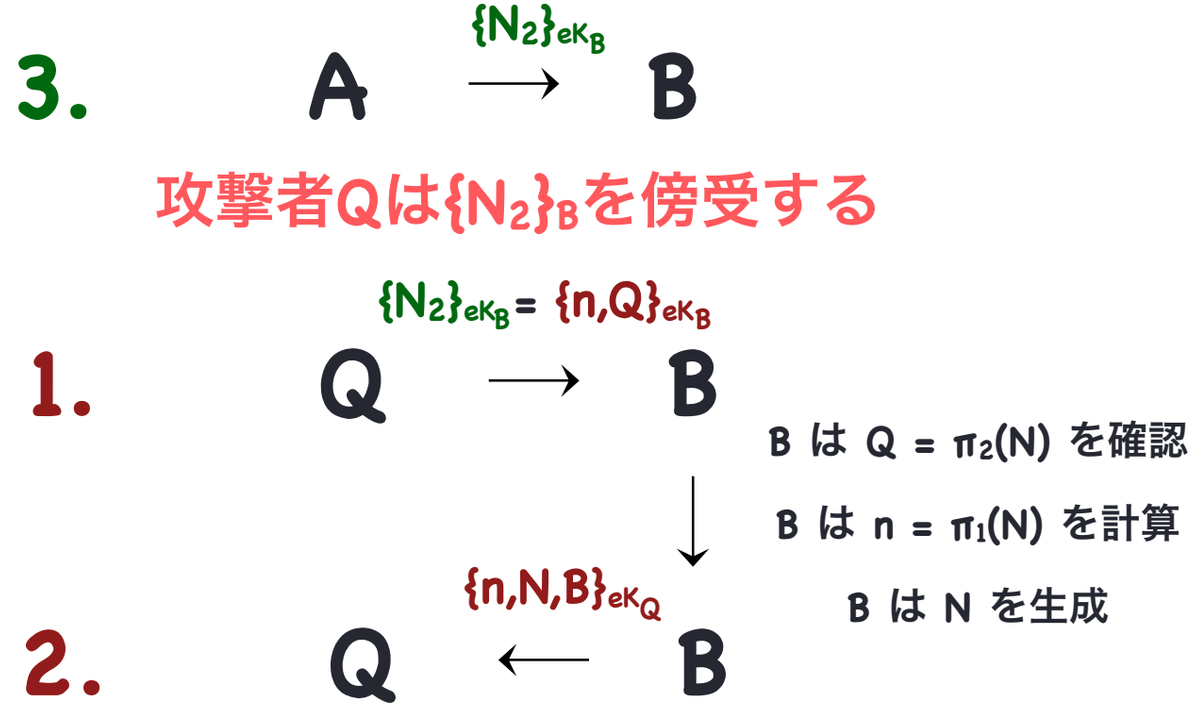
まとめ

- BanaとComonの手法を，鍵がネットワークに送られた場合も扱えるように拡張した。
- 本提案により、不自然な仮定無しでも、健全性定理を証明できるようになった。
- 公理のライブラリーを構築し、様々なプロトコルの検証に適用した。
- 今後の課題：検証の自動化、Observational equivalence
- 詳細は、CCS'13の採択論文を参照（私のホームページから閲覧可能）

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。