Abadi-Rogawayによる暗号メッセージの解析手法とその健全性・完全性について

萩原茂樹　米崎直樹
東京工業大学大学院情報理工学研究科
計算工学専攻

セキュリティの理論研究の２つのアプローチ

- セキュリティを理論的に解析する研究には、二つのアプローチがある。
  - 記号論に基づいたアプローチ
    - 暗号は完全であると仮定して、セキュリティプロトコル解析に適用してきた。
    - 暗号理論の研究者は、計算論に基づくどのような安全性を捉えているか理解したい。
  - 計算論に基づいたアプローチ
    - モデルが実際の暗号方式を反映していて、暗号理論の研究者には受け入れられていた。
    - 記号論の研究者は、これで複雑なプロトコルに対する巧みな攻撃に対する耐性が捉えられるか理解したい。

- これら２つのアプローチによる解析結果の対応が明らかでない。

AbadiとRogawayによる研究

- これら２つのアプローチによる解析結果の対応の研究として、[M.Abadi,P.Rogaway2000,2002]の研究がある。

  完全な対象鍵暗号方式を前提としてメッセージの等価性を記号論的に定義し、メッセージが等価ならばそれらは計算論的に識別不可能であることを示した。すなわち記号論によるメッセージの識別不可能性の、計算論的意味に対する健全性を示した。

Abadi-Rogawayによるアプローチの発展研究

- 暗号文の識別不可能性の記号論による解釈の完全性、[D.Micciancio, B.Warinschi2002],[O.Horvitz, V.Gligor2003]
- 部分情報の漏洩の考慮[P.Adao,G.Bana2005]
- 非対称鍵への拡張[D.Micciancio, B.Warinschi2004]
- 並行システムへの拡張[M.Abadi,J.Jurjens2001]
- 等式規則を持つ記号論への拡張[G.Bana,P.Mohassel,T.Stegers2006]
  ・・

- 本報告では、最も基本的なM.AbadiとP.Rogawayによる健全性と、D.MicciacioとB.warinschiによる完全性を紹介する。

Abadi-Rogaway が用いたメッセージを記述する言語（構文）

M,N ::=		expressions
	K	key (for $K \in$ **Keys**)
	i	bit (for $i \in$ **Bool**)
	(M,N)	pair
	$\{M\}_K$	encryption (for $K \in$ **Keys**)

例: 0とK_2を連結し、K_1で暗号化し、その結果をK_3で暗号化してできたメッセージ

$$\{\{(0,K_2)\}_{K_1}\}_{K_3}$$

２種類の意味論

メッセージとそれらの等価関係に対して、以下の２種類の意味論を与えている。

- 記号論に基づく意味論
  - メッセージMは先述のメッセージの表現と同じ表現と解釈する。
  - メッセージの等価性を、「MとM'から読み取れる情報は同じ」と解釈する。

- 計算論に基づく意味論
  - メッセージMは暗号スキーマ（アルゴリズム）から生成される対応するデータ（0,1の列）と解釈する。
  - メッセージの等価性を、「MとM'に対応するデータの差を識別できる確率的多項式時間チューリング機械(PPT)が無い」、つまり計算論的に識別不可能であることと解釈する。

記号論に基づく意味論

メッセージの内容がどこまで見えるかを表現する為の言語

$P,Q ::=$	expressions	
K	key (for $K \in$ **Keys**)	
i	bit (for $i \in$ **Bool**)	
(P,Q)	pair	
$\{P\}_K$	encryption (for $K \in$ **Keys**)	
$\square$	undecryptable	

ここで、□は内容が解読できない暗号メッセージを表す。

メッセージの内容がどこまで見えるかを計算

$$pattern(M) \quad = \quad p(M,\{K \in \mathbf{Keys} \mid M \vdash K\})$$

ここで、p(M,T)は鍵の集合Tを知っている者によるMの内容がどこまで見えるかを計算する関数。

$$\begin{aligned}
p(K,T) &= K && (\text{for } K \in \mathbf{Keys}) \\
p(i,T) &= i && (\text{for } i \in \mathbf{Bool}) \\
p((M,N),T) &= (p(M,T), p(N,T)) \\
p(\{M\}_K,T) &= \begin{cases} \{p(M,T)\}_K & \text{if } K \in T \\ [] & \text{otherwise} \end{cases}
\end{aligned}$$

（注）

（注）暗号メッセージはそれを暗号化した鍵のみで復号化できると仮定している。

作成できるメッセージの導出

- $M \vdash 0$ and $M \vdash 1$,
- $M \vdash M$,
- if $M \vdash N_1$ and $M \vdash N_2$ then $M \vdash (N_1, N_2)$,
- if $M \vdash (N_1, N_2)$ then $M \vdash N_1$ and $M \vdash N_2$,
- if $M \vdash N$ and $M \vdash K$ then $M \vdash \{N\}_K$,
- if $M \vdash \{N\}_K$ and $M \vdash K$ then $M \vdash N$.

Patternの例

例: $pattern((K_3,\{\{(0,K_2)\}_{K_1}\}_{K_3})) = p((K_3,\{\{(0,K_2)\}_{K_1}\}_{K_3}),\{K_3\})$

$= (K_3,\{\ []\ \}_{K_3})$

すなわち、K3のみを知ることができるため、$(K_3,\{\{(0,K_2)\}_{K_1}\}_{K_3})$ より得られる情報は　$(K_3,\{\ []\ \}_{K_3})$ である。

なぜなら、このメッセージはK3でのみ復号化できると仮定しているので、K3で暗号化されていることだけは知ることができるからである。

メッセージから取り出せる情報の等価性

$M \equiv N$ if and only if $pattern(M) = pattern(N)$

- 例えば $(K,\{0\}_K) \equiv (K,\{1\}_K)$ ではない。なぜなら、
  $pattern((K,\{0\}_K)) \equiv (K,\{0\}_K)$
  $pattern((K,\{1\}_K)) \equiv (K,\{1\}_K)$
- 一方 $(K,\{(\{0\}_{K'},0)\}_K) \equiv (K,\{(\{1\}_{K'},0)\}_K)$
  どちらのパターンも $(K,\{(\square,0)\}_K)$ となる。

メッセージから取り出せる情報の等価性（続き）

鍵を束縛変数としたときのα同値性を含む。

$$M \cong N \quad \text{if and only if} \quad \text{there exits a bijection } \sigma \text{ on such that } M \equiv N\sigma$$

例: α同値性により以下が成り立つ。

$$K \cong K'$$

$$(K,\{(\{0\}_{K''},0)\}_K) \cong (K',\{(\{1\}_{K''},0)\}_{K'})$$

メッセージから取り出せる情報の等価性の例

- 暗号メッセージから内容の長さに関する情報を取り出すことができない。

$$\{(((1,1),(1,1)),((1,1),(1,1)))\}_K \cong \{0\}_K .$$

- 二つの暗号メッセージを見ても、内容が同じかどうか分からない。

$$(\{0\}_K,\{0\}_K) \cong (\{0\}_{K,}\{1\}_K).$$

- 二つの暗号メッセージを見ても、利用された鍵が同じかどうか分からない。

$$(\{0\}_K,\{1\}_K) \cong (\{0\}_{K,}\{1\}_{K'}).$$

計算論に基づくモデルによる意味論

暗号スキーマ

暗号スキーマは以下のアルゴリズムの三つ組み

$$\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$$

$$\mathcal{K} : \text{Parameter} \times \text{Coins} \rightarrow \text{Key}$$

$$\mathcal{E} : \text{Key} \times \text{String} \times \text{Coins} \rightarrow \text{Ciphertext}$$

$$\mathcal{D} : \text{Key} \times \text{String} \rightarrow \text{Plaintext}$$

- κ と ε は確率アルゴリズムである。Coinsは実行中振られたコインの結果の列である。

- 暗号化に用いた鍵と同じ鍵で復号化できる。

$$\forall k \in \text{Key} \forall m \in \text{Plaintext}\ \forall r \in \text{Coins}\ \mathcal{D}_k(\mathcal{E}_k(m, r)) = m$$

- 異なる鍵で復号化できてしまうこともあり得る。

$$\mathcal{D}_{k'}(\mathcal{E}_k(m, r)) = m'$$

暗号スキーマの性質の分類

- 二つの暗号データから、
  - それらの内容に繰り返しがあるかどうか分かるかどうか？
  - 同じ鍵を使って暗号化したか分かるかどうか？
- 暗号データから
  - メッセージの長さに関する情報が取れるかどうか？

- これらが成り立つかどうかにより、８通りに分類することができる。（Type-0 から type-7 まで考えることができる。）

Type-0安全

- 暗号データから、以下に関する如何なる情報も取れない。
  - 内容の繰り返し
  - 使用した鍵
  - 内容の長さ

定義

任意の確率的多項式時間チューリング機械(PPT)の敵Aに対して、以下で定義される敵のアドバンテージが無視できるとき、暗号スキーマΠはtype-0安全であるという。

$$\mathrm{Adv}^{0}_{\Pi[\eta]}(A) \overset{\mathrm{def}}{=} \Pr\left[k,k' \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)=1\right] - \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)=1\right]$$

ここで**0**は復号化失敗を表す特別なデータ

オラクルチューリング機械

- $A^{f(\cdot),g(\cdot)}$ ：オラクルPPT
  - オラクルと呼ばれるブラックボックスに問い合わせができるチューリング機械
  - f(・)と g(・)はオラクル（この場合は２種類のオラクルを持っている。）
  - f(・)の・は問い合わせ用の入力を表している。Aがプログラム中でオラクルf(・)にxで問い合わせると、1ステップで返答f(x)が得られる。

- $A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)$: 二つの暗号化オラクルを持つオラクルPPT
  - $\mathcal{E}_k(\cdot)$ は問い合わせxに対して、$\mathcal{E}_k(x)$ を計算し、その結果のデータを返答するオラクル。$\mathcal{E}_{k'}(\cdot)$ も同様。

- $A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)$
  - $\mathcal{E}_k(\mathbf{0})$は、どのような問い合わせ$x$に対しても、$x$の値に関わらず、$\mathcal{E}_k(\mathbf{0})$ を計算し、その結果のデータを返答するオラクル。

無視できる関数ε

定義: ε ：N→R が以下を満たすとき、 ε を無視できるという。

$$\forall c > 0 \exists N_c \forall \eta \geq N_c \ (\varepsilon(\eta) \leq \eta^{-c})$$

- どんな多項式の逆数に対しても、引数を十分大きく取れば、それよりも小さくなるという意味。
  - 例　$\varepsilon(\eta) = 2^{-\eta}$

性質: 任意の多項式fに対して、

$$\lim_{\eta \to \infty} \varepsilon(\eta) \cdot f(\eta) = 0$$

が成り立つ。

暗号データを調べるPPT

- $A^{f(\cdot),g(\cdot)}$ を
  - 2つの暗号データをみて、
    (1). 内容に繰り返しがあるかどうか、
    (2). 同じ鍵で暗号化されたデータかどうか
  - 1つの暗号データを見て、
    (3). 内容の長さに関する情報が取り出せるかどうか

  を調べようとするオラクルPPTとみる。

内容に繰り返しがあるか調べるPPT

(1). 2つの暗号データをみて、内容に繰り返しがあるかどうかを調べる為には、$A^{f(\cdot),g(\cdot)}$は

- (i) 異なるデータd1,d2をオラクルfやgに問い合わせる。
- (ii) 返答として得られた二つの暗号データを調べて、繰り返しがあれば0を、無ければ1を出力する。

もし、大きな確率でそれを調べることが可能ならば、以下のアドバンテージは大きくなるはずである。

$$\mathsf{Adv}^{0}_{\Pi[\eta]}(A) \overset{\text{def}}{=} \Pr\left[k,k' \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)=1\right] - \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)=1\right]$$

なぜなら、$A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)$ は大きな確率で1を出力し、$A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)$ は0を出力する為。

Type-0安全性はこのアドバンテージが無視できると定義することで、二つの暗号データの内容の繰り返しを調べることができないことを保証している。

使用された鍵データが同じかどうか調べるPPT

(2). 2つの暗号データをみて、使用された鍵データが同じかどうかを調べる為には、$A^{f(\cdot),g(\cdot)}$ は

- (i) データd1,d2をオラクルfとgに問い合わせる。
- (ii) 返答として得られた二つの暗号データを調べて、使用された鍵データが同じであれば0を、そうでなければ1を出力する。

もし、大きな確率で調べることが可能ならば、以下のアドバンテージは大きくなるはずである。

$$\mathbf{Adv}^{0}_{\Pi[\eta]}(A) \stackrel{\text{def}}{=} \Pr\left[k,k' \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)=1\right] - \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)=1\right]$$

なぜなら、$A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)$ は大きな確率で1を出力し、 $A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)$ は0を出力する為。

Type-0安全性はこのアドバンテージが無視できると定義することで、二つの暗号データの使用された鍵データが同じかどうかを調べることができないことを保証している。

内容データの長さを調べるPPT

(3). 暗号データをみて、内容データの長さに関する情報がとれるかどうか調べる為には、 $A^{f(\cdot),g(\cdot)}$ は

- (i) データdをオラクルfやgに問い合わせる。
- (ii) 返答として得られた暗号データを調べて、内容データの長さがdと同じであると判断したときは1を、そうでなければ0を出力する。

もし、大きな確率で調べることが可能ならば、以下のアドバンテージは大きくなるはずである。

$$\mathsf{Adv}^{0}_{\Pi[\eta]}(A) \overset{\text{def}}{=} \Pr\left[k,k' \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)=1\right] - \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)=1\right]$$

なぜなら、 $A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)$ は大きな確率で1を出力し、 $A^{\mathcal{E}_k(\mathbf{0}),\mathcal{E}_k(\mathbf{0})}(\eta)$ は0を出力する為。

Type-0安全性はこのアドバンテージが無視できると定義することで、暗号データの長さに関する情報が得られないことを保証している。

Type-1 安全

- 暗号文から、以下に関する如何なる情報も取れない。
  - 内容の繰り返し、
  - 使用した鍵

定義　任意のPPTの敵Aに対して、以下で定義される敵のアドバンテージが無視できるとき、暗号スキーマ Π はtype-1安全であるという。

$$\mathrm{Adv}^{1}_{\Pi[\eta]}(A) \overset{\mathrm{def}}{=} \Pr\left[k,k' \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\cdot),\mathcal{E}_{k'}(\cdot)}(\eta)=1\right] - \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(0^{|\cdot|}),\mathcal{E}_k(0^{|\cdot|})}(\eta)=1\right]$$

Type-2安全

- 暗号文から、以下に関する如何なる情報も取れない。
  - 内容の繰り返し、
  - 内容の長さ

定義　任意のPPTの敵Aに対して、以下で定義される敵のアドバンテージが無視できるとき、暗号スキーマ∏はtype-2安全であるという。

$$\mathbf{Adv}^{2}_{\Pi[\eta]}(A) \overset{\text{def}}{=} \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\cdot)}(\eta) = 1\right] - \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\mathbf{0})}(\eta) = 1\right]$$

Type-３安全

- 暗号文から、以下に関する如何なる情報も取れない。
  - 内容の繰り返し

定義　任意のPPTの敵Aに対して、以下で定義される敵のアドバンテージが無視できるとき、暗号スキーマ∏はtype-3安全であるという。

$$\mathbf{Adv}^{3}_{\Pi[\eta]}(A) \overset{\text{def}}{=} \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(\cdot)}(\eta) = 1\right] - \Pr\left[k \xleftarrow{R} \mathcal{K}(\eta) : A^{\mathcal{E}_k(0^{|\cdot|})}(\eta) = 1\right]$$

メッセージの解釈

$\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D}), \eta \in Parameter$

であるとき、以下のようにメッセージMの解釈 $[[M]]_{\Pi\eta}$ を定義する。

- 最初にK(η)を実行し、その結果の鍵データをMに出現する鍵記号Kの解釈とする。
- 記号0,1は、そのままデータ0,1と解釈する。
- (M,N) は、MとNの解釈データを連結したデータと解釈する。
- $\{M\}_K$についてはMの解釈データをKの鍵データでその都度暗号化し、その結果のデータと解釈する。

・ 具体的には次のアルゴリズム

解釈されるデータを計算する アルゴリズム

algorithm $\mathrm{INITIALIZE}_{\eta}(M)$

for $K \in Keys(M)$ **do** $\tau(K) \xleftarrow{R} \mathcal{K}(\eta)$

τは鍵記号から鍵データへの関数

algorithm $\mathrm{CONVERT}(M)$

if $M = K$ where $K \in$ **Keys** **then**

return $\langle \tau(K), \text{"key"} \rangle$

if $M = b$ where $b \in$ **Bool** **then**

return $\langle b, \text{"bool"} \rangle$

if $M = (M_1, M_2)$ **then**

return $\langle \mathrm{CONVERT}(M_1), \mathrm{CONVERT}(M_2), \text{"pair"} \rangle$

if $M = \{M_1\}_K$ **then**

$x \xleftarrow{R} \mathrm{CONVERT}(M_1)$

$y \xleftarrow{R} \mathcal{E}_{\tau(K)}(x)$

return $\langle y, \text{"ciphertext"} \rangle$

注意

- メッセージ中に同じ記号が複数回出現する場合、
  - $<k_1,k_2,\text{"pair"}> \leftarrow [[(K,K)]]_{\sqcap\eta}$ のとき、鍵データk_1とk_2は必ず同じ値になる。なぜなら、最初に鍵生成アルゴリズムK(η)の実行により得られたデータが、どちらの記号Kの解釈となる為である。

  - $<c_1,c_2,\text{"pair"}> \leftarrow [[(\{M\}_K,\{M\}_K)]]_{\sqcap\eta}$ である暗号文データc1とc2は同じ値になるとは限らない。なぜなら、 二つの$\{M\}_K$の出現に対して、暗号アルゴリズム $\varepsilon(\eta)$がそれぞれ２回実行され、その結果のデータがそれぞれの解釈となる為である。

二つのメッセージデータの計算論的識別不可能性

- [[M]]$_\Pi$を以下のように定義される確率分布の列とする。

$$[[M]]_\Pi = \{[[M]]_{\Pi\eta} \mid \eta \in \mathbf{Parameter}\}$$

- 以下が成り立つとき、MとNがΠで解釈された結果の確率分布の列は計算論的に識別不可能であるという。

$$[[M]]_\Pi \approx [[N]]_\Pi$$

- つまり [[M]]$_\Pi$ と [[N]]$_\Pi$ の差を識別できるPPTは存在しない、という意味である。

（計算論的）識別不可能性

定義: D = {D$_\eta$}, D’ = {D’$_\eta$} を確率分布の列（アンサンブル）とする。以下を満たすとき、DとD’は（計算論的）識別不可能であるといい、$D \approx D'$ と記述する。

- 任意のPPT Aに対して、以下の関数 ε が無視できる。

$$\varepsilon(\eta) \stackrel{\text{def}}{=} \Pr\left[x \xleftarrow{R} D_\eta : A(\eta, x) = 1\right] - \Pr\left[x \xleftarrow{R} D'_\eta : A(\eta, x) = 1\right]$$

- 直感的には、Aは入力xをもらって、xがD$_\eta$から来たものか、D’$_\eta$から来たものか、識別しようとするPPTである。$\varepsilon(\eta)$はAが識別できる確率を表している。ところが、これが無視できるため、識別不可能となる。多項式回、試行を増やしても識別できる確率はほとんど0になる。

健全性

定理: MとNをサイクルが無いメッセージ、Πがtype-0安全な暗号スキーマであるとする。このとき、

$$M \cong N \quad \Rightarrow \quad [\![M]\!]_{\Pi} \approx [\![N]\!]_{\Pi}$$

暗号のサイクル

・ サイクルがあるメッセージの例

$$\{K\}_K \quad (\{K\}_{K'},\{K'\}_K)$$

定義

$\{\ldots K\ldots\}_{K'}$ のとき $K' \to K$ とする。このとき、M中の暗号文でこのように作成されるグラフにサイクルが無いとき、Mにはサイクルが無いという。

メッセージにサイクルがある場合、記号論に基づく意味解釈と計算論に基づく意味解釈にずれが生じるという議論がある。

- 記号論に基づくモデルでは、鍵が漏洩しない限り暗号文が解かれることは無い。
- 一方、計算論に基づくモデルでは、暗号文が解かれてしまう等の現象が起こりうる。

・ サイクルがある場合を考慮した研究

[P.Laud2002],[P.Adao,B.Bana,J.Herzog,A.Scedrov2005]

健全性の証明

- 概略
  - 対偶を示す。つまり、

    $M \cong N \quad \wedge \quad \neg([\![M]\!]_{\Pi} \approx [\![N]\!]_{\Pi})$

    を仮定し、Πがtype-0安全でないことを示す。

    具体的には、

    [[M]]Πと[[N]]Πの差を識別するPPT Aを用いて、Πがtype-0安全でない証拠となるPPT A_0を構成することにより示す。

鍵の名前の変更

- 例を用いて説明する。MとNを以下とする。

$$M = \{0\}_{K_6}\ \{K_1\ 1\}_{K_4}\ K_2\ \{0\}_{K_3}\ \{K_6\}_{K_4}\ \{K_1\ K_3\}_{K_4}\ \{1\ 1\ 1\}_{K_5}\ 0\ \{K_1\}_{K_6}\ \{K_5\}_{K_2}$$

$$N = \{1\ 1\}_{K_2}\ \{K_3\}_{K_2}\ K_1\ \{K_3\}_{K_2}\ \{K_8\}_{K_2}\ \{1\}_{K_5}\ \{1\ 1\ 1\}_{K_3}\ 0\ \{0\ 0\}_{K_8}\ \{K_3\}_{K_1}$$

- 鍵の名前の変更し、MとNに使われていて、取り出すことができる鍵の名前をそれぞれ合わせる。(J_iとする）
- 取り出せない鍵の名前については、Ki→Kjのときi>jとする。

$$M' = \{0\}_{K_3}\ \{K_1\ 1\}_{K_4}\ J_1\ \{0\}_{K_2}\ \{K_3\}_{K_4}\ \{K_1\ K_2\}_{K_4}\ \{1\ 1\ 1\}_{J_2}\ 0\ \{K_1\}_{K_3}\ \{J_2\}_{J_1}$$

$$N' = \{1\ 1\}_{K_3}\ \{J_2\}_{K_3}\ J_1\ \{J_2\}_{K_3}\ \{K_2\}_{K_3}\ \{1\}_{K_1}\ \{1\ 1\ 1\}_{J_2}\ 0\ \{0\ 0\}_{K_2}\ \{J_2\}_{J_1}$$

- pattern(M')=pattern(N')となる。

- 以下のpatternの列ができる。

M'

||

M_4: $\{0\}_{K_3}$ $\{K_1\,1\}_{K_4}$ J_1 $\{0\}_{K_2}$ $\{K_3\}_{K_4}$ $\{K_1\,K_2\}_{K_4}$ $\{111\}_{J_2}$ 0 $\{K_1\}_{K_3}$ $\{J_2\}_{J_1}$

M_3: $\{0\}_{K_3}$ □ J_1 $\{0\}_{K_2}$ □ □ $\{111\}_{J_2}$ 0 $\{K_1\}_{K_3}$ $\{J_2\}_{J_1}$

M_2: □ □ J_1 $\{0\}_{K_2}$ □ □ $\{111\}_{J_2}$ 0 □ $\{J_2\}_{J_1}$

M_1: □ □ J_1 □ □ □ $\{111\}_{J_2}$ 0 □ $\{J_2\}_{J_1}$

M_0: □ □ J_1 □ □ □ $\{111\}_{J_2}$ 0 □ $\{J_2\}_{J_1}$

||

N_0: □ □ J_1 □ □ □ $\{111\}_{J_2}$ 0 □ $\{J_2\}_{J_1}$

N_1: □ □ J_1 □ □ $\{1\}_{K_1}$ $\{111\}_{J_2}$ 0 □ $\{J_2\}_{J_1}$

N_2: □ □ J_1 □ □ $\{1\}_{K_1}$ $\{111\}_{J_2}$ 0 $\{00\}_{K_2}$ $\{J_2\}_{J_1}$

N_3: $\{11\}_{K_3}$ $\{J_2\}_{K_3}$ J_1 $\{J_2\}_{K_3}$ $\{K_2\}_{K_3}$ $\{1\}_{K_1}$ $\{111\}_{J_2}$ 0 $\{00\}_{K_2}$ $\{J_2\}_{J_1}$

||

N'

ここで $M_i = p(M',\{K \in \mathbf{Keys} \mid M' \vdash K\} \cup \{K_1,\ldots,K_i\})$

$N_i = p(N',\{K \in \mathbf{Keys} \mid N' \vdash K\} \cup \{K_1,\ldots,K_j\})$

Hybrid argumentの利用

- $[\![M]\!]_{\Pi} \approx [\![N]\!]_{\Pi}$ でないと仮定する。
- すると、あるiで　$[\![M_i]\!]_{\Pi} \approx [\![M_{i-1}]\!]_{\Pi}$　でない。なぜなら、無視できない関数を定数で割っても、やはり、無視できないため。(hybrid argument)

- このとき、Πはtype-0安全でないことを示す。

pattern の解釈

- Patternで用いられる□にも解釈を定義する。
  - Initialize に、以下を加える。

$$\tau(K_0) \xleftarrow{R} \mathcal{K}(\eta)$$

  - Convert に以下を加える。

if $M = \square$ **then**

$$y \xleftarrow{R} \mathcal{E}_{\tau(K_0)}(\mathbf{0})$$

return $\langle y, \text{"ciphertext"} \rangle$

- つまり、復号できないことを表す記号□用に鍵を生成し、それで0を暗号化したデータと解釈する。

- $[\![M_i]\!]_{\Pi} \approx [\![M_{i-1}]\!]_{\Pi}$ でない つまり、
  あるPPT Aで$p_i(\eta)$-$p_{i-1}(\eta)$が無視できない。

$$p_i(\eta) = \Pr\left[y \xleftarrow{R} [\![M_i]\!]_{\Pi[\eta]} : A(\eta, y) = 1 \right]$$

$$p_{i-1}(\eta) = \Pr\left[y \xleftarrow{R} [\![M_{i-1}]\!]_{\Pi[\eta]} : A(\eta, y) = 1 \right]$$

- このとき、次のPPT A_0で以下が成り立つことを示せば、Πがtype-0安全でないことが導ける。

$$\Pr\left[k_i, k_0 \xleftarrow{R} \mathcal{K}(\eta) : A_0^{\mathcal{E}_{k_i}(\cdot), \mathcal{E}_{k_0}(\cdot)}(\eta) = 1 \right] = p_i(\eta)$$

$$\Pr\left[k_0 \xleftarrow{R} \mathcal{K}(\eta) : A_0^{\mathcal{E}_{k_0}(\mathbf{0}), \mathcal{E}_{k_0}(\mathbf{0})}(\eta) = 1 \right] = p_{i-1}(\eta)$$

・ A_0は次の通り

algorithm $A_0^{f,g}(\eta)$

for $K \in Keys(M')$ **do** $\tau(K) \xleftarrow{R} \mathcal{K}(\eta)$

$y \xleftarrow{R} \text{CONVERT2}(M')$

$b \xleftarrow{R} A(\eta, y)$

return b

algorithm CONVERT2(M^*)

if $M^* = K$ where $K \in$ **Keys then**

return $\langle \tau(K), \text{"key"} \rangle$

if $M^* = b$ where $b \in$ **Bool then**

return $\langle b, \text{"bool"} \rangle$

if $M^* = (M_1^*, M_2^*)$ **then**

return $\langle \text{CONVERT2}(M_1^*), \text{CONVERT2}(M_2^*), \text{"pair"} \rangle$

if $M^* = \{M_1^*\}_K$ **then**

if $K \in \{J_1, \ldots, J_\mu, K_1, \ldots K_{i-1}\}$ **then**

$x \xleftarrow{R} \text{CONVERT2}(M_1^*)$

$y \xleftarrow{R} \mathcal{E}_{\tau(K)}(x)$

return $\langle y, \text{"ciphertext"} \rangle$

else if $K = K_i$ **then**

$x \xleftarrow{R} \text{CONVERT2}(M_1^*)$

$y \xleftarrow{R} f(x)$

return $\langle y, \text{"ciphertext"} \rangle$

else if $K \in \{K_{i+1}, \ldots K_m\}$ **then**

$y \xleftarrow{R} g(\mathbf{0})$

return $\langle y, \text{"ciphertext"} \rangle$

- M_iとMi-1の違いは、M_iのなかのKiの暗号文が、Mi-1ではundecryptableになっている。それ以外は同じ。

$$M_i: \quad \ldots\{\ldots\}_{K_i} \ldots$$
$$M_{i-1}: \quad \ldots \ [] \ \ldots$$

- このとき、$A_0^{\mathcal{E}_{k_i}(\cdot),\mathcal{E}_{k_0}(\cdot)}(\eta)$ 中のconvert2は$\{\ldots\}_{K_i}$に対して正しくkiで暗号化したデータを返す。つまり convert2(M')は[[Mi]]Π_ηのデータを返す。
- 一方、$A_0^{\mathcal{E}_{k_0}(\mathbf{0}),\mathcal{E}_{k_0}(\mathbf{0})}(\eta)$ 中のconvert2は、□に対応するように、k0で暗号化したデータを返す。つまり convert2(M')は[[Mi-1]]Π_ηのデータを返す。
- よって、

$$\Pr\left[y \xleftarrow{R} [\![M_i]\!]_{\Pi[\eta]} : A(\eta, y) = 1\right] = \Pr\left[k_i, k_0 \xleftarrow{R} \mathcal{K}(\eta) : A_0^{\mathcal{E}_{k_i}(\cdot),\mathcal{E}_{k_0}(\cdot)}(\eta) = 1\right]$$

$$\Pr\left[y \xleftarrow{R} [\![M_{i-1}]\!]_{\Pi[\eta]} : A(\eta, y) = 1\right] = \Pr\left[k_0 \xleftarrow{R} \mathcal{K}(\eta) : A_0^{\mathcal{E}_{k_0}(\mathbf{0}),\mathcal{E}_{k_0}(\mathbf{0})}(\eta) = 1\right]$$

- よって健全性が証明された。

完全性

定理: E_0とE_1がサイクルが無いメッセージ、Πがtype-0安全で、Confusion freeな暗号スキーマであるとする。このとき、

$$[\![E_0]\!]_\Pi \approx [\![E_1]\!]_\Pi \quad \Rightarrow \quad E_0 \cong E_1$$

重要：ここでは、データを見てそのタイプ（鍵データなのか、暗号データなのか、連結データなのか）が分かると仮定している。

Confusion free

- D={D1,D2,…,Dl} を確率分布の列の有限集合とする。以下を満たす無視できる関数$\nu_D(\eta)$が存在するとき、∏はDに対してconfusion freeであるという。
  - 任意のiで

$$\Pr\left[k_1,k_2 \xleftarrow{R} \mathcal{K}(\eta), x \xleftarrow{R} D_i(\eta) : \mathcal{D}_{k_1}(\mathcal{E}_{k_2}(x)) \neq \mathbf{0}\right] \leq \nu_D(\eta)$$

- 直感的には、暗号データが衝突する確率は無視できる、という意味。

完全性の証明

概略

- 対偶を証明する。つまり、以下を証明する。
  - E_0とE_1がサイクルが無いメッセージで、$E_0 \cong E_1$でないとする。さらにΠがtype-0安全で、Confusion freeな暗号スキーマであるとする。このとき、二つの確率分布の列$[[E_0]]_\Pi$と$[[E_1]]_\Pi$を無視できない確率で区別するPPTが存在する。

- $e \leftarrow [[E]]_{\Pi\eta}$ からEのパターンを計算し、E_0のパターンと比較するPPTを考えればよい。

鍵データを集めるアルゴリズムCkr

eから鍵データを集めるアルゴリズムC_{kr}

Algorithm $C_{kr}(e,tT)$

if $e = \langle b,"block"\rangle$ output tT

if $e = \langle k,"key"\rangle$ output $tT \cup \{k\}$

if $e = \langle (m,n),"pair"\rangle$ output $C_{kr}(m,tT) \cup C_{kr}(n,tT)$

if $e = \langle c,"ciphertext"\rangle$ then

if for exactly one $k \in tT, \mathcal{D}_k(y) \neq \perp$

then output $C_{kr}(\mathcal{D}_k(c),tT)$

else output tT

c.f. 記号論に基づく意味での
アルゴリズム

1. $F_{kr}(B,T) = T$;
2. $F_{kr}(K,T) = \{K\} \cup T$;
3. $F_{kr}((E_0,E_1),T) = F_{kr}(E_0,T) \cup F_{kr}(E_1,T)$;
4. $F_{kr}(\{E\}_K,T) = T$, if $K \notin T$;
5. $F_{kr}(\{E\}_K,T) = F_{kr}(E,T)$, if $K \in T$;

- Ckrでは復号化に失敗しない鍵データがたった一つのときのみ、中身を調べる。
- 複数の鍵データで復号に成功する場合は、この二つのアルゴリズムの結果にずれが出てくる。これは暗号データの衝突が起こった場合であるが、この確率は無視できる。(confusion free より)

鍵データをすべて集めるアルゴリズム Crecoverable

- eから取り出せる鍵データをすべて集めるアルゴリズムCrecoverable

Algorithm Crecoverable(e)

$T_0 \leftarrow \emptyset$; $cont \leftarrow true$

while $cont$

$T_{i+1} \leftarrow \mathsf{C}_{\mathsf{kr}}(e, T_i)$

if $T_{i+1} = T_i$ then $cont \leftarrow false$

else $i \leftarrow i + 1$

output T_i

c.f. 記号論に基づく意味でのアルゴリズム

- $G_0(E) = \emptyset$;
- $G_i(E) = \mathsf{F}_{\mathsf{kr}}(E, G_{i-1}(E))$
- $\mathsf{recoverable}(E) = \bigcup_i G_i(E) = G_{\mathsf{lfp}}(E)$

補題: Πがconfusion freeな暗号スキーム、Eがメッセージ、Tが鍵記号の集合であるとする。e←[[E]]$_{\Pi\eta,\tau}$が鍵記号から鍵データへの解釈であるとき、以下を満たす無視できる関数νが存在する。

$$\Pr\left[\mathsf{C}_{kr}(e,\tau(T)) \neq \{\tau(\mathsf{F}_{kr}(E,T))\}\right] \leq |E||T|\nu(\eta)$$

$$\Pr\left[\mathsf{Crecoverable}(e) \neq \tau(\mathsf{recoverable}(E))\right] \leq |E|^3\,\nu(\eta)$$

- メッセージから鍵を取り出すアルゴリズムが、記号的なアルゴリズムとずれる確率は無視できる。

Patternを計算するアルゴリズムpsp

Algorithm psp(e,tT)

if $e = \langle b,"block"\rangle$ output block symbol B corresponding to b;

if $e = \langle k,"key"\rangle$ and $k \in tT$ then output $f(k)$;

if $e = \langle (m,n),"pair"\rangle$ output (psp(m,tT),psp(n,tT));

if $e = \langle c,"ciphertext"\rangle$

if for exactly one key $k \in tT, \mathcal{D}_k(c) \neq \perp$

output $\{\text{psp}(\mathcal{D}_k(c),tT)\}_{f(k)}$

else output □ ;

c.f. 記号論に基づく意味でのアルゴリズム

$\text{p}(K,T) = K$

$\text{p}(B,T) = B$

$\text{p}((M,N),T) = (\text{p}(M,T),\text{p}(N,T))$

$\text{p}(\{M\}_K,T) = \{\text{p}(M,T)\}_K$ (if $K \in T$)

$\text{p}(\{M\}_K,T) =$ □ (if $K \notin T$)

- C_{kr}での場合と同様、pspは復号化に失敗しない鍵データがたった一つのときのみ、中身を調べる。
- 複数の鍵データで復号に成功する場合は、この二つのアルゴリズムの結果にずれが出てくる。これは暗号データの衝突が起こった場合であるが、この確率は無視できる。(confusion free)

補題: Πがconfusion freeな暗号スキーム、Eがメッセージ、Tが鍵記号の集合であるとする。e←[[E]]$_{\Pi\eta,\tau}$が鍵記号から鍵データへの解釈であるとき、以下を満たす無視できる関数νが存在する

$$\Pr\left[\neg\ \mathrm{psp}(e,\tau(T)) \cong \mathrm{p}(E,T)\right] \leq | E | | T | \nu(\eta)$$

系:

$$\Pr\left[tT \leftarrow \mathrm{Crecoverable}(e) : \neg \mathrm{pattern}(E) \cong \mathrm{psp}(e,tT)\right] \leq 2 | E |^3 \nu(\eta)$$

EとE'のpatternが異なるとき、

$$\Pr\left[tT \leftarrow \mathrm{Crecoverable}(e) : \neg \mathrm{pattern}(E') \cong \mathrm{psp}(e,tT)\right] \geq 1 - 2 | E |^3 \nu(\eta)$$

完全性

- [[E_0]]$_\Pi$と[[E_1]]$_\Pi$の違いを識別する次のようなアルゴリズムDを考える。
  1. 最初にpsp(e,Crecoverable(e))でパターンを計算。
  2. それをE_0のパターンと比較。
  3. もし、同じパターンだったら0、そうでなければ1を出力。
- このとき、以下のようにDは無視できない確率で（圧倒的な確率で）[[E_0]]$_\Pi$と[[E_1]]$_\Pi$の違いを識別する。またDは多項式時間で計算できる。

$$\Pr\left[e \xleftarrow{R} \llbracket E_0 \rrbracket_{\Pi(\eta)} : D(e,\eta) = 0\right] - \Pr\left[e \xleftarrow{R} \llbracket E_1 \rrbracket_{\Pi(\eta)} : D(e,\eta) = 0\right] \geq$$

$$(1 - 2 \mid E_0 \mid^3 \nu(\eta)) - 2 \mid E_1 \mid^3 \nu(\eta) = 1 - 2(\mid E_0 \mid^3 + \mid E_1 \mid^3)\nu(\eta)$$

- よって、完全性が示された。

他の安全性を仮定した場合

- 一般的に、undecryptableの記号□は、その暗号文から取り出せる情報を表すと見なせる。
  - Type-0安全では、如何なる情報も取り出せないので、□を用いている。
  - Type-1安全（内容データの長さのみ取り出せる）では、 $\square_n$（nは内容データの長さ）
  - Type-2安全（鍵の情報のみ取り出せる）では$\square_k$
  - □につけるべき情報について一般的な形式化 [P.Adao,G.Bana,A.Scedrov2005]

並行システムへの拡張 [M.Abadi,J.Jurjens2001]

- 暗号データを取り扱う複数のプログラムが並行して動作するシステムを考える。
  - チャネルが複数存在し、各々のチャネルに対して、そのチャネルに出力するプログラムが存在するとする。
  - プログラムの構文はシンプルで、spi計算に類似している。
    - データの出力
    - 暗号化、復号化、連結、分解
    - 条件分岐など

プログラムの構文

- 構文と例

$P,Q ::=$	programs
ε	null value
K	key ($K \in \mathbf{Keys}$)
b	bit ($b \in \mathbf{Bool}$)
$\perp$	error
a	input on channel ($a \in \mathbf{Channels}$)
x	variable ($x \in \mathbf{Var}$)
(P,Q)	pair
$(\nu K)P$	"new" ($K \in \mathbf{Keys}$)
$(\nu r)\{P\}^r_Q$	shared-key encryption ($r \in \mathbf{Coins}$)
$if\ P = Q\ then\ P'\ else\ Q'$	conditional
$case\ P\ of\ (x,y)\ do\ Q\ else\ R$	case: pair ($x,y \in \mathbf{Var}$)
$case\ P\ of\ \{x\}_{P'}\ do\ Q\ else\ R$	case: encryption ($x \in \mathbf{Var}$)

$$c_0 := \{0\}_{K_1}$$

$$c_1 := case\ c_0\ of\ \{x\}_{K_0}\ do\ (0,x)\ else\ case\ c_0\ of\ \{x\}_{K_1}\ do\ (1,x)\ else\ \varepsilon$$

システムに対する２種類の意味論

- このシステムに対して２種類の意味論を与えている。
  - 記号論に基づいた意味論
    - プログラムに記号論に基づいた操作的意味論を与え、システムの実行トレースを定める。
    - システムの等価性を、その実行トレースから得られる情報（パターン）が等しいという等価性と解釈する。
  - 計算論に基づいた意味論
    - プログラムに計算論に基づいた操作的意味論を与え、システムの実行トレースを定める。
    - システムの等価性は、その実行トレースに対する、PPTによる計算論的識別不可能性と解釈する。

健全性

定理: PとQは、暗号化サイクルを生成しない、システムとする。Πはtype-0安全でconfusion-freeな暗号スキームであるとする。このとき、

PとQが記号論に基づく意味論で等価ならば、

PとQは計算論に基づく意味論でも等価である。

まとめ

- 記号論で解読できないメッセージ$\{M\}_k$に対するデータと□に対するデータを識別できるかどうかはtype-0安全かどうかに、ほぼ直接的に対応している。
- 健全性においては、定義に必要な道具立ては仰々しいが、証明の核心部分はほぼ直接対応付けが可能である。
- 完全性においては、計算論の意味論における
  - confusion-freeという条件
  - メッセージデータを見て必ずデータのタイプがわかるという仮定

  が大きな役割を果たしている。これらはほぼ記号論の意味論に対応させている。