暗号理論・数学の基礎

光成滋生

last update: 2025/10/20

概要

目的

- 漠然とした言葉による定義をどうやって数学的な定式化に落としこむかを理解する
- 暗号理論でよく使う数学的な道具を理解する
- 暗号文から平文の情報が漏れないとは

構成

- 収束, O記法, 無視できる関数, PPT
- 疑似乱数, 共通鍵暗号, IND-CPA, PRF
- 頑強性

定式化

あいまいな表現

- 真の乱数と区別がつかない
- ほとんど同じ
- n が十分大きい
- 平文の情報がもれない

無限大をどう扱うか

- $f(x)$ が $x \to \infty$ のとき a に収束するとは
- $f(x) = 2 + 3/x$ のとき $x \to \infty$ で $f(x) \to 2$
  - $f(10) = 2.3, f(1000) = 2.003, f(100000000) = 2.00000003$
  - 「x を大きくすれば2に近づく」ではなく
    「収束先 a が2でなければ x を大きくすると a よりも2に近くできる」と定義する

収束の定義

記法

- $\exists a\sim$: 存在記号: 「〜となる a が存在して...」
- $\forall a\sim$: 全称記号: 「〜となる全ての a に対して...」
- s.t.〜: such that: 「~となるような...」

定義

- $f(x)$ が $x \to \infty$ のとき a に収束するとは
  $\forall \varepsilon > 0, \exists N > 0$ s.t. $\forall x > N, \left|f(x) - a\right| < \varepsilon.$
- 読み方: どんな（小さな） $\varepsilon > 0$ に対しても、それに対応して
  「$x > N$ ならば $\left|f(x) - a\right| < \varepsilon$ となる」ような $N > 0$ を選べる
- どんなに小さい（0でない）誤差 ε を持ってきても、十分大きな N を選ぶと
  それより大きい x は $\left|f(x) - a\right| < \varepsilon$ とできる
- 順序が大事: 「$\exists N > 0 s.t. \forall \varepsilon > 0, \forall x > N, \left|f(x) - a\right| < \varepsilon$」ではない
  - 問: これだとどうなるか?

具体例

$f(x) = 2 + 3/x \to 2$ for $x \to \infty$

- $a = 2$
- $\varepsilon > 0$ を選ぶ. $|f(x) - a| = |2 + 3/x - a| = |3/x| < \varepsilon$ となるためには $x > 3/\varepsilon$ とすればよい
- つまり $N = 3/\varepsilon$ とすれば $x > N$ ならば $|f(x) - a| < \varepsilon$ である

例

- $\varepsilon = 0.000001$ とする
- $N = 3/\varepsilon = 3000000$ とすれば $x > N$ ならば $|f(x) - a| < \varepsilon$ である

問題

- $f(x) = 3 + 4/x^2$ のとき $x \to \infty$ で $f(x) \to 3$ を示せ

O記法の定義

定義

- $f(n) = O(g(n))$ とは $\exists c > 0, \exists N > 0$ s.t. $\forall n > N, |f(n)/g(n)| \leq c.$

読み方

- ある $c > 0$ が存在し、「$n > N$ となる全ての n に対して $|f(n)/g(n)| \leq c$ が成り立つ」ような $N > 0$ が存在する

意味するところ

- まず定数 c が固定して決まり、その c に対して次の条件を満たす定数 N を決められる
  - $n > N$ ならば $|f(n)/g(n)| \leq c$ である
- $n \leq N$ のときは例外があるかもしれないけど十分大きく $(n > N)$ とれば c 以下にできる

例

$$f(n) = 3n^2 + 5n + 4 = O(n^2)$$

直感的には

- $f(n)/n^2 = 3 + 5/n + 4/n^2 \to 3 (n \to \infty)$ だから $f(n) = O(n^2)$

定義にしたがって確認すると

- $c = 4$ とする（3より大きな値を適当に決めた）
- $5/n + 4/n^2 < 1$ なる n を考える
- $n \geq 10$ なら $5/n + 4/n^2 \leq 5/10 + 4/100 = 1/2 + 1/25 < 1$
- よって $N = 10$ とすると $n \geq N$ なら $f(n)/n^2 \leq 3 + 1 = c$ である（QED）
  - もちろんもっと厳密に c や N を求めてもよいが存在を示せばよいので大雑把でもよい

問題

簡単

1. $f(n)$ が $n \to \infty$ で収束しないが $O(1)$ である例を示せ

中級

2. どんな $k > 0$ についても $e^n = O(n^k)$ とはならないことを示せ

- $e^n = \sum_{i=0}^{\infty} \frac{n^i}{i!}$ であることを利用してよい

無視できる関数

十分小さいことを表す

- $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ が無視できる (negligible) とは $\forall c > 0, \exists N \in \mathbb{N}$ s.t. $\forall n > N, f(n) < n^{-c}$.
  - $c > 0$ をどんなに大きくとっても、それに対応して十分大きく N を選べば、
    それより大きい $n > N$ について $f(n) < n^{-c}$ とできる
- $f(n)$ はどんな多項式の逆数よりも速く小さくなる
- このとき $f(n) < \mathtt{negl}(n)$ と書く（$f(n) < \varepsilon(n)$ と書くこともある）

例

- $e^{-n} = \mathtt{negl}(n)$
- e^n はどんな多項式よりも速く大きくなるのでその逆数はとても速く小さくなる
  - 任意の $c > 0$ に対して $k = \mathtt{ceil}(c) + 1$ とすれば $e^n > n^k/k!$.
  - $N = k!$ とすれば $n > N$ なら $e^{-n} < n^{-k}k! = n^{-k+1}(k!/n) < n^{-(k-1)} \leq n^{-c}$.

確率的多項式時間アルゴリズム

決定的 (deterministic) 多項式時間 PT (Polynomial Time) アルゴリズム

- n bitの入力 $x \in \{0,1\}^n$ に対して常に同じ値 $f(x)$ を返す
- 実行時間が n に関する多項式時間

確率的 (probabilistic) 多項式時間 PPT アルゴリズム

- n bitの入力 x に対して乱数 r を取得し $f(x,r)$ を返す（r によって値が変わり得る）
- 実行時間が n に関する多項式時間

n ビットの整数 x について

- x が素数であるかどうかを判定する n についての決定的PTアルゴリズムがある（非効率）
- x が合成数か、判定不能かを返す効率のよいPPTアルゴリズムが存在する
  - 繰り返し適用してずっと判定不能なら多分素数
  - 十分小さい確率を無視して x が素数であるとみなせる

疑似乱数

一様ランダム

- 集合 S からどの要素も同じ確率で取り出すとき $x \xleftarrow[U]{} S$ と書く

理想の乱数と疑似乱数の違いは何か?

- 小さい値 (seed) から決定的アルゴリズムで長い疑似乱数列を生成したとき、
  それが本物の乱数と見分けがつかない
- 見分ける人の能力を考える
  - 攻撃者も通常の計算機を使うと考えてPPTアルゴリズムを実行すると仮定する
- 「見分けがつかない」を無視できる関数で表現する

疑似乱数生成器 PRG の定義

$G : \{0,1\}^n \to \{0,1\}^{l(n)}$（$l(n) > n$ な多項式）がPRGであるとは次を満たすもの

- $\forall \mathcal{A}(x) \in \{0,1\}$: PPT多項式について

$$\left| \Pr\left[\mathcal{A}(r) = 1 \mid r \underset{U}{\leftarrow} \{0,1\}^{l(n)}\right] - \Pr\left[\mathcal{A}(G(s)) = 1 \mid s \underset{U}{\leftarrow} \{0,1\}^{n}\right] \right| < \mathtt{negl}(n)$$

意味

- $\mathcal{A}(x)$ は x が乱数と判定すれば1, そうでなければ0を返す攻撃者（判定者）
  - $l(n)$ bitの真の乱数 r を選ぶ
  - n bitの真の乱数 s から $l(n)(> n)$ bitの疑似乱数 $G(s)$ を作る
  - 両者をそれぞれ $\mathcal{A}$ に渡して乱数と判定する確率の差が無視できる
- どんな判定者も区別できないなら、その疑似乱数は真の乱数とみなしてよいだろう
  - 計算量的識別不可能な疑似乱数
- （注意）PRG が存在するかは未解決問題（少なくとも $P \subsetneq NP$ が必要）

共通鍵暗号の定義

セキュリティパラメータ λ

- 暗号システムの安全性を表すパラメータ. k や他の記号を使うことも多い
  - 1^λ は「1の λ 乗」ではなく「1か λ 個ならんだ文字列」を表す
- アルゴリズム $\mathcal{A}$ について $\mathcal{A}(1^\lambda)$ は $\mathcal{A}$ が λ に関する（確率的）多項式時間アルゴリズムであることを示す（1^λ を省略することも多い）

$\Pi = (KeyGen, Enc, Dec)$ が共通鍵暗号であるとは

- $\mathcal{K}$: 鍵空間, $\mathcal{M}$: 平文空間, $\mathcal{C}$: 暗号文空間
- $KeyGen(1^\lambda) = s \xleftarrow[R]{} \mathcal{K}$: 鍵生成（$\mathcal{K}$ からランダムサンプリング）
- $Enc : \mathcal{K} \times \mathcal{M} \ni (s, m) \mapsto c \in \mathcal{C}$: （確率的）暗号化アルゴリズム
- $Dec : \mathcal{K} \times \mathcal{C} \ni (s, c) \mapsto m \in \mathcal{M}$: （決定的）復号アルゴリズム
  で、$\forall s \in \mathcal{K}, \forall m \in \mathcal{M}$ について $Dec(s, Enc(s, m)) = m$ を満たす

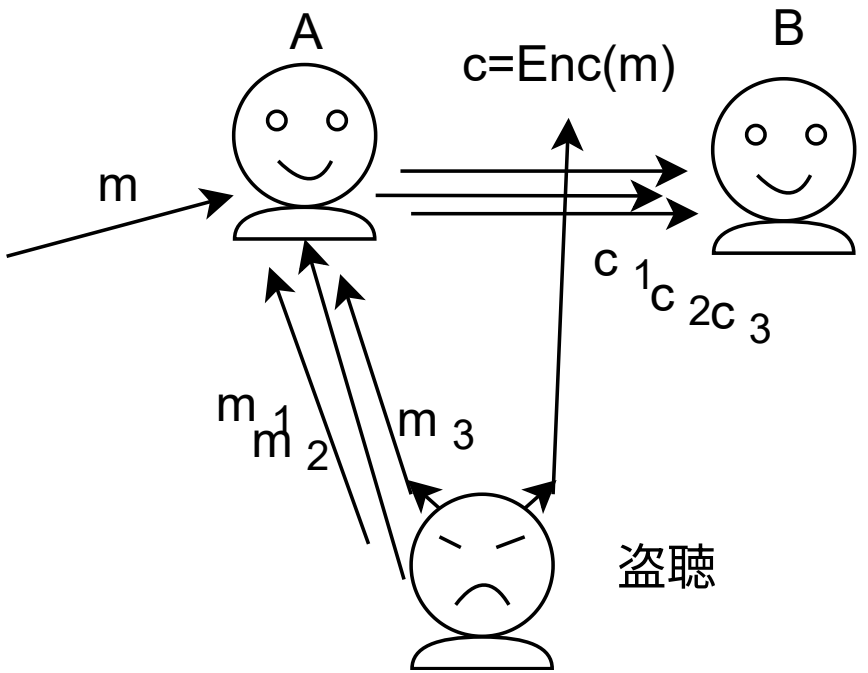
攻撃の種類

一番基本的な状況

- 攻撃対象の暗号文 c しか得られない

選択平文攻撃 CPA (Chosen Plaintext Attack)

- c を受け取る前後で攻撃者は自分で選んだ平文に対応する暗号文を得られる状況
  - そんな都合のよいことがあるのか?
- 例えば
  - AはwebストアでBは仕入れ先業者
  - 攻撃者はAとBの通信を盗聴
  - $c = Enc(m)$ に関する m の情報を得たい
  - 攻撃者がAに発注 $m_1, \ldots$ して暗号文 $c_1, \ldots$ を入手
- こんなときでも m の情報が漏れないで欲しい



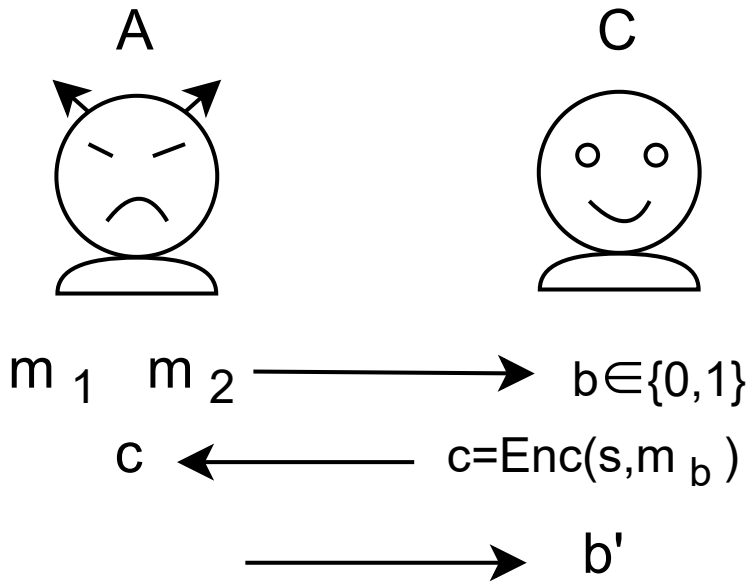
IND-CPA安全

情報が漏れない（強秘匿性）とは

- 自分で選んだ平文 m_1, m_2 のどちらかの暗号文 c をもらってもどちらの平文か当てられない

平文当てゲーム $\mathrm{Exp}(\lambda)$: 実験 (experiment)

1. 挑戦者 C (Challenger): $s = KeyGen(1^\lambda)$
2. 攻撃者 A (Adversary): $m_1, m_2 \in \mathcal{M}$ を選ぶ
3. C: $b \in \{0,1\}$ を選び $c = Enc(s, m_b)$ を A に送る
4. A: c から $b' \in \{0,1\}$ を推測して出力する



- $b = b'$ なら A の勝ち(1): 適当に答えても当たる確率は1/2

Aの優位度 (Advantage) が無視できる＝情報が漏れてない＝IND-CPA安全

- $Adv(\lambda) := \left|\Pr\left[Exp(\lambda) = 1\right] - \frac{1}{2}\right| < \mathtt{negl}(\lambda)$ for $\forall$ PPT Algo Exp.
  このとき暗号は強秘匿性を持つという

疑似ランダム関数 PRF (Pseudo-Random Function)

ランダム関数

- $RF_{n,m} = \{f \mid f : \{0,1\}^n \to \{0,1\}^m\}$: n bitの入力から m bitを出力する関数全体
  $f \underset{U}{\leftarrow} RF_{n,m}$ をランダム関数という（$\left|RF_{n,m}\right| = \left(2^m\right)^{2^n}$）

$F : \{0,1\}^n \times \{0,1\}^{in} \to \{0,1\}^{out}$**: PT関数,** in, out **は** n **の多項式**

- $s \in \{0,1\}^n, F_s(x) := F(s,x)$ が PRF とは
  入力が疑似乱数なら1を返す $\forall$ PPT algo. $\mathcal{A}$,
  $$\left|\Pr\left[\mathcal{A}(F_s) = 1 \mid s \underset{U}{\leftarrow} \{0,1\}^n\right] - \Pr\left[\mathcal{A}(f) = 1 \mid f \underset{U}{\leftarrow} RF_{in,out}\right]\right| < \mathtt{negl}(n)$$
  となることをいう
- ここで $\mathcal{A}(F_s)$ は $\mathcal{A}$ が必要なだけ好きな x を選んで $F_s(x)$ の結果を得られることを意味する
- $\mathcal{A}(f)$ も同様
- $\left|\{0,1\}^n\right| = 2^n$ なので $RF_{in,out}$ よりずっと小さいけれども区別できないということ

PRFを用いたIND-CPA安全な暗号の構成

記号の準備

- F: PRF, $\mathcal{K} = \{0,1\}^n, \mathcal{M} = \{0,1\}^{out}, \mathcal{C} = \{0,1\}^{in} \times \{0,1\}^{out}$
- $KeyGen(1^n) = s \xleftarrow[U]{} \{0,1\}^n$
- $Enc(s, m) = (r, F_s(r) \oplus m)$ for $r \xleftarrow[U]{} \{0,1\}^{in}$
- $Dec(s, c) = F_s(r) \oplus c$
  としたとき$\Pi = (KeyGen, Enc, Dec)$ はIND-CPA安全な共通鍵暗号である

頑強性 (non-malleability) の定義

平文を制御できるような暗号文を作れない

- $c = Enc(m)$ に対して m と関係のある別の m^* に対応する $c^* = Enc(m^*)$ を作れない
- 攻撃者 $\mathcal{A}$ が作った暗号文 c^* の復号結果を教えても元の m の情報を得られない

平文当てゲーム $\mathbf{Exp}_b(\lambda)$

- $k = KeyGen(1^\lambda), (m_0, m_1) \leftarrow \mathcal{A}_1()$ # 平文を2個選ぶ
- $c^* \leftarrow Enc(m_b)$ # 片方の暗号文をもらう
- $(c_1, \ldots, c_n) \leftarrow \mathcal{A}_2(c^*)$ # c^* から何か正しい暗号文($c^* \neq c_i$)を作る
- $d_i := Dec(c_i)$ # その暗号文の結果を教えてもらう
- $b' \leftarrow \mathcal{A}_3(d_1, \ldots, d_n), p_b := \Pr[b' = b]$ # その情報から b を当てる

m_b のどちらの暗号文をもらっても b を当てる確率は変わらない

- $\mathrm{Adv}(\lambda) := |p_0 - p_1| < \mathbf{negl}(\lambda)$ for $\forall$ PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ ならば頑強性を持つ
  - 注意: 細かなパラメータは省略している: k は共通鍵暗号, 公開鍵暗号に応じて適切に選ぶ