# テーマ２：公開鍵暗号と署名の安全性

本日の講義の目的

- P $\neq$ NP 予想速成コース
- ランダムオラクル

### 前回までの復習

公開鍵暗号系を用いると，メッセージの暗号化に加えて署名付きメッセージ作成も可能である．さらに，その安全性を議論する枠組みも計算論的暗号理論で用意されている．しかし（たとえ暗号化が安全だと仮定しても）署名の安全性まで議論するのは非常に難しい．では指を加えてながめているだけでよいだろうか？何か論理的に議論・解析する方法はないだろうか？(ここまでが先週の話)

この打開策として考えられたのが「ランダムオラクル」という方便である．これは随分昔に計算量理論で導入された論法である．今回は，有名な「P $\neq$ NP 予想」を題材に，このランダムオラクルについて説明する．

### 2.1. P $\neq$ NP 予想

NP 問題とは，yes/no の判定問題で，yes の場合には，その証拠があり，証拠（の候補）を与えられれば，それが正しい証拠か否かを判定するのが（計算論的に）簡単な問題である．ここでは例で説明する．

**例 2.1.** 先週の話に合わせるため，問題を関数の形で指定しよう．また，関数の定義域は $\{0,1\}^*$ とする．以下では関数値 1 を yes と，0 を no とみなす．まず次の関数を考える．

$$f_{\mathrm{non}}(x) \;=\; \begin{cases} 1, & \text{if } x \text{ が合成数（の二進表記）,} \\ 0, & \text{otherwise.} \end{cases}$$

つまり，与えられた数（二進数）が合成数か否かを判定する問題である．たとえば，$f_{\mathrm{non}}(11001) = 1$, $f_{\mathrm{non}}(111) = 0$, $f_{\mathrm{non}}(1011) = 0$ だ．合成数の場合，yes の証拠は因数である．それを示せば，正しいかどうかのチェックは簡単（＝多項式時間で計算可能）である．たとえば 10101 (= 21) の証拠は，3（もしくは 7）である．

$$f_{\mathrm{ham}}(w) \;=\; \begin{cases} 1, & \text{if } w \text{ で表されるグラフがハミルトン路を持つ,} \\ 0, & \text{otherwise.} \end{cases}$$

ここで入力として与えられる二進列 $w$ は，グラフの隣接行列（辺の有無を行列にしたもの）を横に並べたものと解釈する．また，ハミルトン路 とは，すべての頂点を１度ず

つ通るたどり方のこと．たとえば，$w = 011101110$ は三角形なのでハミルトン路を持つ．したがって $f_{\mathrm{ham}}(01110110) = 1$ である．一方，$f_{\mathrm{ham}}(0111100010001000) = 0$ である．

似たような問題として，オイラー路（＝一筆書き）を持つか否かを判定する問題がある．

$$f_{\mathrm{eu}}(w) \;=\; \begin{cases} 1, & \text{if } w \text{ で表されるグラフがオイラー路を持つ,} \\ 0, & \text{otherwise.} \end{cases}$$

この場合には，$f_{\mathrm{eu}}(0111101011001000) = 1$ となる．その証拠は $(4, 1, 2, 3, 1)$ という頂点の巡り方である．

NP 問題は，証拠をもらえば判定できるので，その証拠を探しさえすればよい．しかし証拠を探すのが容易でない場合もある．

たとえば，上の例では $f_{\mathrm{eu}}$ のための証拠は（存在すれば）簡単に求めることができる．それに対し，$f_{\mathrm{non}}$ の証拠（つまり素因数分解）を求めるのは困難と言われている．ただし，$f_{\mathrm{non}}$ に対しては別種の（探すのが比較的容易な）証拠があることが最近発見されている．したがって，この二つの関数は多項式時間計算可能である．一方，$f_{\mathrm{ham}}$ に対しては，容易に探すことのできる証拠を与えることは不可能であると予想されている．大ざっぱにいえば，その予想が「$\mathrm{P} \neq \mathrm{NP}$ 予想」である．

正確に言うと「NP 問題の中には，入力に対して判定することが多項式時間では不可能な問題存在する」というのが $\mathrm{P} \neq \mathrm{NP}$ 予想である．つまり，ある NP 問題を表す関数 $f$ が存在して，任意の多項式時間アルゴリズム $\mathtt{A}$ に対して，

$$\mathtt{A} \neq f_{\mathrm{ham}}$$

となる，というのが $\mathrm{P} \neq \mathrm{NP}$ 予想である．ちなみに P は多項式時間で計算できる関数の全体で，NP は NP 問題と呼ばれる関数の全体である．

### 2.2. ランダムオラクル

$\mathrm{P} \neq \mathrm{NP}$ 予想の解決が絶望的なので，研究者たちは別の見方で何とか P と NP の性質の違いを見出せないか模索した．その結果，出てきたのが「ランダムオラクル」という考え方である．

そもそも「オラクル」（神託）とは，何らかのお告げをしてくれる「神」のこと．計算論的には「その計算コストを無視しよう」と決めた外部関数のことである．たとえば，普通は関数 $\sin(x)$ を使おうとすると，その計算コストがかかる．けれども，そのコストを敢えて無視し，瞬時に答えを返してもらえる，と考えるのである．それがオラクル関数である．たとえば，$\sin(x)$ をオラクル関数として使用するアルゴリズムを $\mathtt{A}^{\sin}$ など書く．

そのオラクルにランダムな関数を考える．任意の $x \in \{0,1\}^*$ に対して，ランダムな値（0 or 1）を返す関数 $R$ である．アルゴリズム $\mathtt{A}$ が，そうしたランダム関数を使用して計算することを $\mathtt{A}^R$ と書く．一方，関数（つまり問題）の方もランダム関数を用いて定義できる．さらには，ランダム関数 $R$ を用いて多項式時間で計算できる関数の全体を $\mathrm{P}^R$ と

し，ランダム関数 $R$ を用いて（証拠を調べることのできる）NP 問題の全体を $\mathrm{NP}^R$ とする．そうすると $R$ の選び方によっては $\mathrm{P}^R = \mathrm{NP}^R$ となったり，逆に $\mathrm{P}^R \neq \mathrm{NP}^R$ となったりする．

以下では，$R$ をランダムに選んだとき，確率 1 で $\mathrm{P}^R \neq \mathrm{NP}^R$ となることを示す．ランダム関数など計算に影響なさそうな気がする．したがって，確率 1 で $\mathrm{P}^R \neq \mathrm{NP}^R$ なるのであれば，それは P と NP の違いを見出しているのではないか？ それを突き詰めれば，実際にも $\mathrm{P} \neq \mathrm{NP}$ が示せるのではないか？ そう期待できる[1]のである．

以下の議論のための記号の準備をしておく．

$\mathcal{B}X$ ＝ 定義域が $X$ である 0,1-値関数の全体
注）たとえば，$\mathcal{B}\{0,1\}^*$ は定義域が $\{0,1\}^*$ である 0,1-値関数の全体
$R \leftarrow \mathcal{B}X \iff R$ を $\mathcal{B}X$ から等確率で 1 つ選ぶ
$\Pr\left[\mathtt{A}^R = f^R \,\middle|\, R \leftarrow \mathcal{B}\{0,1\}^*\right]$ ＝ $R$ をランダムに選んだとき，
アルゴリズム $\mathtt{A}^R$ が $f^R$ を計算する確率

ここで $R$ を $\mathcal{B}X$ からランダムに選ぶ，ということは，結局は，各 $x \in X$ に対し，$R(x) = 0$ か $= 1$ かを要素ごとに独立に等確率で決める，ということである．

### 2.3. ランダムオラクルのもとでの $\mathrm{P} \neq \mathrm{NP}$

ランダムオラクルのもとでの $\mathrm{P}^R \neq \mathrm{NP}^R$ が確率 1 で成立することを示す．具体的に議論するために，$\mathrm{NP}^R$ 問題で（ランダムオラクルのもとでは）多項式時間で計算できないものを一つ導入しよう．

**例 2.2.** 例 example という意味で関数名を $f_{\mathrm{ex}}$ とする．任意の $x \in \{0,1\}^*$ に対する，この関数の値を次のように定義する．ただし $n = |x|$（$= x$ のビット長）とする．

$$f_{\mathrm{ex}}^R(x) \;=\; \begin{cases} 1, & \exists y \in \{0,1\}^n \,\left[\, R(y0) = R(y00) = \cdots = R(y0^n) = R(y0^{n+1}) = 1 \,\right], \\ 0, & \text{otherwise.} \end{cases}$$

この $f_{\mathrm{ex}}^R$ が $\mathrm{NP}^R$ 問題であることは明らかだろう．以下では，それが難しいこと，すなわち次の定理を証明する．

**定理 2.1.** すべての $c > 0$，時間計算量が $n^c$ 以下のすべてのアルゴリズム $\mathtt{A}$ に対し，

$$\Pr\left[\forall n \in \mathbf{N},\ \forall x \in \{0,1\}^n \ \left[\mathtt{A}^R(x) = f_{\mathrm{ex}}^R(x)\right] \,\middle|\, R \leftarrow \mathcal{B}\{0,1\}^*\right] \;=\; 0$$

となる．

証明は次週に回すことにする．今回は $f_{\mathrm{ex}}$ の基本的な性質について予備的に解説しておく．

---

[1]そういう期待のもと，ランダムオラクルが考えらたのだが，その後，オラクル無しの場合の解析には結びつかないことがわかってきた．

**事実 1.** 各 $n$ において，すべての $x \in \{0,1\}^n$ で $f_{\mathrm{ex}}^R(x)$ の値は同じである．

そこで以下では各 $n$ に対し，$f_{\mathrm{ex}}^R(0^n)$ の値だけを気にすることにする．この値は $R$ の $\{0,1\}^{n+1\sim 2n+1}$ 上での値のみに依存する．そこで $R$ を $\mathcal{B}\{0,1\}^{n+1\sim 2n+1}$ からランダムにとって来て議論してもよい．

**事実 2.** 各 $n$ で

$$0.39 \ < \ \Pr\left[ f_{\mathrm{ex}}^R(0^n) = 1 \,\middle|\, R \leftarrow \mathcal{B}\{0,1\}^{n+1\sim 2n+1} \right] \ < \ 0.5.$$

証明には次の不等式が有用である．また，$\mathrm{e}^{-0.5} < 0.61$ も使う．

**事実 3.** 任意の $0 < x < 1$ と自然数 $k \geq 2$ に対して（二つ目の場合には $kx \leq 2$ の場合）

$$(1-x)^k \ < \ \mathrm{e}^{-kx} \tag{1}$$

$$(1-x)^k \ > \ 1 - kx \tag{2}$$

## References


[1] Ding-Zhu Du and Ker-I Ko, *Theory of Computational Complexity*, John Wiley & Sons, Inc., 2000.

[2] M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, in *Proc. ACM Conference on Computer and Communications Security*, ACM 1993, pp.62–73.