I240 暗号理論 2019
一方向性関数、擬似乱数生成器、擬似ランダム関数 (2)

2019/11/8, 11/13 講師 藤崎

1 擬似ランダム関数族 (Pseudo-Random Function (PRF) Family)

■擬似ランダム関数族の定義 擬似ランダム関数（族）とは、大ざっぱに言うと一見出力がランダム関数に見える関数（族）である。

二変数関数 $F : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{l(n)}$ を考える。$a \in \{0,1\}^n$ に対して、$F_a(\cdot) := F(a,\cdot)$ と定義する。$a \leftarrow \{0,1\}^n$ と一様ランダムに選んだ関数 $F_a(\cdot)$ が、$f \leftarrow \mathsf{Func}(\{0,1\}^m, \{0,1\}^l) := \{f : \{0,1\}^m \to \{0,1\}^l\}$ と一様ランダムに選んで来たランダム関数 f と多項式時間制限のアルゴリズムでは計算量的識別不可であるとき、関数族 $\mathcal{F} := (F)_{n>0}$ は擬似ランダム関数（族）であるという。

正確に定義を述べる。関数族 $\mathcal{F} = (F)_{n>0}$ に対する次のような動作をする $t(n)$-時間確率的アルゴリズム D を考える：D は，任意のサイズ n をパラメータとして取り，オラクル $\mathcal{O}$ に任意の質問 $x \in \{0,1\}^{m(n)}$ を（適応的に）$q(n)$ 回行う。全ての質問が終わった後，D は 1 又は 0 を出力する。オラクルには，関数 F_a もしくはランダム関数 f のどちらかがあらかじめ決定されており，D は $q(n)$ 回の質問を終えた後，どちらの関数と対話したのか判定するアルゴリズムである。オラクルが F_a の時の（すなわち，D の質問 $x \in \{0,1\}^{m(n)}$ に対して $F_a(x)$ を返すオラクルの時)、D がビット b を出力する事象を $\mathbf{D}^{F_a}(1^n) = b$ と書くことにする。同様に，オラクルがランダム関数 f の時（すなわち，D の質問 $x \in \{0,1\}^{m(n)}$ に対して $f(x)$ を返すオラクルの時)、D がビット b を出力する事象を $D^f(1^n) = b$ と書くこととする。任意の n に対して $t(n)$-時間確率的アルゴリズム（族）D が、F をランダム関数から識別する成功確率を

$$\mathsf{Adv}^{\mathsf{prf}}_{D,\mathcal{F}}(q,n) := \left| \Pr_{a\leftarrow\{0,1\}^n, D}[D^{F_a(\cdot)}(1^n) = 1] - \Pr_{f\leftarrow\mathsf{Func}, D}[D^{f(\cdot)}(1^n) = 1] \right|$$

と定義する。ここで q は D のオラクルへの質問回数である。

任意の $t(n)$-時間確率的アルゴリズム D に対して，十分大きな全ての n で $\mathsf{Adv}^{\mathsf{prf}}_{D,\mathcal{F}}(q,n) \le \epsilon(n)$ が成立する時、関数族 $\mathcal{F}$ を $(t(n), q(n), \epsilon(n))$-**擬似ランダム関数族**（又は，$(t(n), q(n), \epsilon(n))$-PRF ）と呼ぶ。$t(n), q(n)$ が多項式制限で（$t(n), q(n) = O(\mathrm{poly}(n))$)、$\epsilon(n)$ が無視できる時（$\epsilon(n) = \mathsf{negl}(n)$）関数族 $\mathcal{F}$ を，**擬似ランダム関数族**と呼ぶ。

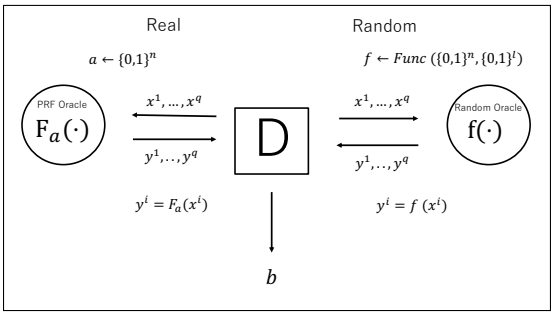

図 1 擬似ランダム関数の識別ゲーム

2 擬似乱数生成器と擬似ランダム関数の等価性

2.1 擬似乱数生成器 → 擬似ランダム関数族

G を，n ビット入力を $2n$ ビット出力に伸長する擬似乱数生成器とする。$G_0(x) := G(x)[1,\ldots,n]$, $G_1(x) := G(x)[n+1,\ldots,2n]$ と定義する。すなわち，$G(x) = G_0(x)||G_1(x)$。この時，二変数関数 $F : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^n$ を，

$$F(a,x) := \quad G_{x_{m(n)}}\Big(\cdots G_{x_2}\Big(G_{x_1}(a)\Big)\cdots\Big)$$

と定義する。ただし，$x := x_1 \cdots x_{m(n)} \in \{0,1\}^{m(n)}$，$x_i \in \{0,1\}$ は x の i ビット目を表す。このように作られた関数 F を Goldwasser-Goldreich-Micali (GGM) 擬似ランダム関数と呼ぶ [GGM86]。この時次の定理が成り立つ。

定理 1（[GGM86]） G を，n ビット入力を $2n$ ビット出力に伸長する $(t(n),\epsilon(n))$-擬似乱数生成器とする。この時 GGM 擬似ランダム関数 F により導入される関数族 $\mathcal{F}$ は、$(t'(n),q(n),\epsilon'(n))$-擬似ランダム関数族であり

$$t'(n) \le t(n) - q(n)(m(n)-1)T_G(n), \qquad \epsilon'(n) \le q(n)m(n)\epsilon(n)$$

を満足する。ただし，$G : \{0,1\}^n \to \{0,1\}^{2n}$ を $T_G(n)$-時間アルゴリズムとする。

証明の前に一つ補題を述べる。

補題 2 $X_1,\ldots,X_q$ を確率分布 $\mathcal{X}$ に従う独立な確率変数、$Y_1,\ldots,Y_q$ を確率分布 $\mathcal{Y}$ に従う独立な確率変数とする。$X_i \approx_{(t,\epsilon)} Y_i$ ならば、$(X_1,\ldots,X_q) \approx_{(t,q\epsilon)} (Y_1,\ldots,Y_q)$ が成り立つ。

（ヒント）$Z_0 := (X_1,\ldots,X_q)$, ..., $Z_i := (Y_1,\ldots,Y_i,X_{i+1},\ldots,X_q)$, ..., $Z_q := (Y_1,\ldots,Y_q)$ と定義して、全ての t-時間アルゴリズム D に対して、ある t-時間アルゴリズム $D_1,\ldots,D_q$ が存在して $\mathsf{Dist}_D(Z_0,Z_1) \le \sum_{i=1}^q \mathsf{Dist}_{D_i}(Z_{i-1},Z_i)$ となることを示せば良い。

問題 3 補題 2 を証明せよ。

(証明) [定理 1 の証明]

■証明の概要 GGM 擬似ランダム関数族 $\mathcal{F}$ に対する (t',q',ϵ')-識別器 D_F を考える。次に、D_F をサブプロトコルとして使い擬似乱数生成器 G に対する識別器 D_G を構成する。G が $(t,\epsilon$-擬似乱数生成器という仮定から、D_G の走行時間を t となるように調整すると、これの識別確率の上限は ϵ であるから、結局 D_F の識別確率の上限を抑えることできる。

■二分木とラベル 深さ m の二分木を考える。頂点のラベルを ϵ とし、頂点を深さ 0 と定義し、頂点から節点までの距離を深さとする。深さ m' $(0 < m' \le m)$ の節点に、図 2 のように $x := x_1 \cdots x_{m'}$ $(x_i \in \{0,1\})$ をつける。ラベル ϵ に $a \in \{0,1\}^n$ を、ラベル $x := x_1 \cdots x_{m'}$ に、

$$f_x(a) := G_{x_{m'}}\Big(\cdots G_{x_2}\Big(G_{x_1}(a)\Big)\cdots\Big)$$

を対応させると、$x \in \{0,1\}^m$ のとき $f_x(a)$ の値が $F_a(x)$ になる。便宜上長さの無いビット列 ϵ に対して $f_\epsilon(a) := a$ と定義する。

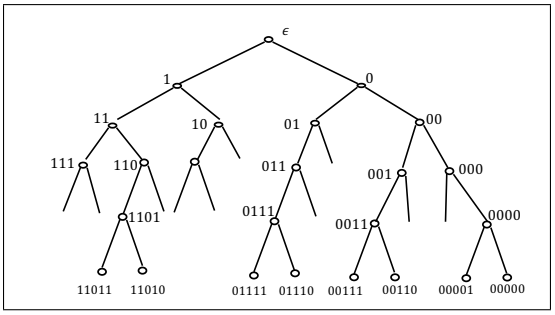

図 2　二分木とラベル

■ハイブリッドアーギュメント（ゲーム変換）　つぎのようにゲームでオラクルを F_a から少しずつ変化させていき、最後のゲームのオラクルをランダム関数 f にする。Game i で D_F がアクセスするオラクルを O_i と書き、$p_i := \Pr_{O_i,D}[D_F^{O_i}(1^n) = 1]$ と定義する。

- Game 0: $a \leftarrow \{0,1\}^n$ と一様ランダムに選ぶ。D_F はオラクル $O_0(\cdot) := F_a(\cdot)$ に（最大 q 回）アクセスし、最後にビット b を出力する。
- Game 1: オラクル O_1 を次のように変える。それ以外にゲームに変更はない。深さ 1 の全てのラベル $\alpha \in \{0,1\}$ に対応する値を $a_1, a_0 \leftarrow \{0,1\}^n$ のような独立な一様ランダム列とし、あとは D_F の質問 $x = bx' \in \{0,1\}^m$（$x' = x_2 \cdots x_m$）に対して、O_1 は $f_{x'}(a_b) = G_{x_m}\Big(\cdots G_{x_2}(a_b)\cdots\Big)$ を返す。
- Game i: オラクル O_1 を次のように変える。それ以外にゲームに変更はない。深さ i の全てのラベル $b \in \{0,1\}^i$ に対応する値を独立な一様な乱数 a_b に変更する。D_F の質問 $x = bx' \in \{0,1\}^m$（$x' = x_{i+1} \cdots x_m$）に対して、O_1 は $f_{x'}(a_b) = G_{x_m}\Big(\cdots G_{x_2}(a_b)\cdots\Big)$ を返す。
- Game m:オラクル O_m を次のように変える。それ以外にゲームに変更はない。$f \leftarrow \mathsf{Func}(\{0,1\}^m, \{0,1\}^n)$ と一様ランダムに選び、$O_m = f$ とする。これは、深さ m の全てのラベル $x\{0,1\}^n$ に対応する値を $\{0,1\}^n$ 上の独立かつ一様ランダムな値 $a_x = f_\epsilon(a_x)$ を選び、質問 x への O_m の返信を a_x としたのと同じである。

三角不等式から、

$$\mathsf{Adv}^{\mathsf{prf}}_{D,\mathcal{F}}(q,n) = |p_0 - p_m| \le \sum_{i=1}^{m} |p_{i-1} - p_i|.$$

■識別器 D_G　$|p_{i-1} - p_i|$ を評価するために、次のような GGM 擬似乱数生成器 G の識別器 D_G を D_F を使って構成することを考える。識別器の入力を $\alpha^1, \ldots, \alpha^q \in \{0,1\}^{2n}$ とし、$\alpha^i = \alpha^i_1 | \alpha^i_0$　$(\alpha^i_1, \alpha^i_0 \in \{0,1\}^n)$ と定義する。$\alpha^1, \ldots, \alpha^q$ は、独立な $a^1, \ldots, a^q \leftarrow \{0,1\}^n$ から計算された、$\alpha^i = G(a^i)$ であるか、または独立な $\alpha^1, \ldots, \alpha^q \leftarrow \{0,1\}^{2n}$ と選ばれた乱数のどちらかである。この時、補題 2 から、D_F は最高で $(t, q\epsilon)$-識別器にしかならない。よって、D_G の走行時間を t に抑えれば、$|p_{i-1} - p_i|$ の上限を抑えられる。

アルゴリズム D_G**:**

1. 入力 $\alpha^1, \ldots, \alpha^q \in \{0,1\}^{2n}$ を受け取る。
2. リスト $L = \emptyset$ と設定する。
3. D_F を走らせる。D_F が i 回目の質問として $x = cbx' \in \{0,1\}^n$（$c \in \{0,1\}^{i-1}$, $b \in \{0,1\}$, $x' \in \{0,1\}^{m-i}$）聞いて来た時、$(cb, a_{cb}) \notin L$ ならば、$a_{c0} := \alpha^i_0$, $a_{c1} := \alpha^i_1$ とし $(c0, a_{c0})$, $(c1, a_{c1})$ を L に登録し、$f_{x'}(a_{cb})$ を返す。

4. D_F がビット b' を出力したら、b' を D_G の出力として出力する。

ここで、$\alpha^1, \ldots, \alpha^q \in \{0,1\}^{2n}$ が擬似乱数の出力とすると D_F は Game (i-1) で動いていることになり、$\alpha^1, \ldots, \alpha^q \in \{0,1\}^{2n}$ が乱数の出力とすると、D_F は Game i で動いていることになる。

D_F の走行時間を $t' \leq t - q(m-i)T_G(n)$ とすると、D_G の走行時間は、$t' + q(m-i)T_G(n) \leq t$ であるから、$|p_{i-1} - p_i| \leq q\epsilon$ が成り立つ。

よって、G が (t,ϵ)-擬似乱数生成器であれば、$\mathcal{F}$ は、

$$t'(n) \leq t(n) - q(n)(m(n)-1)T_G(n), \qquad \epsilon'(n) \leq q(n)m(n)\epsilon(n)$$

なる (t', q, ϵ')-擬似ランダム関数族である。証明終わり。 ■

2.2 擬似ランダム関数族 → 擬似乱数生成器

関数 $F : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{l(n)}$ により導入される擬似ランダム関数族を $\mathcal{F} := (F)_{n>0}$ とする。関数 $G : \{0,1\}^n \to \{0,1\}^{l'(n)}$ $(n < l'(n))$ を

$$G(x) := \quad \Big(F_x(0) \,||\, F_x(1) \,||\, \cdots \,||F_x(M-1)\Big)[1, \ldots, l'(n)]$$

と定義する。ただし $M := \left\lceil \frac{l'(n)}{l(n)} \right\rceil$ である。

この時，次の定理が成立する。

定理 4 関数 $F : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}^{l(n)}$ により導入される関数族 $\mathcal{F} := (F)_{n>0}$ が $(t(n), q(n), \epsilon(n))$-擬似ランダム関数族であるとする。$\max\{n, l(n)q(n)\} \leq l'(n)$ となるような任意の $l'(n)$ に対して、上記のように構成された関数 G は，$(l'(n) - n)$ の伸長度を持つ $(t(n), \epsilon(n))$-擬似乱数生成器である。

(証明)

■証明の概要 擬似乱数生成器 G に対する識別器 D_G を考える。この時、D_G を使って擬似ランダム関数族 $\mathcal{F}$ に対する識別器 D_F を構成する。D_F はオラクル O にアクセスし、O が擬似ランダム関数か真のランダム関数かを識別する。

アルゴリズム D_F^O:

1. 入力として 1^n を受け取る。
2. $M = \left\lceil \frac{l'(n)}{l(n)} \right\rceil$ を計算し、$0, 1, \ldots, (M-1)$ をオラクルに質問し、$O(0), O(1), \ldots, O(M-1)$ を受け取る。
3. ビット列 $\alpha = \Big(O(0) \,||\, \ldots \,||\, O(M-1)\Big)[1, \ldots, l'(n)]$ を作る。
4. G に対する識別器 D_G にビット列 α を入力する（α は $l'(n)$-ビット列であることに注意)。
5. $D_G(\alpha)$ の出力 b を D_F の出力として出力する。

■識別確率の解析

$$\Pr_{X, D_G}[D_G(G(X)) = 1] = \Pr_{X, D_F}[D_F^{F_X(\cdot)}(1^n) = 1]$$
$$\Pr_{U_{l'}, D_G}[D_G(U_{l'}) = 1] = \Pr_{f \leftarrow \mathsf{Func}, D_F}[D_F^{f(\cdot)}(1^n) = 1].$$

なので、

$$\mathsf{Adv}^{\mathsf{prg}}_{D_G,G}(n) = \mathsf{Adv}^{\mathsf{prf}}_{D_F,\mathcal{F}}(q,n)$$

D_F のオラクルへのアクセス回数は、$M \leq q(n)$ 回。D_F の走行時間は、D_G の走行時間 t' 以外はほぼ無視できるので、D_G の走行時間を t で制限すれば、$\mathsf{Adv}^{\mathsf{prg}}_{D_G,G}(n) \leq \epsilon(n)$ が成立。よって、$\mathcal{F}$ が (t,q,ϵ)-PRF ならば、G は (t,ϵ)-PRG となる。 ■

参考文献

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.