テーマ２：公開鍵暗号と署名の安全性

本日の講義の目的

- 計算論的暗号理論速成コース
- 平均計算量の枠組み，公開鍵暗号，公開鍵署名

1.1. 平均計算量の枠組み入門

計算論的暗号理論 computational cryptography とは，秘密通信に関わる当事者（送信者と受信者）だけでなく，仮想敵対者（盗聴者や暗号偽造者など）も,「計算論的に見ると万能でなく計算量的な限界がある」という想定のもとで秘密通信等の方法を議論する分野である．

その議論の枠組みを簡単に示すために，ここでは（多項式時間）一方向関数を例にとって説明する．

以下では，関数はすべて 0,1-記号列の集合 $\{0,1\}^*$ 上の関数とする．関数の多項式時間計算可能性・不可能性について，以下のように定義する[1]．

定義 1.1. 入力列長 n を計算量サイズパラメータとしたとき，時間計算量が $\mathrm{time}A(n) = n^{O(1)}$ となるアルゴリズムを多項式時間アルゴリズムという．

定義 1.2. 定義域が $\{0,1\}^*$ である関数 f が多項式時間計算可能であるとは，任意の文字列 $w \in \{0,1\}^*$ に対し，$f(w)$ を計算する多項式時間アルゴリズム $\mathtt{A}$ が存在することである．

定義 1.3. 定義域が $\{0,1\}^*$ である関数 g が平均的に多項式時間**強**計算不可能であるとは，任意の多項式時間アルゴリズム $\mathtt{A}$ に対し，

$$\forall c > 0,\ \exists n_c,\ \forall n \geq n_c\ \left[\Pr\left[\mathtt{A}(w) = g(w) \,\middle|\, w \leftarrow \{0,1\}^n\right] \leq n^{-c}\right] \tag{1}$$

が成り立つことである．ここで $\Pr[\cdots|w \leftarrow \{0,1\}^n]$ とは，w を $\{0,1\}^n$ から一様ランダムに選んだときに $\cdots$ が起きる確率である．

計算論的暗号理論では，計算量サイズパラメータをもう少し一般化して，入力サイズ以外のものを安全性パラメータとして用いることもある．そのように選んだパラメータに対して，上記の (1) を満たす関数を無視できるほど小さい（関数）などという．

[1] このテーマでは簡単のため，鍵生成などの特殊の場合を除いて通常の計算は決定性アルゴリズムで行われるものとする．本来は乱数を利用した乱択アルゴリズムを考えるべきであるが，ここでは省略する．

定義 1.4. パラメータ $k \in \mathbf{N}$ に対する関数 $\varepsilon(k)$ が

$$\forall c > 0,\ \exists k_c,\ \forall k \geq k_c\ \left[\varepsilon(k) \leq k^{-c}\right]$$

を満たすとき，$\varepsilon(k)$ を 無視できるほど小さい (negligible) という．

一方向関数とは，f 自身は多項式時間計算可能だが，その逆関数 f^{-1} の計算は平均的に多項式時間強計算不可能な関数である．ただし，f^{-1} は一般には関数にならないので「強計算不可能性」の定義を少し変える必要がある．

定義 1.5. 一方向関数 f とは，多項式時間計算可能で，しかも f^{-1} が次の意味で平均的に多項式時間強計算不可能な関数である：関数 f の入力長をサイズパラメータとしたとき，任意の多項式時間アルゴリズム $\mathtt{A}$ に対し，

$$\forall c > 0,\ \exists n_c,\ \forall n \geq n_c\ \left[\Pr\left[\mathtt{A}(f(w)) \in f^{-1}(f(w)) \,\middle|\, w \leftarrow \{0,1\}^n\right] \leq n^{-c}\right] \tag{2}$$

が成り立つ．ただし $f^{-1}(y) = \{x | f(x) = y\}$ とする．

この定義では，通常のようにアルゴリズムに与える入力長をサイズパラメータにするのではなく，元々の関数の入力長をサイズパラメータ（もしくはセキュリティパラメータ）としている点に注意しよう．

例 1.1. 上の定義を満たす一方向関数を示すのは少々難しい．けれども，たとえば，次のような単純な乗算関数は（そこそこの）一方向性を持っていると予想されている．それを基礎に上記のような一方向関数を構成する方法も知られている [1].

$$\mathrm{mult}(w) \ = \ \begin{cases} 0\mathrm{bin}\,(x \cdot y)\,, & w = \mathrm{bin}(x)\mathrm{bin}(y) \text{ で } |\mathrm{bin}(x)| = |\mathrm{bin}(y)| \text{ のとき} \\ 1w, & \text{その他のとき} \end{cases}$$

ただし，$\mathrm{bin}(x)$ で自然数 x の二進表記を表すことにする．

2.2. 公開鍵暗号と公開鍵署名入門

以下では [2] に従い，公開鍵暗号と公開鍵署名に関する定義を説明を交えずに述べる．その直感的な意味や，公開鍵暗号の実際の使い方については講義で述べる．

以下では乱択アルゴリズム $\mathtt{A}$ の入力記号列 x に対する出力の全体を $[\mathtt{A}(x)]$ で表すことにする．

定義 1.6. 任意の $k \in \mathbf{N}$ に対し，次の条件を満たす対 (f, f^{-1}) を $k^c + c$ 時間以内に生成する乱択アルゴリズム $\mathtt{G}$ を（仮の）落とし戸関数生成器という[2]：

[2] ここでは簡単のため，関数 f の定義域が $\{0,1\}^k$ 全体の場合に限った説明をする．

f, f^{-1} は $\{0,1\}^n$ 上の単射（を計算するプログラムのコード）で，その計算時間は各々 n^c+c である．また，f^{-1} は f の逆関数．つまり，任意の $x \in \{0,1\}^n$ に対し $f^{-1}(f(x)) = x$ となる．

ここでの k をセキュリティパラメータという．また，上記の c は k に依存しない正の定数とする．なお，G の生成する関数 f が，つねに $\{0,1\}^k$ 上の全射である場合には，G を落とし戸置換生成器という．ここでは簡単のため，このような落とし戸置換（とその生成器）を考える．

関数 f を暗号化関数，そして f^{-1} を復号関数と呼ぶ．ただ，通常の生成器は，たとえば f として，プログラム自身を出力するのではなく，暗号化関数を計算するプログラム Encode へのパラメータ e を出力する．つまり，実際の f は

$$f(x) \;=\; \mathtt{Encode}(e, x)$$

と定義されるのである．このときの e を暗号鍵もしくは公開鍵という．同様に復号関数 f^{-1} を計算する鍵を秘密鍵（もしくは復号鍵）という．上記の枠組みでは f 自身を公開鍵，そして f^{-1} 自身を秘密鍵と呼ぶことにする．

定義 1.7. （仮の）落とし戸関数生成器 G の生成する関数族が次を満たすとき，G を（安全な）落とし戸関数生成器という：任意の多項式時間（乱択）アルゴリズム A に対し，次の確率がセキュリティパラメータ k に対して無視できるほど小さい．

$$\Pr\left[\, \mathtt{A}(f, y) = x \,\middle|\, (f, f^{-1}) \leftarrow \mathtt{G}(k), x \leftarrow \{0,1\}^n \,\right] \tag{3}$$

例 1.2. よく知られている RSA 暗号[3] を，我々の形式化で書くと次のようになる．

まず，RSA 暗号系は自然数上で定義されているが，以下では自然数は適当な形で $\{0,1\}^*$ 上にコード化されているものとする．また，アルゴリズム（正確にはプログラム）もコード化されているものとする．

次のような関数を計算するアルゴリズムを，それぞれ，暗号化アルゴリズム E，復号アルゴリズム D として定義する．

準備

p, q：任意に選んだ異なる素数の組

e：以下の L より小さく，L と素となる数の中から任意に選んだもの

$N = pq$, $L = \mathrm{LCM}(p-1, q-1)$, $d \equiv e^{-1} \pmod{L}$

定義（任意の自然数 $x < N$, $y < N$ に対し）

$\mathtt{E}(N, e, x) \;=\; x^e \pmod{N}$

$\mathtt{D}(N, d, y) \;=\; y^d \pmod{N}$

[3] RSA = Rabin-Shamir-Adleman

重要なのは $\mathtt{D}(N, d, \mathtt{E}(N, e, x)) = x$ となる点である．

また，次のような多項式時間乱択アルゴリズム pgen を考える．

<u>定義</u>（任意の自然数 k に対し）

$\mathtt{pgen}(k)$ ＝ 二進 k 桁の素数 p（等確率で）

$\mathtt{egen}(L)$ ＝ L 未満で L と素となる数（等確率で）

そうすると，次のような出力を持つ $\mathtt{G}_{\mathrm{RSA}}$ が，RSA に対応する落とし戸関数生成器である．

$$\mathtt{G}_{\mathrm{RSA}}(k) \ = \ (\mathtt{E}(N, e, \cdot), \mathtt{D}(N, d, \cdot)),$$

ただし，$p \leftarrow \mathtt{pgen}(k)$, $q \leftarrow \mathtt{pgen}(k)$, $N = pq$, $L = \mathrm{LCM}(p-1, q-1)$, $e \leftarrow \mathtt{egen}(L)$, そして $d \equiv e^{-1} \pmod{L}$ とする．

注）この場合暗号化関数 $f(x) = \mathtt{E}(N, e, x)$，復号関数 $f^{-1}(y) = \mathtt{D}(N, d, y)$ の定義域は $\{0, 1, ..., N-1\}$ である．これは二進列で考えると $\{0,1\}^k$ の部分集合で $\{0,1\}^k$ そのものではない．この点，今回の定義とちょっと違うが，まあ大目に見てもらいたい．また，関数 $f(x)$ は完全に 1 対 1 ではない．その点も大目に見てもらおう．

定義 1.8. 落とし戸置換生成器 G を考える．これに加えて次の条件を満たす多項式時間乱択アルゴリズム Sign, Verify を合わせたものを，<u>公開鍵暗号＆署名系</u>と呼ぶ（以下では k をセキュリティパラメータの 1 つの値，(f, f^{-1}) を $\mathtt{G}(k)$ の出した公開鍵と暗号鍵，そして x を任意の $\{0,1\}^k$ の元とする）：

(1) $[\mathtt{Sign}(f^{-1}, x)] \subseteq \{0,1\}^k$.

(2) $\forall \sigma \ \Big[\ \sigma \in [\mathtt{Sign}(f^{-1}, x)] \iff \mathtt{Verify}(f, x, \sigma) = 1 \ \Big]$.

(3) 任意の多項式時間乱択署名偽造アルゴリズム F に対し，以下の確率がセキュリティパラメータ k に対して無視できるほど小さい．

$$\Pr\Big[\mathtt{Verify}(f, x, \sigma) = 1 \,\Big|\, (f, f^{-1}) \leftarrow \mathtt{G}(k), (x, \sigma) \leftarrow \mathtt{F}^{\mathtt{Sign}(f^{-1}, \cdot)}(k) \Big] \tag{4}$$

ただし，署名偽造アルゴリズム F は，x についての署名を Sign に頼むことはできない．

例 1.3. RSA 暗号を用いた最も単純な署名系は，自分の復号関数 f を用いて，メッセージ x に対して $\sigma = f(x)$ と計算する方法である．

References

[1] 渡辺治，一方向関数の基礎理論，離散構造とアルゴリズム　，近代科学社，1994.

[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.