公開鍵暗号入門

情報セキュリティ大学院大学　有田正剛

平成 19 年 5 月 8 日

目 次

1　はじめに **1**

2　公開鍵暗号の基本機能 **2**

3　公開鍵暗号の定義 **3**

4　Rabin 暗号 **3**
4.1　必要な整数論 4
4.2　暗号方式 4

5　暗号の強秘匿性 **5**
5.1　Rabin 暗号の問題点 5
5.2　強秘匿性の定義 6

6　Blum-Goldwasser 暗号 **7**

7　おわりに **9**

1　はじめに

公開鍵暗号は今や私たちにとってとても身近な存在である。我たちはそれとは意識せずに日々色々な場面で公開鍵暗号を使っている。Internet では SSL、高速道路では ETC、また駅では Suica が私たちの代理として暗号演算を実行している。

この入門講義では、そのような公開鍵暗号とはどんな技術か、そのからくりはどうなっているか、またなぜ安全といえるか、解説したい。

公開鍵暗号は予め秘密を共有しないで秘密の通信を行う方法である。そのような公開鍵暗号をつくる上で、数学の一分野である整数論がとても役

に立つ。ここでは例としてRabin暗号を取り上げ、それが整数の平方演算を利用して作られることをみる。そして、私たちや私たちが作り出したコンピュータにとって整数を因数分解することが困難であるために、Rabin暗号が暗号として成立する（すなわち、暗号文から元のメッセージを求めることができない）のをみる。

しかし、公開鍵暗号をさらに「強秘匿」なものにするには、整数論だけでは足りない。我々にとって整数論的課題のどこが難しいのかを見極め、乱数をうまく用いて、ある工夫を施す必要がある。そのような工夫を用いて、Rabin暗号を強化することで、強秘匿な暗号（すなわち、暗号文から元のメッセージのどのような部分情報さえ求めることができない暗号）であるBlum-Goldwasser暗号が作られることをみる。

2　公開鍵暗号の基本機能

暗号とは、秘密の通信を行うために利用される技術である。暗号には、大きく分けて、共通鍵暗号と公開鍵暗号がある。共通鍵暗号は、「予め共有した秘密を用いて秘密の通信を行う方法」である。予め共有した秘密は共通鍵と呼ばれる。送信者はメッセージを共通鍵で暗号化し、暗号文を受信者に送る。暗号文を受け取った受信者はそれを共通鍵で復号してもとのメッセージを得る。予め共有した秘密があるなら、それを用いれば秘密通信ができだろうことは、具体的な方法はともかくとして、もっともなことである。

一方、公開鍵暗号は「予め秘密を共有しないで秘密の通信を行う方法」である。これは、一見すると受けいれがたく、そんなことができるものか、と思えるが、我たちやコンピュータの能力が有限であることを利用すれば実現できる。

公開鍵暗号では各自の鍵は公開鍵と私有鍵の組からなる。公開鍵は（その所有者への）メッセージの暗号化に用い、私有鍵は（その所有者が自分宛の）暗号文の復号に用いる。ある公開鍵で暗号化された暗号文は、それとセットになっている私有鍵でしか、復号されないようになっている。

公開鍵暗号を用いて、実際にメッセージを暗号化して送るには、以下のようにする。まず、受信者は自分の公開鍵を、その名の通り、公開する。送信者は、公開された受信者の公開鍵を手に入れ、この公開鍵を用いてメッセージを暗号化し、受信者に送る。暗号文を受け取った受信者はそれを自分の私有鍵で復号してもとのメッセージを得る。ここで、この受信者以外は、メッセージの暗号化に用いられた公開鍵に対応する私有鍵を持っていないので、この暗号文を復号することはできない。

3　公開鍵暗号の定義

公開鍵暗号は、鍵生成アルゴリズム G、暗号化アルゴリズム E そして復号アルゴリズム D の 3 つの効率的なアルゴリズムの組と定義される。

鍵生成アルゴリズム G は、セキュリティパラメータ k を入力すると公開鍵 e と私有鍵 d の組を出力する：

$$(e, d) \leftarrow \mathsf{G}(k).$$

ここで、セキュリティパラメータ k とは暗号の強度を指定するパラメータである。

暗号化アルゴリズム E は、公開鍵 e とメッセージ m を入力すると暗号文 c を出力する：

$$c \leftarrow \mathsf{E}(e, m).$$

復号アルゴリズム D は、私有鍵 d と暗号文 c を入力するとメッセージ m を出力する：

$$m \leftarrow \mathsf{D}(d, c). \tag{1}$$

ただし、

完全性 G によって生成された任意の鍵ペア (e, d) と任意のメッセージ m に対して

$$\mathsf{D}(d, \mathsf{E}(e, m)) = m$$

でなければならない。これは、暗号化して復号したら必ずもとのメッセージが得られるということを意味する。

また、暗号文 $c(\leftarrow \mathsf{E}(e, m))$ を入手した敵が、c および公開鍵 e を用いて、m を計算できるようでは暗号といえない。つまり、

一方向性 どのような効率的なアルゴリズム A も e と c から m を求めることはできない

ことが必要である。

公開鍵暗号を作るとは、完全性と一方向性を同時に満たす 3 つのアルゴリズムの組 $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ を求めることに他ならない。

4　Rabin 暗号

公開鍵暗号の一例として、Rabin 暗号を紹介する。Rabin 暗号は整数を法とする平方演算の特性を利用してつくられる。

4.1 必要な整数論

まず、必要となる整数論的な事実を述べる。

n を異なる 2 つの素数 p,q の積である整数とする。$a \bmod n$ は整数 a を整数 n でわった余りを表す。

整数 n に対して $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ とする。Q_n を n を法とする平方剰余の集合とする。すなわち、Q_n はある $b(\in \mathbb{Z}_n)$ について $a = b^2 \bmod n$ となる $a(\in \mathbb{Z}_n)$ の集合である。n を法とする平方剰余 a に対して、$a = b^2 \bmod n$ となる b を、a の n を法とする平方根と呼ぶ。

事実 1 整数 $n(= pq)$ の素因数分解を効率的に実行するアルゴリズムが存在しないならば、n を法とする平方根を計算する効率的なアルゴリズムも存在しない。

事実 2 任意の素数 p を法とするならば、平方根を計算する効率的なアルゴリズムが存在する。すなわち、ある効率的なアルゴリズム Sqrt があって、素数 p と、p を法とする平方剰余の集合 Q_p からランダムに選ばれた a を入力として、$a = b^2 \bmod p$ となる b を出力する: $b \leftarrow \mathsf{Sqrt}(p, a)$.

事実 1 と事実 2 は、整数 $n(= pq)$ に対して、平方関数 $y = \mathrm{RABIN}_n(x) = x^2 \bmod n$ がトラップドア付き一方向関数であることを示している。

すなわち、順方向の $y = \mathrm{RABIN}_n(x)$ の計算は、平方を計算し n で割り算するだけなので効率的に実行できる。しかし、事実 1 より、逆に y から $x = \mathrm{RABIN}_n^{-1}(y)$ を求める効率的なアルゴリズムは (素因数分解が困難ならば) 存在しない。

ところが、n の素因子 p, q が与えられれば、そのような逆方向の計算も効率的に実行できる: $x = \mathrm{RABIN}_n^{-1}(p, q, y)$. (この p, q のような一方向性を破る情報をトラップドアと呼ぶ。) 実際、アルゴリズム Sqrt を用いて y の mod$p$ での平方根 x_p を計算する。同様に、y の mod$q$ での平方根 x_q を計算する。(x_p, x_q) を中国人の剰余定理を用いて mod$n$ にリフトして y の n を法とする平方根 x を得る。

4.2 暗号方式

Rabin 暗号 $Rabin = (\mathsf{G}, \mathsf{E}, \mathsf{D})$ は、トラップドア付き一方向関数 $y = \mathrm{RABIN}_n(x)$ を用いて、以下のように構成される。

鍵生成アルゴリズム $\mathsf{G}(k)$

それぞれ k ビットの異なる素数 p, q をランダムに生成する。$n = pq$ を公開鍵 e とし、素数 p, q を私有鍵 d とする。

暗号化アルゴリズム $\mathsf{E}(e, m)$

公開鍵 e から n を取り出して、$c = m^2 \bmod n$ を計算し、c を出力する。

復号アルゴリズム $\mathsf{D}(d, c)$

私有鍵 d から p と q を取り出して、アルゴリズム Sqrt を用いて c の $\bmod p$ での平方根 $\pm m_p$ を計算する。同様に、c の $\bmod q$ での平方根 $\pm m_q$ を計算する。$(\pm m_p, \pm m_q)$ を中国人の剰余定理を用いて $\bmod n$ にリフトして (得られる 4 通りのメッセージ候補から)m を得て、出力する。

トラップドア付き一方向関数 $y = \mathrm{RABIN}_n(x)$ の n を公開鍵、トラップドア p, q を私有鍵とするわけである。

Rabin 暗号が完全性条件を満たしているのは、暗号化は平方演算で復号は平方根演算なので、明らかである。一方向性条件は、整数 $n(= pq)$ の素因数分解が困難であるとの仮定のもと、事実 1 からしたがう。

5 暗号の強秘匿性

5.1 Rabin 暗号の問題点

Rabin 暗号を用いて秘密通信をしてみる。今、送信者 Alice が受信者 Bob に (ある案件について) 賛成か反対かのいずれかを他者には秘密にして伝えたいとする。あらかじめ、賛成は整数 m_{yes} で反対は整数 m_{no} で表すことになっている (このルールは公開されているとする)。Alice は賛成しているとして、そのことを以下のように Rabin 暗号で Bob に伝える。Alice は、Bob の公開鍵 n_B を入手し、$c = m_{yes}^2 \bmod n_B$ を計算し、暗号文として c を Bob に送信する。Bob は、暗号文 c を受信し、自身の私有鍵 p_B, q_B を用いて c の n_B を法とする平方根を求めて、Alice のメッセージ m_{yes} を得る。この送信の途中で敵 Eve が c を覗き見しているかも知れないが、敵 Eve は暗号文 c と Bob の公開鍵 n_Bを 入手しても、対応する私有鍵 p_B, q_B を持たないので、c の n_B を法とする平方根、すなわち m_{yes} を求めることはできない、はずだった。

ところが、このケースでは、敵 Eve は c から Alice のメッセージ m_{yes} を簡単に求めることができる。Eve はまず m_{no} を Bob の公開鍵 n_B で暗号化して c' を求める。c' と c を比較するとそれらは異なるので、c に暗号化されていたのは m_{yes} であることが分かる。

ここでは、メッセージのパターンが m_{yes} と m_{no} の 2 通りという極端な設定で考え、そのせいでメッセージが丸ごとばれてしまったが、メッセー

ジがいく通りあってもやはり不都合がある。実際、Eve が自分でメッセージ m' を選択し、それを暗号化した結果 c' と c を比較すれば、m' が c に暗号化されているかどうかがわかる。これを繰り返せば、(c に暗号化された)m のとり得る範囲を狭めていくことができる。Rabin 暗号では、暗号文からもとのメッセージに関する情報が漏れているということである。暗号文を映像に例えれば、Rabin 暗号は衛星放送でよく見かけるスクランブル映像のようなものである。番組内容の詳細は分からないが、映画かスポーツ中継かぐらいは分かる。

5.2 強秘匿性の定義

前節で Rabin 暗号では暗号文からもとのメッセージに関する情報が漏れてしまっているのを見た。Rabin 暗号を、もとのメッセージに関するどのような情報も漏れないよう、強化することができ、そのようにして得られる暗号として 6 節で紹介する Blum-Goldwasser 暗号がある。

しかし、その前に、ゴールを明確にしたい。「暗号文からもとのメッセージに関するどのような情報も漏れていない」とはどういうことか？個々の暗号方式に依存せず、普遍的でかつ容易に検証できる定義がほしい。それは暗号の強秘匿性と呼ばれ、つぎにように敵 A と挑戦者 C との間のゲームを用いて定式化することができる。

強秘匿性 公開鍵暗号 $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ に対して、敵 A と挑戦者 C との間でゲームを行う。ルールは以下の通り。まず、挑戦者 C が鍵生成アルゴリズム G を実行し、鍵ペア (e, d) をつくる。挑戦者 C は公開鍵 e を敵 A に渡す。e を受け取ったら、敵 A は何らかの計算をしてメッセージの組 (m_0, m_1) を選択し、挑戦者 C に渡す。これに対して挑戦者 C は m_0 か m_1 かのいずれかをランダムに選び（これを m_b とかく）公開鍵 e で暗号化して暗号文 c をつくる: $c \leftarrow \mathsf{E}(e, m_b)$. 挑戦者 C は暗号文 c を敵 A に渡す。敵 A は受け取った暗号文 c が m_0 を暗号化したものか、m_1 を暗号化したものかどちらかを推測し、推測結果 b' を出力する。推測が当たったら ($b = b'$ なら) 敵 A の勝ち (そうでないなら挑戦者 C の勝ち) とする。

このとき、どのような敵 A もせいぜい 1/2 の確率でしか勝てないならば、公開鍵暗号 $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ は**強秘匿**と呼ばれる。

もし、公開鍵暗号 $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ がメッセージの情報を 1 ビットも漏らしていないならば、敵 A がどんなに強力でも、c が m_0 を暗号化したのか m_1 を暗号化したのか、全く分からず、b' をあてずっぽうで出力するしかない。このとき敵 A が勝つ確率は 1/2 である。

一方、公開鍵暗号 $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ がメッセージの情報を何らかの形で漏らしているならば、巧みな敵 A が、その漏洩を突くようなメッセージの組 (m_0, m_1) を選択して挑戦者 C に提出すれば、その敵 A は高い確率で勝つはずである。

6 Blum-Goldwasser 暗号

Rabin 暗号を強化することで、強秘匿な暗号である Blum-Goldwasser 暗号が得られる。強化のポイントは、暗号化に乱数を導入することと hard-core ビットの利用である。

4 節で見たように、関数 $y = \mathrm{RABIN}_n(x)(= x^2 \bmod n)$ は一方向関数 (の有力候補) である。しかし、y は x の全てを隠せているわけではない。実際、x と (x とは異なる)x' が与えられたとき、どちらが y の原像かと問われれば、x であると分かってしまう ($x^2 \bmod n$ が y になることを確認する)。このため、Rabin 暗号は強秘匿な暗号にはならないのだった。

ところが、$y(= \mathrm{RABIN}_n(x) = x^2 \bmod n)$ は、x の最下位ビット $x_0 = \mathrm{LSB}(x)$、すなわち x が偶数かそれとも奇数かということ、は隠せていることが知られている。つまり、y が与えられても、0 と 1 のどちらが x の最下位ビットか分からない。このことを関数 $y = \mathrm{RABIN}_n(x)$ は hard-core ビットとして最下位ビット $\mathrm{LSB}(x)$ をもつという。関数 $y = \mathrm{RABIN}_n(x)$ に限らず、どのような一方向関数も hard-core ビットをもつ（ように変形できる）ことが知られている。

Blum-Goldwasser $= (\mathsf{G}, \mathsf{E}, \mathsf{D})$ は以下のように構成される。

鍵生成アルゴリズム $\mathsf{G}(k)$

それぞれ k ビットの、8 でわって 7 余る、異なる素数 p, q をランダムに生成する。$n = pq$ を公開鍵 e とし、素数 p, q を私有鍵 d とする。

暗号化アルゴリズム $\mathsf{E}(e, m)$

m のビット長を l とする。公開鍵 e から n を取り出して、

1. $\mathbb{Z}_n$ からランダムに z を選び、$x_0 = z^2 \bmod n$ を求める。
2. $i = 1, \ldots, l$ に対して、$x_i = x_{i-1}^2 \bmod n$ を計算する。
3. $mask = (\mathrm{LSB}(x_0), \mathrm{LSB}(x_1), \ldots, \mathrm{LSB}(x_{l-1}))$ を求め、$c_1 = mask \oplus m$ を計算する ($\oplus$ はビット毎の排他的論理和)。
4. $c = (c_1, x_l)$ を出力する。

復号アルゴリズム $\mathsf{D}(d, c)$

私有鍵 d から p と q を取り出して、$c = (c_1, y)$ に対し、

1. p, q を用いて平方根演算を繰り返し、$y(= x_l)$ から x_0 を求める。
2. $i = 1, \ldots, l$ に対して、$x_i = x_{i-1}^2 \bmod n$ を計算する。
3. $mask = (\mathrm{LSB}(x_0), \mathrm{LSB}(x_1), \ldots, \mathrm{LSB}(x_{l-1}))$ を求め、$m = mask \oplus c_1$ を計算し出力する。

Blum-Goldwasser 暗号では、Q_n のランダム要素 x_0 をとり、$x_{i+1} = \mathrm{RABIN}_n(x_i)$ の hard-core ビット $\mathrm{LSB}(x_i)$ を利用して、(メッセージ m と同じ長さの) ビット列 $mask = (\mathrm{LSB}(x_0), \mathrm{LSB}(x_1), \ldots, \mathrm{LSB}(x_{l-1}))$ を生成し、$mask$ をメッセージ m に足しこむことによって暗号文 $c_1 = mask \oplus m$ を生成している。ただし、c_1 だけでは誰も復号できなくなるので、x_l を暗号文に添付している: $c = (c_1, x_l)$.

$c = (c_1, x_l)$ を受け取った正当な復号者は、私有鍵 p, q を用いて平方根演算を繰り返し、x_l から x_0 を求めることができる。一旦 x_0 が分かれば、暗号化のときと全く同じ手続きで $mask$ を生成すれば、$m = mask \oplus c_1$ によってメッセージ m が得られる。

一方、敵が暗号文 $c(= (c_1, x_l))$ に含まれる x_l を見ても、hard-core ビットの性質から c_1 を作るのに用いた $mask$ の各ビットは分からない。実際、$mask = (\mathrm{LSB}(x_0), \mathrm{LSB}(x_1), \ldots, \mathrm{LSB}(x_{l-1}))$ の各ビットを見ると

- $\mathrm{LSB}(x_{l-1})$ は $x_l(= x_{l-1}^2 \bmod n)$ の hard-core ビット
- $\mathrm{LSB}(x_{l-2})$ は $x_{l-1}(= x_{l-2}^2 \bmod n)$ の hard-core ビット
- $\ldots$
- $\mathrm{LSB}(x_0)$ は $x_1(= x_0^2 \bmod n)$ の hard-core ビット

となっている。敵は x_l を知っているが、hard-core ビットの性質から、$\mathrm{LSB}(x_{l-1})$ は全く分からない。敵は、x_{l-1} の部分情報を知っているかもしれないが、$\mathrm{LSB}(x_{l-2})$ は全く分からない。(x_{l-1} を知っていても $\mathrm{LSB}(x_{l-2})$ は全く分からないから。) 以下同様に繰り返し、敵は、x_1 の部分情報を知っているかもしれないが、$\mathrm{LSB}(x_0)$ は全く分からない。

以上から、$mask$ は敵にとって未知の (メッセージ m と同じ長さの) 乱数列に等しいことが分かった。すると、$c = mask \oplus m$ は m の情報を完全に隠してしまうので、強秘匿性の定義のゲームに現れた攻撃者 A に勝ち目はなく、Blum-Goldwasser 暗号 $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ は強秘匿となる。映像に例えれば、Blum-Goldwasser 暗号はサンドストームといったところである。映画かスポーツ中継か、そもそも放送中なのかも分からない。

7　おわりに

公開鍵暗号の例として、整数を法とした平方関数の一方向性を利用した、Rabin 暗号を紹介し、それを hard-core ビットを利用して強化することで強秘匿な Blum-Goldwasser 暗号が構成されることを見た。

ここでは、攻撃モデルとしては選択メッセージ攻撃しか扱わなかった。選択メッセージ攻撃とは、敵がメッセージを選択し、それに対する暗号文が与えられる攻撃モデルである。より強力な攻撃モデルとして、敵が暗号文を選択し、それに対するメッセージが与えられる攻撃である選択暗号文攻撃がある。また、安全性定義についても、強秘匿性のほかに頑健性と呼ばれる安全性がある。