電気通信大学 情報理工学域 II 類 情報通信工学プログラム/電子情報学プログラム

情報通信工学実験 B / 電子情報学実験 B
実験項目 情報通信（情報セキュリティ）[1]

課題「暗号化の理解とプログラミング」— RSA 暗号の理解とプログラミング —

概要

一般に、情報（メッセージ）を送信者 (情報源) から受信者 (利用者) に送信する際、図 1 に示す 3 種類の**符号化**処理が行われる。最初に、効率性を目的として、**情報源符号化（データ圧縮）**を行なう。次に、安全性を目的として、**暗号化（暗号）**を行なう。最後に、信頼性を目的として、**通信路符号化（誤り訂正符号）**を行なう。そこで、本実験項目「情報セキュリティ」では、「暗号化」に関する符号化を取り上げ、RSA 暗号の符号化 (暗号化) と復号化の具体的な手法について理解し、その符号化と復号化のプログラムを作成することを課題とする。

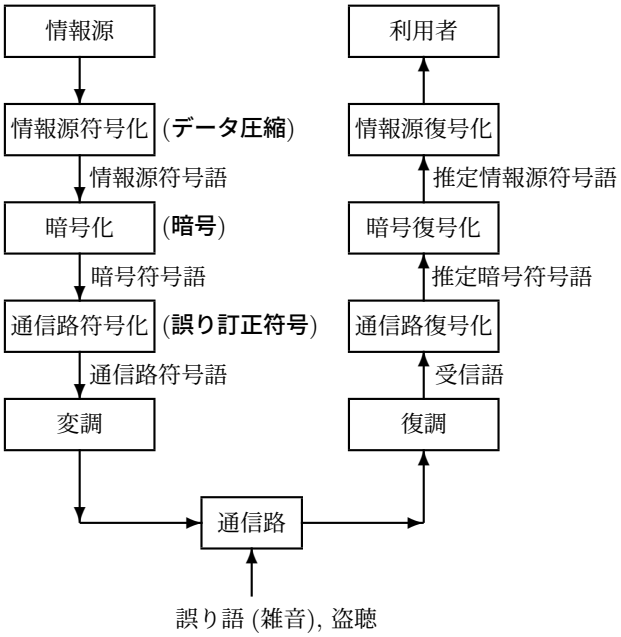

図 1: ディジタル通信システム

テキストの構成

本テキストの構成は次のとおり．1 節では，本課題の目的を述べる．2 節では，提出するレポートの要件を述べる．3 節では，本テキストの記述内容の目的について述べる．4 節では，暗号につ

[1] ©Masazumi Kurihara, Univ. of Electro-Comm. (sol/doc/class/jikken/jikken2022/text/rsatext2022.tex(2022/8/14/14:30))

いての大まかな枠組みについて説明をする．4.1 節では，具体的な暗号の例として，シフト暗号について説明する．4.2 節では，暗号システムの形式的記述の方法について説明する．そして，前節のシフト暗号における「文字のシフト」という操作が「整数の算術演算」として考えることができることを説明する．4.3 節では，二つの暗号システムとして，秘密鍵暗号システム（または、共通鍵暗号システム）と公開鍵暗号システムの二つに大きく分類できることを説明する．本課題で扱う RSA 暗号は，後者の公開鍵暗号システムに属する暗号となる．5 節では，本課題に関係する整数の諸性質について説明をする．この節は，必要に応じて読んで下さい．RSA 暗号の理解、およびプログラム作成に当り，すでに数学的知識が既知の場合は，読み飛ばして構わないです．6 節では，本課題で扱う RSA 暗号について説明をする．7 節では，本課題テーマにおける 7 個の課題内容について説明する．最後に，8 節では，本課題に関するプログラムを作成する際の注意として，プログラミングの準備と参考プログラムについて説明をする．

課題を実施する前に

本課題を実施する前に，まず，1 節の課題の目的と 2 節の提出レポートの各要件を確認して下さい．その上で，7 節の各課題で要求されている内容に対応するプログラムを作成して下さい．そして，2 節の提出レポートに要求されている各項目に対応する内容を記述し，レポートを作成して下さい．

参考のプログラム

課題のプログラムを作成するに当り，参考のプログラムを幾つか用意してあります．それらのプログラム (ソースファイルや実行ファイル) の取得方法や使用方法については，本課題の Web ページの項目「参考資料」→「参考プログラム」に記述してあります．また，課題 1 の RSA 暗号の暗号化，復号化のプログラムを作成するためのヒントを記述した Web ページも「参考資料」→「課題 1 のプログラムを作成するためのヒント (練習プログラム)」に用意してありますので，必要な方は参考にして下さい．

1　課題の目的

1. 暗号方式の一つである RSA 暗号の具体的な暗号化と復号化の方法について理解する．
2. RSA 暗号の暗号化と復号化のプログラムを作成する．具体的には、7 節に示す課題を行う．

2　提出レポートの要件（内容）

2.1　レポートの表紙

レポートは LaTeX やワードなどで作成したものを PDF ファイルに変換し，アップロードされることを想定しています。そのため，本科目の実験で用意されている実験レポート表紙を用いるのではなく、次の項目の情報を記載した 1 ページ目を表紙として作成して下さい．レポートの本文は、２ページ目から記述して下さい．

1. （科目名）202x 情報通信工学実験 B/電子情報学実験 B
2. （実験項目） 情報通信（情報セキュリティ）
3. （実験期間） 202x/xx/xx ～ 202x/xx/xx
4. （提出日） 202x/xx/xx
5. （学籍番号）
6. （氏 名）

2.2 レポートの要件（内容）

レポートは、以下の項目順に記述して下さい。

1. 下記の内容を簡潔にまとめ，レポートせよ．
   (a) RSA 暗号の構成法 (公開鍵と秘密鍵の作成手続き) とその利用方法について説明せよ．
   (b) 公開鍵暗号を利用した認証方式について説明せよ．
   (c) 現時点において，RSA 暗号の鍵の長さ (ビット長) は，どの程度が妥当であると議論されているかを調べ，レポートに記せ．(調査は，ネット検索でよいが，その根拠となる URL やそのタイトルなどを参考文献に明示すること．)
2. 7 節の課題 3 にて計測した実行処理時間の表とグラフを記述する．
3. 7 節の課題 4〜7 に示された問題の解答を記述する．
4. プログラミングで工夫した点：暗号化/復号化のプログラムを作成するに当たり，プログラミングで工夫した点を記述する．
5. 考察：作成したプログラムが、課題として要求されている要件を満たしているのか、妥当なのか、などについて検討する。例えば、メッセージを送信する者、そのメッセージを受け取る者、そして、盗聴者の 3 者がいる場合に、作成した RSA 暗号プログラムは実際に使用できる仕様になっているのか、などについて検討する。その他の作成したプログラムでも、他者が利用することを想定した場合、利用できる、または、利用しやすい仕様になっているのか、などを検討する。
6. 感想とコメント：行った実験全体についての反省点や改善点など、感じたことを記述する．
7. 参考文献：レポートには “参考文献” という節（あるいは項目）を設け，主に参考にした文献を明記する．ここで，文献とは，書籍に限らない．インターネットなどを利用して得られたものも明記すること．その場合，URL とそのページを閲覧した日付を明記する．また，そのページのタイトルがあればそれも明記する．自分のアイデアと他人のアイデアを明確に区別すること．

8. 7 節の課題 1,2,4~7 の「プログラムソース」とその「実行例」[2] を (印刷したものを) レポートに添付すること．読みやすいように、各課題ごとに改ページをして下さい．

   プログラムソースには、注釈やコメントを記入し、プログラムの大まかな説明をすること．

   実行例には、端末（ターミナル）でのプログラムの実行、そして、その出力結果をコピペして記して下さい．書き方は 8.2 節の実行例や Web サイトの「参考プログラム」の記載内容を参照して下さい．ただし，実行回数が多数であったり，出力結果が長い、または、大きいなどレポート作成に適さない場合は，その一部 (または大半) を省略しても構わない．その際は，省略した旨をコメントとして記すこと．

2.3 作成した課題プログラムのソースファイルの提出

レポート（PDF ファイル）とは別に、作成した課題プログラムのソースファイル（拡張子が「 .c」であるファイル）も提出して下さい。複数のソースファイルはアーカイブして、１個のファイルにまとめたものを提出して下さい。アーカイブファイルを作成する方法は、本課題の Web ページの項目「説明用資料」→「複数のファイルをアーカイブし、1 個のファイルにまとめる方法」を参照して下さい．

提出するプログラムソースには、他の人がコンパイルしたり、実行できるように、コメント欄にプログラムソースのコンパイル方法と実行方法を明記すること。教員がそのソースをコンパイルし，実行ファイルを生成し，実際に正しく動作するかなどを確認する場合があります．

注意：プログラムソースを印刷した（PDF にした）ものはレポートに添付することになっています。したがって、ここで指示しているプログラムソースの提出は、拡張子が「.c」 であるファイルを提出して下さい．PDF ファイルがアップロードされていた場合は，プログラムソースは未提出という扱いになりますので注意して下さい．

3 はじめに

本テキストの目的は、本実験課題で扱う RSA 暗号 [5] とよばれる暗号化アルゴリズムの説明およびそのプログラム作成のための諸注意などを説明することにある。

一般に、あるひとつの暗号化アルゴリズムを定義したり、説明する方法は、一通りではなく幾通りかあり、それぞれ趣が異なる場合がある。そこで、本テキストでは、本実験課題の目的を達成するのに都合のよいと思われる方法で種々の定義や説明を与えることにする。もっと一般的な暗号（暗号化）の取り扱いについては、暗号に関連する講義や暗号の専門書の内容に譲ることとする。

本テキストでの暗号に関する説明を記述するにあたり、文献 [1, 2] を参考にした。

4 暗号化

本節では、暗号化の具体例とそれを通して暗号システムの形式的な記述および暗号システムの分類について説明する。

[2] 課題 1 と 2 の RSA 暗号の実行例では、8.2 節の実行例を参照し、次の操作とその結果の内容を必ず記載すること．1) 暗号化の実行、2) 復号化の実行、3) diff コマンドによるひら文ファイルと復号後のひら文ファイルの比較、4) ひら文ファイル、暗号文ファイル、そして、復号後のひら文ファイルのそれぞれのファイルサイズ。

4.1 暗号の具体例（シフト暗号）

はじめに、暗号やそれに関連した数学的準備などの形式的な話しをする前に、具体的な暗号についてみていこう。

A さんから B さんへ伝えたいメッセージを第三者の C さんに見られても分からないようにメッセージを暗号化することを考える。そこで、以下のような暗号を利用することにする。

定義 4.1 メッセージおよびメッセージを暗号化した暗号文はいずれも 26 文字からなるアルファベット, A,B,...,Z, により構成されているものとする。アルファベットの各文字を辞書式順序に並べる。

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

このとき、暗号化は上記の順序に従って各文字を右へ k だけ循環シフトさせる変換に対応させる。このようにして暗号化された暗号文をもとのメッセージに復元するには、暗号文の各文字を左に k だけ循環シフトさせる変換を行なえばよい。 □

この暗号化の様子から上記の暗号を**シフト暗号**ということにする。実際には、上記の暗号を利用して A さんから B さんへ暗号文を送信する前に、k という鍵となる情報を共有しておく必要がある。しかも、この情報は第三者の C さんには知られていないものとする。

例 4.2 例えば、$k = 5$ として、A さんが B さんに伝えたいメッセージ

WEWILLMEETATCHOFUSTATION

を暗号化するとその暗号文は、

BJBNQQRJJYFYHMTKZXYFYNTS

となる。これを B さんに送信すればよい。

第三者の C さんが A さんと B さんの間でシフト暗号を利用することを知っていても、鍵 k の値を知らなければ暗号文からもとのメッセージを読みとることはできない、と言えなくもない。データ伝送の簡単な様子を図 1 に示す。しかし、現実的には、鍵の選択肢は 26 通りしかないのですべての値を試して、復元後の文を読み比べることで暗号文を解読することはそう難しくはないかもしれない。 □

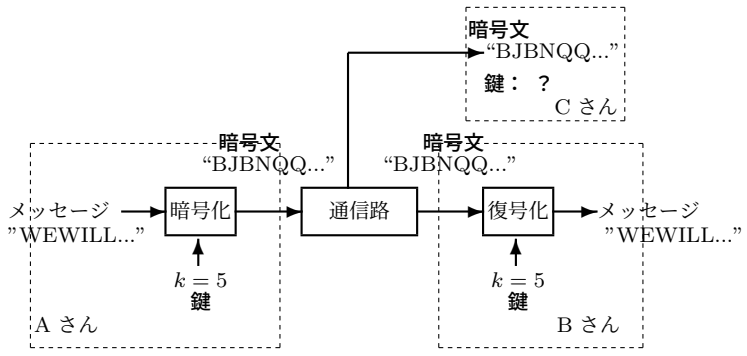

図 1: シフト暗号を用いたデータ伝送の様子

4.2 暗号システムの形式的記述

本節では、前記の具体的な暗号の例を参考に、暗号に関連する定義や術語などを整理する。

暗号の目的は、A さんと B さんの二人の間で盗聴可能な通信路を利用して情報を通信することができることである。ただし、その際、第三者の C さんにはその情報の内容が分からない方法で通信できなければならない。このような盗聴可能な通信路を**公開通信路**ということにする。

伝達したいメッセージや情報を**平文（ひらぶん）**といい、記号 M で表す。暗号化用の鍵 K_e と K_e によって定まる復号化用の鍵 $K_d(= K_{K_e})$ をもつある暗号方法を利用することを考える。ここで、K_{K_e} の意図は、復号化用の鍵が、暗号化用の鍵 K_e によって定まることを示すために、添字に K_e を記述している。このとき、A さんと B さん以外に C さんも利用する暗号方法を既知とするが、暗号化用の鍵 K_e を知るのは A さんのみで、また、復号化用の鍵 K_d を知るのは B さんのみであるとする。

A さんは暗号化用の鍵 K_e を用いて伝達したい平文 M を**暗号文** C に変換し、公開通信路を利用して暗号文 C を B さんに送る。B さんは復号化用の鍵 K_d を用いて受け取った暗号文 C をもとの平文 M に変換し、A さんが伝えたかった情報を理解することができる。一方、C さんは公開通信路を盗聴することで暗号文 C を入手することはできるが、復号化用の鍵 K_d を持たないため暗号文 C からもとの平文 M を得ることは困難で、容易にその内容を知ることはできない。上記の説明の概略を図 2 に示す。このような暗号方法の考え方を少しばかり形式的に以下に示す。

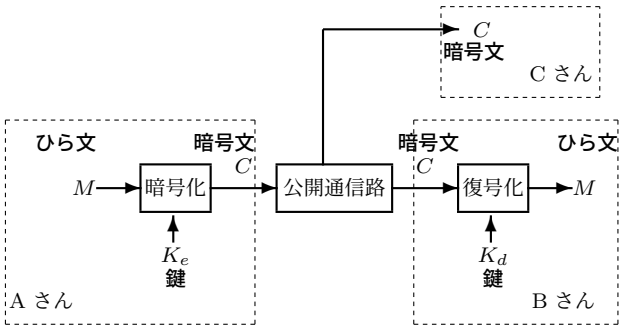

図 2: 暗号システムの概略図

定義 4.3 暗号システムは次の 6 項目を満たす 5 つの要素 $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ から構成されるものと考える（定義しよう）。

1. $\mathcal{M}$：平文全体からなる有限集合
2. $\mathcal{C}$：暗号文全体からなる有限集合
3. $\mathcal{K}$：暗号化鍵全体からなる有限集合
4. $\mathcal{E}$：各鍵 $K \in \mathcal{K}$ に対し定まる暗号化の規則全体からなる集合
   鍵 K による規則 $e_K \in \mathcal{E}$ は $\mathcal{M}$ から $\mathcal{C}$ への写像と考える。
5. $\mathcal{D}$：各鍵 $K \in \mathcal{K}$ に対し定まる復号化の規則全体からなる集合
   鍵 K による規則 $d_K \in \mathcal{D}$ は $\mathcal{C}$ から $\mathcal{M}$ への写像と考える。
6. 任意の暗号化鍵 $K \in \mathcal{K}$ と平文 $M \in \mathcal{M}$ に対し、$d_K(e_K(M)) = M$ を満たす。 □

前記の A さんが鍵 K_e を用いて平文 M を暗号化することは、K_e によって定まる暗号化の規則を用いて平文 M を暗号化することである。また、B さんが復号化用の鍵 K_d を用いて暗号文 C

を復号化することは、K_d によって定まる復号化の規則を用いて暗号文 C を復号化することであるが、しかし、そもそも K_d は暗号化鍵 K_e に依存して定まるものであるから、K_d によって定まる復号化の規則というものは、実は暗号化鍵 K_e によって定まる復号化の規則であると考えてよいことに注意する。

上記の暗号システムの定義にしたがって前節にて説明したシフト暗号を形式的に記述してみよう。まず、準備として、以下の二点について説明する。はじめに、アルファベットの文字を数字に対応させることでシフト処理を算術演算に置き換えることを考える。アスキーコードを 10 進数に変換し、文字と数字の対応を以下のようにする。

文字：	A	B	C	D	E	F	G	H	I	J	K	L	M
整数：	65	66	67	68	69	70	71	72	73	74	75	76	77
26 で割った余り：	13	14	15	16	17	18	19	20	21	22	23	24	25
文字：	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
整数：	78	79	80	81	82	83	84	85	86	87	88	89	90
26 で割った余り：	0	1	2	3	4	5	6	6	7	8	9	10	12

(1)

次に、65 から 90 までの 26 個の整数の集合 $\{65, 66, \ldots, 90\}$ を $\mathbf{Z}_{26}$ で表すことにする。そして、前記表中の 26 で割った余りの整数値より、次に示す $\mathbf{Z}_{26}$ の中での加算と減算の結果は、$\mathbf{Z}_{26}$ の整数になる。このことを、$\mathbf{Z}_{26}$ は加算と減算について閉じているという。任意の $a, b \in \mathbf{Z}_{26}$ に対し、

$$a + b \equiv c \pmod{26}, \qquad \text{ただし, } 65 \leq c \leq 90, \tag{2}$$

$$a - b \equiv c \pmod{26}, \qquad \text{ただし, } 65 \leq c \leq 90. \tag{3}$$

つまり、常に $a+b \in \mathbf{Z}_{26}$ と $a-b \in \mathbf{Z}_{26}$ が成り立つ。以上より、シフト暗号を形式的に記述する：

1. $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbf{Z}_{26}$.
2. $K \in \mathcal{K}$ と $M \in \mathcal{M}$ に対し、$C = e_K(M) \equiv M + K \pmod{26}$.
3. $K \in \mathcal{K}$ と $C \in \mathcal{C}$ に対し、$M = d_K(C) \equiv C - K \pmod{26}$.

ここで、シフト暗号におけるシフト数 k に対応する暗号化鍵 $K \in \mathbf{Z}_{26}$ は、$k \equiv K \pmod{26}$ を満たすものであることに注意する。

暗号システムの定義の中では特に復号化鍵 K_d についての説明はないが、シフト暗号の場合で言えば $K_d = K$ と考えてもさして問題なく、暗号化鍵と一致すると考えればよい。

例 4.4 (続き) メッセージ

WEWILLMEETATCHOFUSTATION

を数字に対応させて変換すると

87 69 87 73 76 76 77 69 69 84 65 84 67 72 79 70 85 83 84 65 84 73 79 78

となる。シフト数 $k = 5$ の場合、鍵は $K = 83$ となり、その暗号文は、

66 74 66 78 81 81 82 74 74 89 70 89 72 77 84 75 90 88 89 70 89 78 84 83

となる (図 3 を参照)。ここで、各文字を数字で表現した場合に「区切り文字」として、空白を挿入していることに注意する。 □

4.3 二つの暗号システム

本節では、幾つかある暗号の分類の仕方の中で、暗号システムを**秘密鍵暗号システム** (または、共通鍵暗号システム) と**公開鍵暗号システム**の二つに分類する場合の説明をする。

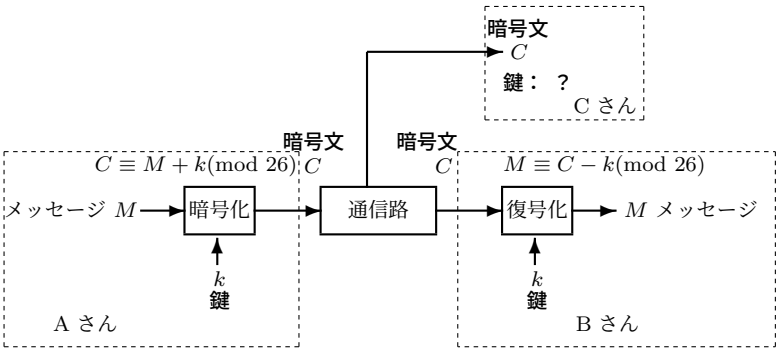

図 3: 算術演算処理で表現した場合のシフト暗号の暗号化と復号化の様子。暗号化は、ひら文 M と鍵 k を用いて、暗号文を $C \equiv M + k \pmod{26}$ として計算する。一方、復号化は、暗号文 C と鍵 k を用いて、ひら文を $M \equiv C - k \pmod{26}$ として計算する。

4.3.1 秘密鍵暗号システム (または、共通鍵暗号システム)

A さんと B さんは暗号化鍵 $K \in \mathcal{K}$ を選ぶことで、K に依存した暗号化規則 e_K と復号化規則 d_K が定まる。このとき、K から d_K を導出する困難さが e_K を導出するのと同程度ならば、K を第三者に公開してしまうと暗号が安全でなくなる。したがって、このような場合、K を第三者には**秘密にする**必要があるため、このような暗号システムを**秘密鍵暗号システム** (または、共通鍵暗号システム) という。前節までに説明したシフト暗号は、この秘密鍵暗号システムの一種である。秘密鍵暗号システムの簡単な概念図は図 4 のようになる。

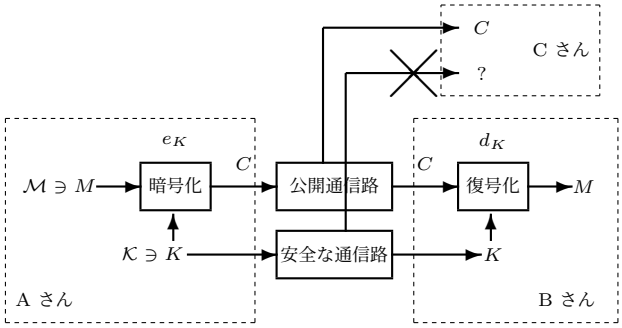

図 4: 秘密鍵暗号システム

秘密鍵暗号システムの問題点は、暗号文を送信する前に第三者には知られないように A さんと B さんの間で暗号化鍵 K を共有できるように**安全な通信路**を利用して送信する必要があることである。しかし、もし、第三者には知られないように暗号化鍵 K を通信し、共有できる方法があれば、平文もそのような方法で通信すればよいことになる。現実的にはそのような安全な通信路を利用することは容易ではないと考えられている。

4.3.2 公開鍵暗号システム

暗号化鍵 K から d_K を導出する困難さが e_K を導出することと比較して計算量的に実行不可能であると考えられる場合、暗号化鍵 K を A さんと B さん以外の第三者にも**公開する**ことができ

る。このことで、事前に K を安全な通信路を利用して通信する必要はなくなる。一方、暗号化鍵 K から直接 d_K を導出することは容易ではない。しかし、公開された暗号化鍵 K とは異なるある秘密の情報となる**秘密鍵** K_d が存在し、その鍵 K_d を用いると比較的容易に d_K を導出することができるならば、B さんのみが秘密鍵 K_d をもつことで、暗号文を復号化することができる。このような暗号システムを**公開鍵暗号システム**という。本課題で扱う RSA 暗号は、この公開鍵暗号システムの一種である。公開鍵暗号システムの簡単な概念図は図 5 のようになる。

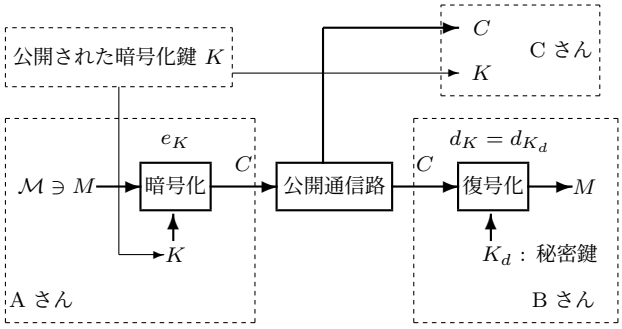

図 5: 公開鍵暗号システム

公開鍵暗号システムの説明をもう少し簡単に説明しよう。秘密鍵暗号システムでは、平文 M を暗号化する暗号化鍵 K_e と暗号文 C を復号化する復号化鍵 K_d は同じものである。そのため、A さんと B さんは鍵 $K_e = K_d$ を第三者に分からないように通信し、共有する必要があった。

一方、公開鍵暗号システムでは、平文 M を暗号化する暗号化鍵 K_e と暗号文 C を復号化する復号化鍵 K_d は異なるものと考えている。そのため、暗号化鍵 K_e を用いても暗号文 C を復号化することはできなく、復号化できるのは復号化鍵 K_d を用いた場合だけであると考えている。したがって、鍵を共有する必要はない。

そこで、暗号文を受け取る側の B さんは、B さん専用の**暗号化鍵**と**復号化鍵**を作成し、暗号化鍵を**一般に公開し**、復号化鍵を**秘密**にしておく。そして、A さんは B さんに伝えたい平文を公開されている暗号化鍵を用いて暗号文を作成し、公開通信路を利用して B さんに送信すればよい。第三者の C さんは、公開されている暗号化鍵と公開通信路を利用して暗号文を入手できるが、その二つだけでは暗号文を元の平文に容易には戻せないということである。このとき、A さん自身も復号化鍵を知らないので自分で作成した暗号文を元の平文へは復元できない。したがって、A さんが作成した暗号文を復号化できるのは復号化鍵をもつ B さんのみだけである。これで暗号の目的である情報の通信が可能であると考える訳である。

5　数学的準備（整数の諸性質）

本節は、読み飛ばしても構わない。本節では、次節で説明する RSA 暗号で必要となる整数に関する諸性質について簡単にまとめたものを説明する。しかし、まずは、本節を読み飛ばし、RSA 暗号の説明で分からない数学的用語などがある場合にその都度、本節に戻って参考にすればよいと思う。

では、以下に整数に関連する事実をおさらいする。

5.1 除法の定理

定理 5.1 a, b を整数とし、$b > 0$ とする。そのとき、

$$a = bq + r, \quad 0 \leq r < b$$

を成り立たせる整数 q と r がたかだか 1 組だけ存在する。 □

たとえば、$(a, b) = (57, 17)$ のとき、$(q, r) = (3, 6)$ となる。また、$(a, b) = (-57, 17)$ のとき、$(q, r) = (-4, 11)$ となる。

上記の q, r をそれぞれ a を b で割った**整商**、**余り**（または**剰余**）という。詳しくは、r は a を b で割った負でない最小剰余という。$r = 0$, すなわち $a = bq$ となる場合には、a は b で割り切れるという。

5.2 最大公約数

a, b を 2 つの整数とする。もし、$a = bq$ となる整数 q が存在するならば、a は b で割り切れる、b は a を割り切るという。またこのとき、a は b の倍数、b は a の**約数**という。このことを記号で $b|a$ と書く。

a_1, a_2 を 2 つの整数とし、0 でないとする。a_1 と ,a_2 の両方を割り切る整数を a_1, a_2 の**公約数**とよぶ。d が正の公約数で、さらに、次の性質をもつとき、d は a_1, a_2 の**最大公約数**とよぶ。

「e を a_1, a_2 の任意の公約数とすれば、$e|d$ である。」

最大公約数は（もし存在すれば）一意に定まる。それでは、最大公約数が存在することは次の定理で保証される。

定理 5.2 a_1, a_2 を 2 つの整数とし、0 でないとする。x_1, x_2 を任意の整数として

$$x_1 a_1 + x_2 a_2$$

の形に表される整数全部の集合を J とし、J に含まれる最小の正の元を d とする。そのとき、d は a_1, a_2 の最大公約数である。また、J は d のすべての倍数の集合と一致する。 □

a_1, a_2 の最大公約数を、greatest common divisor の頭文字を用いて、$\gcd(a_1, a_2)$ と表す。この定義より、明らかに、$\gcd(a_1, a_2) = \gcd(a_2, a_1)$ であることに注意する。

系 5.3 a_1, a_2 の最大公約数を d とすれば、

$$d = u_1 a_1 + u_2 a_2$$

となるような整数 u_1, u_2 が存在する。 □

2 つの整数の最大公約数を求める方法としては、Euclid 法がある。

補題 5.4 整数 a, b の差 $a - b$ が $m(\neq 0)$ で割り切れるならば、

$$\gcd(a, m) = \gcd(b, m)$$

である。 □

この補題にもとづいて Euclid 法 は以下のように説明できる。

アルゴリズム 5.5 (Euclid 法) $a \geq b$ と仮定し、次のような等式の系列をつくる：

$$\begin{aligned} a &= bq_1 + r_2, & 0 < r_2 < b \\ b &= r_2q_2 + r_3, & 0 < r_3 < r_2 \\ r_2 &= r_3q_3 + r_4, & 0 < r_4 < r_3 \\ &\vdots & \vdots \\ r_{n-2} &= r_{n-1}q_{n-1} + r_n, & 0 < r_n < r_{n-1} \\ r_{n-1} &= r_nq_n \end{aligned} \tag{4}$$

補題 5.4 より、

$$\gcd(a,b) = \gcd(b,r_2) = \gcd(r_2,r_3) = \cdots = \gcd(r_{n-1},r_n)$$

となるが、$r_n | r_{n-1}$ であるから $\gcd(r_{n-1},r_n) = r_n$. したがって、最後の除数 r_n が a,b の最大公約数である。ゆえに、$\gcd(a,b) = r_n$ となる。これが Euclid 法 である。 □

整数 a,b に対して $\gcd(a,b) = 1$ であるとき、a,b は**互いに素**であるという。

例 5.6 $(a,b) = (144,5)$ のとき、以下の計算より $\gcd(144,5) = 1$ となる。

$$\begin{aligned} 144 &= 5 \cdot 28 + 4, & 0 < 4 < 5 \\ 5 &= 4 \cdot 1 + 1, & 0 < 1 < 4 \\ 4 &= 1 \cdot 4 \end{aligned} \tag{5}$$

また、$(a,b) = (144,60)$ のとき、以下の計算より $\gcd(144,60) = 12$ となる。

$$\begin{aligned} 144 &= 60 \cdot 2 + 24, & 0 < 24 < 60 \\ 60 &= 24 \cdot 2 + 12, & 0 < 12 < 24 \\ 24 &= 12 \cdot 2 \end{aligned} \tag{6}$$

□

Euclid 法を用いることで、a,b の最大公約数 $\gcd(a,b) = d$ を求めることができた。このとき、系 5.3 より a,b,d はある整数 f,g を用いて次のような関係式で表される。

$$fa + gb = d$$

例えば、$(a,b) = (144,60)$ の場合、$\gcd(144,60) = 12$ であるが、これは、$(f,g) = (-2,5)$ として、

$$-2 \cdot 144 + 5 \cdot 60 = 12$$

と書くことができる。また、$(a,b) = (144,5)$ の場合、$\gcd(144,5) = 1$ であるが、これは、$(f,g) = (-1,29)$ として、

$$-1 \cdot 144 + 29 \cdot 5 = 1$$

と書くことができる。

整数 a,b に対して、その最大公約数と同時にこのような f,g を求める方法を**拡張 Euclid 法**という。この方法を以下に説明する：

アルゴリズム 5.7 (拡張 Euclid 法) アルゴリズムへの入力は、a, b である。ただし、$a > b$ とする。初期値として、

$$\begin{aligned} r_{-1} &:= a,\ f_{-1} := 1,\ g_{-1} := 0 \\ r_0 &:= b,\ f_0 := 0,\ g_0 := 1 \end{aligned}$$

とする。

このとき、各 $i \geq 1$ に対して、r_{i-2} を r_{i-1} で割ったときの整商 q_i と余り r_i を計算する：

$$r_{i-2} = r_{i-1}q_i + r_i, \quad 0 < r_i < r_{i-1}$$

そして、f, g を更新する：

$$\begin{aligned} f_i &:= f_{i-2} - q_i f_{i-1} \\ g_i &:= g_{i-2} - q_i g_{i-1}. \end{aligned}$$

r_i は単調減少しているために必ず終了が来る。ここで、$r_n \neq 0$ とする。このとき、$\gcd(a, b) = r_n$ と $f_n a + g_n b = r_n$ を得ることができる。 □

例 5.8 $(a, b) = (144, 60)$ の場合：

i	f_i	g_i	r_i	q_i
-1	1	0	144	−
0	0	1	60	−
1	1	-2	24	2
2	$\underline{-2}$	$\underline{5}$	$\underline{12}$	2
3	5	-12	0	2

(7)

したがって、$\gcd(144, 60) = 12$ と $f_2 a + g_2 b = d \to \underline{-2} \cdot 144 + \underline{5} \cdot 60 = \underline{12}$ を得る。 □

例 5.9 $(a, b) = (144, 5)$ の場合：

i	f_i	g_i	r_i	q_i
-1	1	0	144	−
0	0	1	5	−
1	1	-28	4	28
2	$\underline{-1}$	$\underline{29}$	$\underline{1}$	1
3	5	-144	0	4

(8)

したがって、$\gcd(144, 5) = 1$ と $f_2 a + g_2 b = d \to \underline{-1} \cdot 144 + \underline{29} \cdot 5 = \underline{1}$ を得る。 □

5.3 最小公倍数

正数 a, b の倍数である正数を、a と b の公倍数であるという。a と b の正の公倍数のうちに最小の数 ℓ が存在する。それを a と b の最小公倍数 (least common multiple) といい、$\mathrm{lcm}(a, b)$ と記す。

2 つの正数の最大公約数と最小公倍数に関する次の定理を示す。

定理 5.10 2 つの正の整数 a, b の最小公倍数 $\mathrm{lcm}(a, b)$ は、積 ab を最大公約数 $\gcd(a, b)$ で割ったものに等しい。つまり、

$$\mathrm{lcm}(a, b) = \frac{ab}{\gcd(a, b)}. \tag{9}$$

□

5.4 素数

a を 1 より大きい整数とすれば、a は少なくとも 2 つの正の約数 1 と a をもつ。a がこれら以外に正の約数をもたないとき、a を**素数** (prime number) という。たとえば、$2, 3, 5, 7, 11, 13, \ldots$ である。素数でない整数 ≥ 2 は合成数とよばれる。

命題 5.11 B を整数 > 1 とする。そのとき、任意の正の整数 a は、

$$a = c_k B^k + c_{k-1} B^{k-1} + \cdots + c_1 B^1 + c_0 B^0 \tag{10}$$

$0 \leq c_0 < B$, $0 \leq c_1 < B$, ...,$0 \leq c_k < B$, の形に一意に表される。この表現を a の B **進展開**という。 □

5.5 合同関係

整数全体の集合 $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$ を $\mathbf{Z}$ で表す。同値関係の具体例となる $\mathbf{Z}$ における合同関係について説明する。

m を 1 つの与えられた正の整数とする。a, b を 2 つの整数とするとき、もし $a - b$ が m で割り切れるならば、a, b は m を**法** (modulo) として（あるいは法 m に関して）合同であるという。このことを記号で

$$a \equiv b \pmod{m}$$

と書く。

次に、m を法とする合同関係による $\mathbf{Z}$ の類別について考える。この場合の各類は法 m に関する剰余類とよばれる。1 つの剰余類は m を法として互いに合同な数全体の集合である。すなわち、a をその剰余類の代表とすれば、$a + mt$ の形に表される整数全体の集合である。

任意の整数 a は

$$a = mq + r, \quad 0 \leq r < m$$

の形に一意に表される。したがって、a は $0 \leq r < m$ であるような 1 つしかもただ 1 つの r と、m を法として合同となる。いいかえれば、任意の整数 a は、それぞれ $0, 1, \ldots, m-1$ を代表とする m 個の剰余類のいずえれか 1 つ、しかもただ 1 つだけに含まれる。ゆえに、法 m に関する剰余類は全部でちょうど m 個存在する。

例 5.12 $m = 3$ とすれば、法 3 に関して $\mathbf{Z}$ は 3 個の剰余類に分割される。それらは、3 の倍数全部の集合、3 で割ると 1 余る数全体の集合、 3 で割ると 2 余る数全体の集合である。 □

定理 5.13 m_1, m_2 を互いに素な正の整数とし、b_1, b_2 を任意の整数とする。そのとき、連立合同式

$$x \equiv b_1 (\mathrm{mod}\ m_1) \tag{11}$$

$$x \equiv b_2 (\mathrm{mod}\ m_2) \tag{12}$$

は、積 $m = m_1 m_2$ を法としてただ 1 つの解をもつ。 □

5.6 剰余環（商環） $\mathbf{Z}_m$

整数 m に対し，集合 $\mathbf{Z}_m$ を $\{0, 1, 2, \ldots, m-1\}$ と定義する．このとき，集合 $\mathbf{Z}_m$ 上に m を法とする加法 $+$ と乗法 $\times$ を定義することで，集合 $\mathbf{Z}_m$ は、 0 を零元、1 を単位元とする m 個の元からなる有限環となる．

6 RSA 暗号

本節では、RSA 暗号について説明する。

6.1 RSA 暗号の暗号化と復号化

1. RSA 暗号での演算は、整数における n を法 (modulo) とした合同関係に基づく演算である。ここで、n は二つの大きな素数 p と q の積 $n = pq$ である。
2. 暗号化と復号化は、n を法とした合同関係に基づく次のべき乗の計算である。
   与えられた暗号化鍵 (公開鍵 (publick key) という) $(e, n) = K_{pub}$ と復号化鍵 (秘密鍵 (private key) という) $(d_e, n) = K_{pr}$ に対し、それぞれ次のように定める。

$$\text{暗号化：} \quad C = e_{K_{pub}}(M) \equiv M^e \pmod{n} \tag{13}$$

$$\text{復号化：} \quad M = d_{K_{pr}}(C) \equiv C^{d_e} \pmod{n} \tag{14}$$

   ここで、ひら文 M と暗号文 C は、$0 \leq M \leq n-1$ と $0 \leq C \leq n-1$ を満たす整数である。そして、$e_{K_{pub}}(\ )$ を鍵 K_{pub} による暗号化、$d_{K_{pr}}(\ )$ を鍵 K_{pr} による復号化という。

6.2 RSA 暗号の鍵の生成

はじめに、次のようないくつかの整数を選択したり、計算したりする。

1. 互いに異なる任意の素数 p, q を選び、その積を計算する。

$$n := pq \tag{15}$$

2. $(p-1)$ と $(q-1)$ の最小公倍数 (least common multiple) L を計算する.

$$L := \mathrm{lcm}(p-1, q-1) \tag{16}$$

3. L と互いに素で L より小さな任意の整数 e を選ぶ。[3]

$$\gcd(e, L) = 1, \quad 1 < e < L \tag{17}$$

4. 次式を満たす整数 d_e を計算する。

$$ed_e \equiv 1 \pmod{L}, \quad 1 < d_e < L \tag{18}$$

   このような d_e を求める方法として拡張 Euclid 法 (アルゴリズム 5.7) を利用すればよい。その理由は、例 6.1 を参照。

以上の手続きで得られた整数値より、暗号化鍵は $(e, n) = K_{pub}$、復号化鍵は $(d_e, n) = K_{pr}$ となる。

[3] 文献 [5] では，整数 e を選ぶ際に，$(p-1)$ と $(q-1)$ の最小公倍数 L ではなく，$(p-1)$ と $(q-1)$ の積 $(p-1)(q-1)$ を用いて次のように選んでいる：$\gcd(\ e, (p-1)(q-1)\ ) = 1, \quad 1 < e < (p-1)(q-1)$. しかし，本稿では，文献 [2] にしたがい，式 (16),(17) のように選ぶようにする.

6.3 RSA 暗号の形式的記述と保証および具体的な計算について

前節まで説明より、RSA 暗号の形式的な記述は以下のようになる。

1. 任意の互いに異なる素数 p, q に対し、 $n := pq$ とする。
2. $\mathcal{M} = \mathcal{C} = \mathbf{Z}_n = \{0, 1, \ldots, n-1\}$.
3. $\mathcal{K} = \{(e, n) \mid \gcd(e, L) = 1 \text{ and } 1 < e < L\}$ where $L = \mathrm{lcm}(p-1, q-1)$.
4. d_e： $ed_e \equiv 1 \pmod{L}$ を満たす整数。
5. $K_{pub} = (e, n) \in \mathcal{K}$ と $M \in \mathcal{M}$ に対し、$C = e_{K_{pub}}(M) \equiv M^e \pmod{n}$.
6. $K_{pr} = (d_e, n) \in \mathcal{K}$ と $C \in \mathcal{C}$ に対し、$M = d_{K_{pr}}(C) \equiv C^{d_e} \pmod{n}$.
7. 暗号化鍵 (e, n) を公開し、p, q, d_e の値を秘密にする。d_e が秘密鍵で (d_e, n) が復号化鍵となる。

以上のことを図示したものが、図 6 となる。

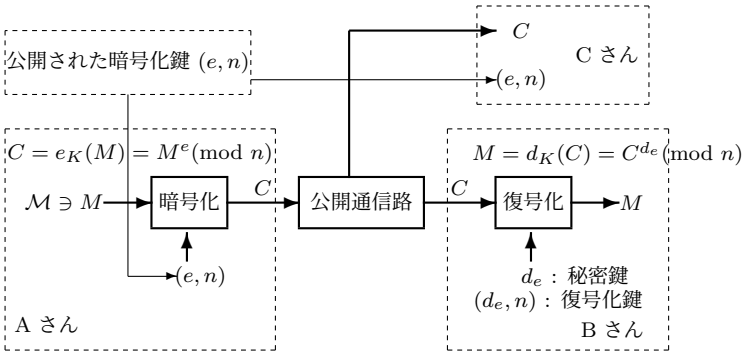

図 6: RSA 暗号を用いた公開鍵暗号システム

RSA 暗号の暗号化規則 $e_{K_{pub}}$ と復号化規則 $d_{K_{pr}}$ の合成写像は、$d_{K_{pr}}(C) = d_{K_{pr}}(e_{K_{pub}}(M)) = M$ となる。つまり、

$$C^{d_e} = (M^e)^{d_e} = M^{ed_e} \equiv M \pmod{n} \tag{19}$$

RSA 暗号において、この関係の成立が保証されていることが大切である。

公開されている情報 (e, n) から整数 d_e を求めることで RSA 暗号を解読することを考える。d_e の定義からその値を容易に求めるには二つの素数 p, q の値を求める必要がある。そのためには整数 n を素因数分解すればよい。しかし、この整数 n の素因数分解を完了することは計算量的に実行不可能であるという考えに基づくと、整数 d_e を求めることで RSA 暗号を解読することは困難であるというになる。したがって、d_e を求めるという解読において、RSA 暗号の安全性は、 素因数分解の困難さに基づいているといえる。

例 6.1 与えられた整数 e, L は, $\gcd(e, L) = 1 \text{ and } 1 < e < L$ を満たすとする。このとき、条件: $ed_e \equiv 1 \pmod{L}$ を満たす整数 d_e は、拡張 Euclid 法を用いて計算できることを示す。

まず、拡張 Euclid 法に (e, L) を入力すると、出力として、次の f, g, d が求まる: $f \cdot e + g \cdot L = d$. このとき、$\gcd(e, L) = 1$ より、$d = 1$ となる。

次に、式 $f \cdot e + g \cdot L = d$ を L を法として考えると

$$f \cdot e + g \cdot L \equiv 1 \pmod{L} \tag{20}$$

$$f \cdot e \equiv 1 \pmod{L} \tag{21}$$

何故なら、左辺では、$g \cdot L \equiv 0 \pmod{L}$ であるから。このとき、f は、上記条件を満す整数 d_e になる。ただし、拡張 Euclid 法の結果から得られる f は正整数とは限らない。正整数として処理したい場合は、L を法として考えればよいのは明かである。

ゆえに、与えられた整数 e, L に対し、拡張 Euclid 法を用いることで、秘密鍵 d_e を計算することができる。 □

例 6.2 先の例で取り上げた 26 文字のアルファベットからなる平文を RSA 暗号で暗号化することを考える。

文字数は 26 個であるから選択すべき二つの素数 p, q は、$n = pq \geq 26$ を満たす必要がある。そこで、$(p, q) = (17, 19)$ とすることで、$n = 323$ となる。

次に、$(p-1)$ と $(q-1)$ の最小公倍数は、$L = 144$ となる。$L = 144$ に対し、$1 < e < 144 = L$ を満たし、互いに素なる整数 e として 5 を選択する。

そして、拡張 Euclid 法に $(e, L) = (5, 144)$ を入力し、秘密鍵 d_e を計算すると $d_e = 29$ となる。

以上より、公開鍵（符号化鍵）は $K = (e, n) = (5, 323)$, 秘密鍵は $d_e = 29$, 復号化鍵は $(d_e, n) = (29, 323)$ となる。

たとえば、文字 “ C ” に対応する数字は “ 67 ” であるから、これを暗号化すると、$(67)^5 \equiv 288 \pmod{323}$ より、$e_K(67) = 288$ となる。一方、これを復号化すると、$(288)^{29} \equiv 67 \pmod{323}$ より、$d_K(288) = 67$ となり、確かにもとの平文 “ C ” と一致する。

このとき、$(67)^5$ を一度に計算する必要はなく、以下のように計算することが可能である。

1) $(67)^2 = 67 \times 67 = 4489 \equiv 290 \pmod{323}$,

2) $(67)^3 = (67)^2 \times 67 \equiv 290 \times 67 = 19430 \equiv 50 \pmod{323}$,

3) $(67)^4 = (67)^3 \times 67 \equiv 50 \times 67 = 3350 \equiv 120 \pmod{323}$,

4) $(67)^5 = (67)^4 \times 67 \equiv 120 \times 67 = 8040 \equiv 288 \pmod{323}$.

つまり、(mod 323) において扱う整数は 0 から $322 \times 322 = 103684$ までで十分である。ゆえに、103684 以下の整数計算を正しくできる電卓があれば一応手計算でも暗号化を行なうことができるということ。計算量を考慮した場合、べき乗の計算には、例 6.3 に示すような工夫をすることができる。

それでは、メッセージ WEWILLMEETATCHOFUSTATION を暗号化する。まず、各文字を数字に変換すると

87 69 87 73 76 76 77 69 69 84 65 84 67 72 79 70 85 83 84 65 84 73 79 78

であるから、二桁ずつ数字を取り出して暗号化の計算をすると暗号文

83 103 83 99 247 247 229 103 103 50 12 50 288 21 129 185 187 87 50 12 50 99 129 10

を得る。ここで、各文字を数字で表現した場合に「区切り文字」として、空白を挿入していることに注意する。さて、A さんは、この暗号文を B さんに送信すればよい。そして、 B さんは自分しか知らない秘密鍵を含む復号化鍵 $(d_e, n) = (29, 323)$ を用いて暗号文を復号化する。 □

$(p, q) = (17, 19)$ 程度の大きさの値でも、暗号化や復号化の計算をすべて電卓で計算するのは容易ではない．そこで、本課題の目的の一つとして、RSA 暗号の暗号化および復号化プログラムを書くことにする．

最後に，べき乗の計算を効率よく行うアルゴリズム，2 進展開法，を説明しておく．

例 6.3 (2 進展開法)

整数 M と e に対するべき乗 M^e の計算を素直に実行すれば、$M^e = \underbrace{M \times M \times \cdots \times M}_{e}$ のように $e-1$ の乗算回数を実行する必要がある。このべき乗の計算を約 $2\log_2 e$ の乗算回数で実行する

アルゴリズムを示す。

ポイントは、整数 e の 2 進展開

$$e = \sum_{i=0,1,2,\ldots} e_i 2^i = e_0 + e_1 2 + e_2 2^2 + \cdots$$

を利用する。このとき、

$$M^e = M^{e_0 + e_1 2 + e_2 2^2 + \cdots} = \prod_{i=0,1,2,\ldots} M^{e_i 2^i}$$

と書けることに注意する。さらに、

$$\begin{aligned} M^{2^0} &= M \\ M^{2^1} &= (M^{2^0})^2 \\ M^{2^2} &= (M^{2^1})^2 \\ M^{2^3} &= (M^{2^2})^2 \\ &\vdots \end{aligned}$$

より、$M^{2^{i+1}} = (M^{2^i})^2$ が成り立つ。以上より、整数 M と e に対し、$x = M^e$, を求めるアルゴリズム (Input: (M, e), Output: $x = M^e$) を以下に示すことができる.

注意: RSA 暗号の符号化における 2 進展開法を用いた計算では，オーバーフローを発生させないようにするために，ステップ 3 と 4 の乗算 $x := x \times y$ と $y := y \times y$ では必ず modulo での計算を行う。つまり、n で割った余りを代入すること。

1. Initial setting: $t := e$, $x := 1$, $y := M$.
2. Compute $q := \lfloor \frac{t}{2} \rfloor$ and $r := t - 2 \times q$.
3. IF $r = 1$ THEN $x := x \times y \pmod n$.
4. IF $q = 0$ THEN output x and halt,
   ELSE compute $y := y \times y \pmod n$, set $t := q$, and goto step 2.

たとえば、整数の組 $(M, e) = (5, 11)$ に対するべき乗 $M^e = 5^{11}$ の実行例は以下のようになる。ここで、11 の 2 進展開は $11 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3$ となり、変数 r の値 (の並び) が 11 の 2 進系列を表している。下位ビットが左側、上位ビットが右側に対応する。

	初期値							
q	–	5	$\rightarrow$	2	$\rightarrow$	1	$\rightarrow$	0
r	–	1		1		0		1
t	11	5		2		1		–
x	1	$5 = 5^{2^0}$		$125 = 5^{2^0+2^1}$		$125 = 5^{2^0+2^1}$		$48828125 = 5^{2^0+2^1+2^3} = 5^{11}$
y	$5 = 5^{2^0}$	$25 = 5^{2^1}$		$625 = 5^{2^2}$		$390625 = 5^{2^3}$		–

□

例 6.4 (2 進展開法)　例 6.3 において、modulo 計算を考慮した場合の例として，パラメータを $(M, e, n) = (65, 29, 323)$ とした場合の計算途中過程を以下に示す．$65^{29} \pmod{323} \equiv 107$. 作成するプログラムの動作確認に利用して下さい．ここで、$29 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4$

である。

	初期値									
q	–	14	→	7	→	3	→	1	→	0
r	–	1		0		1		1		1
t	29	14		7		3		1		–
x	1	65		65		12		141		107
y	65	26		30		254		239		–

□

7 課題

RSA 暗号の理解とプログラミングに関連する課題 1~7 を以下に示す．また，課題 4~7 には，作成するプログラムの指示以外に問題も示す．作成した課題プログラムを利用して，問題の解を求めレポートに解答しなさい．問題を解くためにプログラムを修正した場合，その修正したプログラムをレポートとして提出する必要はない．その理由は，課題で要求されたプログラムからの出力結果をデータ処理することで解答を得ることができると想定しており，プログラムの修正までは必要ないと考えているため．

1. RSA 暗号の暗号化および復号化のプログラムを作成せよ。具体的には, 以下のような暗号化および復号化の仕様を満たすこと。[4]

暗号化:

入力	： 暗号化鍵 (e, n)，ひら文ファイル.
出力	： 暗号化ファイル.

復号化:

入力	： 復号化鍵 (d_e, n)，暗号化ファイル.
出力	： 元のひら文ファイル.

以下の (a)~(c) を実行、確認し，その結果をレポートに記述せよ．(本資料の 21 ページの 8.2 節 参考プログラム の「1. RSA 暗号の暗号化（符号化）と復号化プログラムの実行例」の実行内容を参考にせよ．)

(a) 用意されているデータファイル 256byte.dat をひら文ファイルとして用いて，暗号化鍵 $(9929, 32399)$, 復号化鍵 $(9329, 32399)$ で正しく動作するか確認せよ[5]。(鍵は、$(p, q) = (179, 181)$ を用いて作成したものである。)

(b) diff コマンドを用いて、ひら文ファイルと復号したファイルが一致することを確認せよ。

(c) ひら文ファイル，暗号化したファイル，復号したファイルの 3 種類のファイルサイズを比較せよ。

[4]RSA 暗号の暗号化や復号化で行われるべき乗の計算では、大きな値の整数を扱うために、C 言語で用意されているべき乗を計算する関数を用いた場合、正しく計算できない可能性がある。そこで、例 6.2 にて示したように、M の e 乗のべき乗 M^e を $e-1$ 回の乗算に分解することを考える。ただし、単純に、$e-1$ 回の乗算を for 文で繰り返しただけでは、やはり、大きな値の整数になり、正しく計算ができない可能性がある。そのために、1 回の乗算ごとに、(mod n) の計算をすることで、毎回の計算結果は n より小さくなる。この事実を利用した方法でべき乗の計算をするプログラムを作成することが大切と思います。

[5]データファイル 256byte.dat は、他の参考プログラムと同じ場所からコピーできる。そして，テスト用のデータファイル 256byte.dat は，1 バイトで表現できる 256 通りのすべての文字（ビット列）が記述されているファイルである．このデータファイルで暗号化と復号化が正しく動作すれば、どのようなファイルでも正しく動作するプログラムを作成できていると考えることができる．

2. 例 6.3 に示した 2 進展開法を採用した RSA 暗号のプログラムを作成せよ。
   前記の課題 1 と同様に，上記の (a)∼(c) を実行，確認し，その結果をレポートに記述せよ．

3. 課題 1 と 2 において作成したプログラムに対し，次の 10 通りの e の値 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 に対し，それぞれの実行処理時間を計測し，表にまとめ，それをグラフに示せ．

   (a) 実行処理時間を計測する方法は，本資料の 22 ページの 8.2 節 参考プログラム の「2. 実行処理時間の計り方」の実行内容を参考にせよ．

   (b) 例 6.3 に示したように，2 進展開法を採用した RSA 暗号プログラム (課題 2) の場合，その実行処理時間は，鍵のサイズ e に対し $O(\log_2(e))$ となっているかを確認し，その結果をレポートに記述せよ．2 進展開法を採用したプログラムの場合，実行する計算機の性能とデータファイルのサイズの兼ね合いで，e の値を変化させても，log 関数を表すような明確なデータを得ることが難しい．そこで，現時点の CED の計算機では，実測の際，テスト用のデータファイルとして 2Mbyte 程度のファイル名 2Mbyte.dat のデータファイルを用いると，log 関数を表すようなデータを得ることができるはずである (計算機の性能に依存する)．[6]

4. 1 から 10000 までの間に存在する素数をすべて求めるプログラムを作成せよ。

   以下の問題 (a),(b) を調べ，レポートに記述せよ．

   (a) $2 \leq p \leq 7816$ を満たす素数 p の個数を解答せよ．

   (b) 2 を 1 番目の素数とすると，797 番目の素数の値を解答せよ

5. 異なる素数 p と q に対し、$p-1$ と $q-1$ の最小公倍数 $L = \mathrm{lcm}(p-1, q-1)$ を求めるプログラムを作成せよ。(本資料の 12 ページに記した拡張 Euclid 法 (アルゴリズム 5.7) を用いて作成することで，下記の 6, 7 のプログラムも容易に作成できる (例 6.1 を参照)．)

   問題: $(p, q) = (131, 179)$ と設定した場合の最小公倍数 $L = \mathrm{lcm}(p-1, q-1)$ を求め，レポートに記述せよ．

6. 整数 L に対し、$\gcd(e, L) = 1$ かつ $1 < e < L$ を満たすすべての整数 e を求めるプログラムを作成せよ。

   問題: $L = 10212$ と設定した場合，$\gcd(e, L) = 1$ かつ $1 < e < L$ を満たす整数 e は何個存在するかを調べ，レポートにその個数を解答せよ．

7. 整数 L と e に対し、$ed \equiv 1 \pmod{L}$ かつ $1 < d < L$ を満たす整数 d を求めるプログラムを作成せよ。

   問題: $L = 10212$, $e = 9191$ と設定した場合，$ed \equiv 1 \pmod{L}$ かつ $1 < d < L$ を満たす整数 d を求め，レポートにその値を解答せよ．

[6] データファイル 2Mbyte.dat は、他の参考プログラムと同じ場所からコピーできる。

8 プログラミング

8.1 プログラミングの準備

計算機上でプログラミングを行なうことで RSA 暗号の暗号化・復号化を実行させることを考える。ここでは、C 言語における整数型 int で、4 バイト (32 ビット) を使って整数を扱うという立場で、プログラミングの方針および注意事項などを箇条書の形で列挙する。また、多倍長計算に関してはここでは特に扱わないとする。

1. ここでのプログラミングの目標は、次の 2 つに大別される。一つ目は、暗号化で、暗号化鍵を用いて”foo” というファイル名の平文のデータを RSA 暗号で暗号化し、その暗号文を”goo” というファイル名のファイルに書き出す暗号化プログラムを作成すること。二つ目は、復号化で、復号化鍵を用いて暗号化されたデータのファイル”goo” を復号化し、もとの平文のデータに戻し、 “hoo” というファイル名のファイルに書き出す復号化プログラムを作成すること。プログラムが正しく動作しているかは、二つのファイル foo と hoo が一致することを diff コマンドで確認する。ここに示したファイル名に意味はない。

2. 暗号化の具体的な方法としては、foo というファイルから 1 バイト毎にデータを読み取り、暗号化し、goo というファイルに暗号文を書き出す。これをファイル foo の EOF を読み取るまで繰り返し、終了する。このとき、1 バイトは 8 ビットの 0 と 1 の列である。そこで、1 バイトのデータを文字として処理するのではなく、それらを 0 から $255 = 2^8 - 1$ までの整数値として処理することで暗号化を行なう。

3. したがって、平文 M は $0 \leq M \leq 255$ であるから、$255 < n = pq$ を満たすような素数 p, q を選択する必要がある。

4. もし、$n > 255$ ならば、暗号文 C は 256 の以上の整数値をとる場合があることに注意する。つまり、1 バイトの平文 M を暗号化すると 1 バイトでは表現できない整数値になり、暗号文 C を記録するには 2 バイト以上の枠を考える必要がある。

   例えば、先の例 6.2 において、文字 “C” に対応する整数値 67 は、暗号化で $e_K(67) = 288$ となる。整数値 288 を表すには、2 バイトが必要である。

5. 具体的にプログラミングを行なうには、利用するプログラミング言語で扱える整数値の大きさ (範囲) を考慮する必要がある。C 言語での整数型 int は、4 バイト (32 ビット) を使う符号付き整数型となり、正の整数は $2^{31} - 1$ までしか正しく扱うことができない。つまり、32 ビットのうち上位 1 ビットを符号用に利用するため符号以外の部分で利用できるのは 31 ビットである。

6. 復号化のことを考慮すると、少なくとも n^2 までの整数値を正しく扱い、計算するできる必要がある。大雑把に計算すると、$n \leq 2^{15} - 1$, すなわち、n が 15 ビット以内で表現できる正の整数値ならば、$n^2 < 2^{30} < 2^{31} - 1$ を満たす。

7. したがって、$2^8 = 256 \leq n \leq 2^{15} - 1$ を満たすように p, q を選択すればよさそうである。

   たとえば、$(p, q) = (29, 31)$ とすると、$2^4 < 29 < 2^5$ かつ $2^4 < 31 < 2^5$ であるから、$2^8 < 2^9 = 512 < n = pq = 899 < 1024 = 2^{10}$ となる。したがって、プログラミングの具体的な (p, q) の例として $(29, 31)$ を利用することができる。一方、$(p, q) = (241, 251)$ とすると、

$2^7 < 241 < 2^8$ かつ $2^7 < 251 < 2^8$ であるから、$2^{14} < 2^{15} = 32768 < n = pq = 60491 < 65536 = 2^{16}$ となる。ゆえに、プログラミングの具体的な (p, q) の例として $(241, 251)$ を利用することは適切ではない。

8. たとえば、$(p, q) = (29, 31)$ を選択した場合、$n = pq = 899 > 2^9$ であるから 1 バイトの平文 M に対する暗号文 C は少なくとも 2 バイトないと表現できない数である。ここでは、プログラミングを容易にするためにデータの取扱いをビット単位ではなくバイト単位で扱うことを考えている。一方、2 バイトあれば十分でもある。このとき、暗号化プログラムは 1 バイトの平文を読み込み、暗号化する毎に常に 2 バイトの暗号文を書き出すようにする。このようにした場合、復号化プログラムは、暗号文のファイルから 2 バイト単位でデータを処理するように工夫し、それを暗号文 C として処理し、復号化を実行すればよい。

9. C 言語での fgetc や fputc の関数では、データを 1 バイト単位でしか処理することができない。そこで、前記のように 2 バイト以上で表現されるデータをファイルの入出力で扱うアイデアとして B 進展開するという方法が考えられる。B 進展開については数学的準備の 3.3 節を参照。

   例えば、先の例 6.2 において、文字 “C” に対応する整数値 67 は、暗号化で $e_K(67) = 288$ となる。この整数値 288 を表すには、2 バイトが必要であるが、これを次のように分解し、1 バイト単位で処理できるようにする。すなわち、$B = 256$ とする。$288 = c_1 \times (256)^1 + c_0 \times (256)^0$ を満たす $(c_1, c_0) = (1, 32)$ の各 c_i は、 255 以下の整数値である。したがって、それらは 1 バイトで表現でき、288 に対応するデータとしては、1 と 32 を暗号文のファイルに保存すればよい。復号する場合は、暗号文のファイルから 1 と 32 を得てから、逆の計算をし、288 を求めた後に、$d_K(288) = 67$ を計算すればよい。

8.2 参考プログラム

本節では、課題プログラムを作成するに当たり参考になると思われるプログラムの実行例を示す。

1. RSA 暗号の暗号化（符号化）と復号化プログラムの実行例

   前節のプログラミングの準備で挙げた各項目に従って作成したプログラムの実行例を以下に示す。学生は、自身が作成したプログラムをコンパイルすることで得られる実行ファイルで実行すること。

   RSA 暗号の参考プログラム rsa.out を用いて、公開鍵 $(9929, 32399)$ と秘密鍵 $(9329, 32399)$ のペアでの実行例を示す。平文データのファイル名: 256byte.dat, 暗号化データのファイル名: encoded256byte.dat, 復元したデータのファイル名: decoded256byte.dat とする。

```
$
$ hostname
sol.cc.uec.ac.jp
$ ls -l | grep 256    <--- ls コマンドで，ひら文のファイル 256byte.dat のサイズを確認
-rw-r--r-- 1 ka103019 faculty   256 11 月 10  2017 256byte.dat
$
$ ./rsa.out -e 9929 32399 256byte.dat encoded256byte.dat          <--- 暗号化の実行 (1)
$ ./rsa.out -d 9329 32399 encoded256byte.dat decoded256byte.dat  <--- 復号化の実行 (2)
$ diff -s 256byte.dat decoded256byte.dat           <--- diff コマンドでデータを比較 (3)
ファイル 256byte.dat と decoded256byte.dat は同一です
```

```
$ ls -l | grep 256                  <--- ls コマンド で 3 種類のファイルサイズの比較 (4)
-rw-r--r-- 1 ka103019 faculty   256 11 月 10  2017 256byte.dat
-rw-r--r-- 1 ka103019 faculty   256  7 月 25 17:59 decoded256byte.dat
-rw-r--r-- 1 ka103019 faculty   512  7 月 25 17:59 encoded256byte.dat
$
```

2. 実行処理時間の計り方

   (a) 最初に、2 進展開法を採用した RSA 暗号プログラム rsabin.c (実行ファイル名: rsabin.out), と 2 進展開法を採用していない RSA 暗号プログラム rsa.c (実行ファイル名: rsa.out) の実行ファイルを用意する．学生は、自身が作成したプログラムをコンパイルすることで得られる実行ファイルを用意する．

   (b) 次に，10 通りの $e = 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192$ について，公開鍵 $(e, 32399)$ を設定した場合の実行時間を計測する．ただし，これらの e の値は実行時間を測るためのもので、実際に有効な鍵の値ではないことに注意する。

   (c) 最後に、ファイルサイズが約 2M bytes(2 メガバイト) のデータファイル 2Mbyte.dat を用意する。データファイル 2Mbyte.dat は、他の参考プログラムと同じ場所からコピーできる。以下に、time コマンドにオプション -p を指定した実行例を示す。

```
$
$ hostname
red99
$ ls -la | grep rsa
-rwxr-xr-x 1 ka103019 staffs   10044 Aug 25 15:23 rsa.out
-rwxr-xr-x 1 ka103019 staffs   10175 Aug 25 15:23 rsabin.out
$ ls -la | grep 2Mbyte.dat
-rw-r--r-- 1 ka103019 staffs 2097152 Aug 25 15:23 2Mbyte.dat
$ /usr/bin/time -p ./rsa.out -e 16 32399 2Mbyte.dat encodedMbyte.dat
real 0.46
user 0.40
sys 0.00
$
（中省略）
$ /usr/bin/time -p ./rsa.out -e 8192 32399 2Mbyte.dat encodedMbyte.dat
real 198.46
user 198.37
sys 0.00
$ /usr/bin/time -p ./rsabin.out -e 16 32399 2Mbyte.dat encodedMbyte.dat
real 0.22
user 0.16
sys 0.00
$
（中省略）
$ /usr/bin/time -p ./rsabin.out -e 8192 32399 2Mbyte.dat encodedMbyte.dat
real 0.44
user 0.39
sys 0.00
$
```

man コマンド で time コマンドを調べ、プログラムの実行処理時間としてユーザ CPU 時間 (user) のデータを採用することにする。上記の結果を表 1 に示す。妥当な値を得るためには、複数回のデータを取り、その平均値を求める必要がある．

表 1: 実行処理時間 (秒)(ファイルサイズ 約 2M bytes)

鍵の値 e	16	32	64	128	256	512	1024	2048	4096	8192
rsa.out	0.40	0.80	1.57	3.13	6.22	12.40	24.79	49.53	99.42	198.37
rsabin.out	0.16	0.18	0.21	0.23	0.26	0.28	0.31	0.35	0.36	0.39

参考文献

[1] Douglas R. Stinson, 暗号理論の基礎，櫻井幸一監訳、共立出版，1996.

[2] 池野信一, 小山謙二, 現代暗号理論（電子情報通信学会)、コロナ社，1986.

[3] 松坂和夫、代数系入門（第 28 刷発行)、岩波書店、2003.

[4] 石田信、代数系入門（第 12 刷発行)、実教出版、1990.

[5] R.L.Rivest, A. Shamir and L. Adleman, “A Method for Obtaining Digital Signatures and Publick-Key Cryptosystems,” *Communications of the ACM*, 21(1978), pp.120–126, 1978.

[6] 椋田 (むくだ) 實, はじめての C (改訂第三版), 技術評論社, 1995(平成 7 年).

[7] 平林雅英，ANSI C 言語辞典 (初版 第 10 刷発行), 技術評論社, 2000(平成 12 年).

[7]

[7]dvipdfmx -S -P 0x0004 rsatext2020.dvi(2022/8/14/14:30)