

# 暗 号

## — 小学校専門科目「数学」での実践 —

ば ば よし とも  
馬 場 良 始

(大阪教育大学 数学教育講座)

(2015年3月31日受付)

**概要：**2014年度の小学校専門科目「数学」で、現代社会の基盤を支える最重要事項の1つである「RSA暗号」が、実は17世紀のフランスの数学者フェルマーの整数に関する実にシンプルな「フェルマーの小定理」を用いて作られていることを説明し、その仕組みを簡単な問題を解くことにより体験してもらった。(このテーマの解説は、オープンキャンパスの模擬授業や、数学専攻・数学専門科目の授業でも行っている。)この論文では、その1時間半の講義の内容を詳述し、受講生の反応も紹介する。さらに、講義の内容から一步踏み込み、暗号にまつわる話題、RSA暗号の実際の使われ方、RSA暗号の数学的背景、RSA暗号以外の公開鍵暗号も解説する。

**検索語：**初等整数論、フェルマーの小定理、オイラーの定理、RSA暗号

### I. 始めに

暗号は現代社会の基盤を支える最も重要な技術の1つであると断言できる。例えば、インターネット・ショッピングでは、クレジット・カードの情報(カード番号、氏名、有効期限、カードの裏面に記載されている3桁のセキュリティ・コード)を入力することにより、簡単に買い物ができる。また、さまざまなICカード(学生証までICカード化された)で、さまざまなサービスが行われている。企業間の商取引も、電子的なものなしには考えられない。実は、こうしたことが安全に行われている背景には、高度な暗号化技術が貢献している。このような暗号について、考えて見よう。

まず、暗号の基本的な用語を定義しておく。

- 平文(ひらぶん)：元の文章。
- 暗号化：平文を暗号文に変換すること。
- 復号：正当な受信者が、暗号文を元の平文に戻すこと。
- 解読(または、(暗号を)破る)：暗号文を傍受した者が、暗号文を元の平文に戻すこと。
- 鍵：暗号化する規則や、復号するときの規則のこと。(例えば、この後に紹介するシーザー暗号では、「3文字ずらす」が鍵。)

### II. 暗号の歴史

まずは暗号の歴史を簡単に振り返ってみる。(講義で話したことより、少し詳しく述べてみよう。)暗号の起源は古く、数千年の歴史を持つ。戦時下においては、軍事技術の一つとして発達してきた。

紀元前1世紀に登場したシーザー暗号は、ユリウス・カエサル(英語読みで、ジュリアス・シーザー)が用いたとされ、暗号の歴史の中でもとりわけ有名なものである。これは元のアルファベットから文字をある数だけずらして作成する暗号方式で、例えば3文字後ろにずらすとき

平文：book → 暗号文：errn

となる。

単純に文字をずらすシーザー暗号は改良され、文字の置き換え表を作り、それに従って暗号化・復号をする換字式暗号が作られた。しかし、ある文字を単に別の文字に置き換えるような換字式暗号は、頻度分析により簡単に解読される。たとえば英語だと

- e や t といった文字の使用頻度が高い
- the という単語がよく使われる、
- tion のような綴りのパターンが多用される

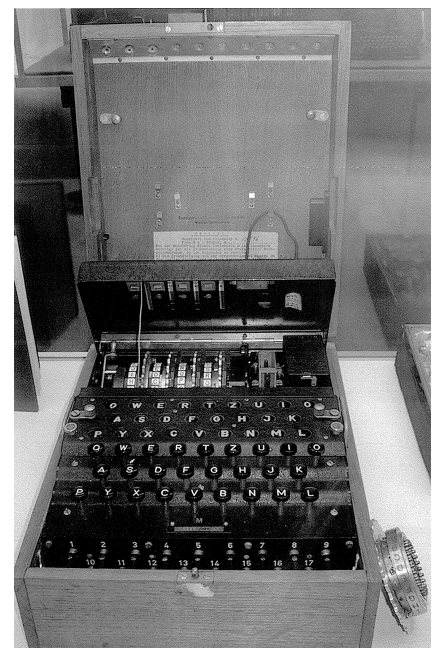
などの、元の言語が持っている特徴が暗号文の中に残ってしまうので、それを手がかりにして解読されてしまう<sup>\*1</sup>。

頻度分析法が発達し、単なる換字式暗号の安全性が低下したので、16世紀頃、言語の特徴を暗号文に(ほとんど)出さない新たな換字式暗号であるヴィジュネル暗号が考案された(詳細は、例えば[27]参照)。この暗号は非常に安全性が高く「解読不能の暗号」とも呼ばれたが、1863年に暗号文から言語の特徴を見いだす方法がを見つけ出され破られてしまった。この間、18世紀には、ヨーロッパの列強各国が、他国の通信文の入手・解読を行う「ブラック・チェンバー」と呼ばれる国家秘密情報機関を作るようになったり、19世紀中頃には、電信が実用化されて、通信文は傍受・解読されるものとして、暗号を用いるのが常識となったりした結果、安全な暗号の必要性はさらに高まった。

さらに19世紀末には無線が発明され、電線を引くことなく地球上のどこでも通信が可能(傍受も可能)になった。ほぼ同時期に第1次世界大戦が勃発し、安心して無線で通信できる強力な暗号の開発が急務となったが、生み出された暗号は、短期間の間に解読されるか、実用性がないかのいずれかであった。

第1次世界大戦後、容易に破られない暗号解読技術を求めて、さらに複雑な手順の暗号が次々に考案された。そして、そのような複雑な手順を簡単な操作で間違いなく行うために、機械式の暗号機が作られるようになった。このようなものの中で、歴史上特に有名なものに、第2次世界大戦中にドイツ軍が使用していたエニグマ暗号機がある。解読されない強い暗号を求めて、さまざまな暗号システムが考え出された中で、換字のしくみをもっと複雑にし、さらに1文字ごとに暗号化手順を変えろという、当時おそらく最強の暗号システムであり、しかも鍵は毎日変更された。暗号機は持ち運び可能なタイプライター型で、1兆を越す暗号鍵を簡単に設定・選択でき、暗号文の復号も同じ暗号機でできた。

第1次世界大戦の結果、再び独立国となったポーランドは、国家維持のために隣国であるドイツ、ロシアの通信暗号を解読し続ける必要に迫られていた。暗号解読は、国家の存続を賭けた最重要課題であったのだ。そして、1930年代に、ドイツ軍が絶対の自信を持っていたエニグマは、ポーランド暗号局に所属していた数学者マリアン・レイエフスキによった破られるこ



エニグマ暗号機  
([22] から引用)

<sup>\*1</sup>このような解読手法は、エジプトのヒエログリフ、バビロニアの楔形文字、古代トルコのルーネ文字、インドのブラーフミー文字などの古代文字解読にも使われた。

とになる<sup>\*2</sup>。発見された解読法によるエニグマ暗号の解読は順調に続けられた。しかし1938年、エニグマ暗号機はより強力なものに改良され、突然従来の手法では解読できなくなった。しかも、1939年の4月に、ドイツはポーランドとの不可侵条約を破棄を発表。また、暗号局が改良されたエニグマ暗号機の解読研究を進めていくにつれ、解読には膨大な予算が必要になることが明らかになってきた。このため、ポーランドはそれまで得られていた解読情報を、イギリス・フランスに提供することを決定し、1939年7月にエニグマのレプリカと解読機ポンプ作成法の書類が両国に手渡された<sup>\*3</sup>。そして、その直後の1939年9月、ドイツはポーランド侵入し、第2次世界大戦が勃発することになる。

ポーランドの解読情報を得たイギリスもまた、ドイツの潜水艦Uボートによるイギリス商船の無差別攻撃を受け、イギリス本国への食糧補給が困難になる問題を抱えていた。そのため、ポーランドに倣い、暗号解読チームに大量の数学者を雇い入れ、バッキンガムシャー州のブレッチレー・パークに巨大暗号解読拠点を作り上げ(スタッフの総数は、最盛期には7千人に達した)、改良型エニグマの解読機ポンプ(電気機械式。ポーランドの解読機に敬意を払い同じ名前が付けられた)や、その後ドイツ軍が使い始めたローレンツ暗号の解読機コロサス(電子式デジタルコンピュータ)を完成をさせた。しかし、解読機が完成し、エニグマ暗号は即時に解読されている事実は秘密にされ、ドイツ軍は終戦までエニグマを使い続けた。エニグマの解読成功は、ドイツの敗北を2年は早めたと言われ、もし解読の成功がなければ、ドイツにも原爆が投下された可能性は十分あったと言われている。

注1. チューリングを代表とするイギリス数学者チームによるポンプ作成は、あくまで史実に基づいたフィクションであるものの、この3月に公開された映画「イミテーション・ゲーム/エニグマと天才数学者の秘密」で描かれている。

第2次世界大戦中の、チューリングを代表とする数学者たちの暗号研究の結果、英米は優れた暗号技術を獲得した。そして、戦後もそれを最重要秘密事項とし続けた<sup>\*4</sup>。実は、エニグマ暗号機解読の事実は、1974年に初めて情報公開されたのである。現代では、第2次世界大戦の連合国側にとっての最大の英雄は、実は「数学者たち」であったと言われている<sup>\*5</sup>。

一方、第2次世界大戦時の日本の暗号の一部は、開戦前からアメリカによって破られていたことが知られている。この時期の日本の暗号は、大別すると3種類(政府が用いる外交暗号、海軍暗号、陸軍暗号)存在したが、外交暗号は、定期的に新しいものに更新されていたにもかかわらず、1935年のロンドン第2次軍縮会議での日本政府と会議に出席する日本代表団との通信が、すべてアメリカ政府に解読されていた。その後、開戦直前の半年間、開戦を避けるために日米の外交交渉が行われたが、その折りの通信もほとんどアメリカに漏れていた<sup>\*6</sup><sup>\*7</sup>。このような状況下で、アメリカ国務

<sup>\*2</sup>ポーランド暗号局は、これまで暗号解読に適任とされてきた言語構造の専門家ではなく、数学者が適任ではないかと考え、選考の結果3人の数学者にエニグマの解読を託した。3人の内の1人であるレイエフスキは、従来通りの統計学的手法に加え、置換群の性質を用いて解読に成功したのである。

<sup>\*3</sup>ポーランド政府のこの申し出を、イギリス・フランス両政府は驚きをもって受け入れたという記録が残っている。

<sup>\*4</sup>英米にとって暗号技術は、国家を安定的に運営していくための最重要基盤であった。その一端は、後に述べる3つの事実。つまり、商用暗号の標準システムDESの選定時における、アメリカ国家安全保障局(NSA)の介入、PGPを開発しこれをフリー・ソフトウェアとして配布したジーマンとアメリカ政府との1990年代の法廷論争、実は公開鍵暗号やRSA暗号をいち早く開発していたが、その事実は秘匿事項として公表しなかったイギリスの英国政府通信本部の行動、からも伺い知れるが、戦後のエニグマ暗号機について、次のような事実もあった。

戦後、イギリスは何千台ものエニグマ暗号機を入手し、それらをもって植民地であった地域に配給した。旧植民地の人々は、1974年までエニグマ暗号機で作られる暗号は安全であると信じ込んでいた。そして、イギリスは長年にわたり秘密通信を解読していた。

<sup>\*5</sup>終戦後、暗号解読に関わった人たちは、大戦中の国への貢献を明かすことは許されなかった。そのため、ある若い暗号解読者は、母校の校長から「前線に出なかったおまえは母校の恥さらしだ」という手紙を受け取ったという。

<sup>\*6</sup>日本の外交暗号を解読する「パープル暗号機」と呼ばれるものさえ作成されていた。

<sup>\*7</sup>あるアメリカの報告は次のように述べている。敵の暗号を解読できていることは、軍幹部および政府上層部が決定を下す際の心理状態を変えた。敵の内情を知っているということは、非常に気持ちを楽にする。時間が経つにつれてその感覚がいつのまにか大きくなることは、関係者の思考、やり方、癖、行動を、定期的にじっくり観察していれば分かるものだ。この種の情報が得られていることで、われわれはより長期的な計画を、より確信をもって立てることができるし、その際にも苦悩は少なく、より前向きになれるのである。

長官コーデル・ハルによる“The Ten Pointed-Note”（俗称ハル・ノート）が出され、日本は一気に開戦に向かうことになる<sup>\*8</sup>。

そして、真珠湾攻撃<sup>\*9</sup>に関係する海軍暗号も、開戦前から部分的に破られていたことは知られており、真珠湾攻撃が行われることを、アメリカ政府は事前に知っていてその情報を握りつぶしたのか（このような考え方は、開戦直後からアメリカ保守派の一部でささやかれていた）、それとも本当に知らなかったのかは現在も論争が続いている。

なお、陸軍暗号は終戦まで破られなかった<sup>\*10</sup>。

**注 2.** 第2次世界大戦中の太平洋戦線で、アメリカ軍はアメリカ先住少数民族であるナヴァホ族の言葉を暗号として使用した。ナヴォホ族の兵士が、通信文を瞬時にナヴォホ語に翻訳してそのまま無線で送信する（ナヴァホ語は軟口蓋音や、鼻音や、舌のもつれるような音が続く奇妙な言葉で、ナヴァホ語を解しない者が通信文を無線で傍受しても、それを書き取ることさえ難しい）。そして、受信した通信文を、ナヴォホ族の兵士が聞き取り米語に翻訳して内容を伝える。複雑な暗号機によって時間を掛けて暗号化・復号するのではなく、瞬時に暗号化・復号が行われ、しかも信頼性の高い暗号であった。ナヴォホ兵がいなければ、硫黄島を占領でき

<sup>\*8</sup>第1次世界大戦へのアメリカ参戦に関して、戦後、上院議員のジェラルド・ナイ（Gerald Nye）を委員長とするアメリカ合衆国上院のナイ委員会（Nye Committee, 1934年4月設置）は、軍需産業は第1次大戦を通じてアメリカの外交政策に強い影響力を保持し、兵器産業に投資し、英仏に戦時融資していた銀行家が、開戦時の大統領であるウッドロウ・ウィルソン大統領（民主党）に対して、戦争に介入するよう圧力をかけていたことを明らかにした。これが、第2次世界大戦の初期段階におけるアメリカの中立政策（孤立主義）の要因となり、実際、アメリカが第2次世界大戦に参加する時点でのアメリカの大統領ローズベルトは、史上初の3選目の選挙において、既に始まっていたヨーロッパでの第2次世界大戦には参加しないことを公約にして当選している。（日本が満州国を建国し、中国大陆での紛争は拡大していったが、当時のアメリカの世論は、アメリカがアジアの紛争に巻き込まれることには反対していた。）このため、日独からの先制攻撃なしに第2次世界大戦に参戦することは事実上不可能な状況にあった。

ちなみに、ローズベルト大統領（民主党）は、世界大恐慌の直後の不況下の1933年に大統領に就任した。「ニューディール政策」と呼ばれる経済政策で有名であるが、アメリカ経済がこの政策で立ち直ったとは言いがたい。しかし、第2次世界大戦への参戦による、史上最大の軍拡・軍需経済・戦時経済の著しい増大によって、アメリカ経済は完全に回復した。

また、ソ連崩壊後の1995年以降にアメリカ政府により順次公開されていった「ヴェノナ文書」によると、ローズベルト政権・アメリカ政府・マスコミ・出版業界・平和を謳う各種団体等にはソ連・コミンテルンのスパイが300人以上潜入しており、日本が北進してソ連と開戦するのではなく、アメリカと開戦するように誘導していたことが判明している。ソ連に北方領土を明け渡すことを密約したヤルタ会談（戦後、朝鮮半島が分断され、東ヨーロッパがソ連の勢力下に置かれたのもヤルタ会談が原因である）で大統領側近として参加していたアルジャー・ヒス、在米日本資産の凍結を主導し日本の金融資産を無価値にして破産に追い込み、ハル・ノートの原案作成にも関与したハリー・デクスター・ホワイ特等、政権上層部にも潜り込んでいた。（ソ連のスパイは日本の上層部にも入り込んでいたことが知られている。近衛内閣のブレーンの朝日新聞記者尾崎秀実らコミンテルンのスパイ・グループが1941年のゾルゲ事件で逮捕されている。）

開戦前、「強い日本がアジアに混乱をもたらしている」という考えの元に対日経済制裁を仕掛けたローズベルト政権を、「強い日本がアジアに安定をもたらす」と共和党は主張し批判していた。（親日なのではなく、アジアの紛争に巻き込まれたくないという考えが根本にある。）そして終戦後、日本の敗戦・弱体化により、中国に共産党政権が誕生・アジアに（東ヨーロッパにも）共産主義勢力が拡大し、アメリカはさらにアジアでの紛争に巻き込まれ、アメリカ人の血がアジアに流れた。このため、第2次世界大戦の真の勝者はソ連だけと言われることもある。

<sup>\*9</sup>日本軍の真珠湾攻撃は、戦闘機の攻撃で戦艦が撃沈できることを世界で初めて明らかにしたと言われている。アメリカ側の損害は、海軍の戦死、行方不明、負傷後の死者を合わせて2008名、負傷710名、海兵隊の戦死109名、負傷69名、陸軍戦死218名、負傷364名、陸海軍基地の技術者戦死68名、負傷69名であり、主力戦艦8隻を含む19隻が被害を受け、主力艦は壊滅状態であった。（2隻の主力空母は、湾外に居たため難を逃れている。）ローズベルト大統領は、日本の第1攻撃が、このように大きなダメージを与えることが可能であるとは露にも思っていなかったようで、海軍長官は12月15日の記者会見で、「沈没したのは『アリゾナ』と駆逐艦3隻、補助艦1隻だけである」と事実と異なった公表を行っている。

真珠湾攻撃前、大多数のアメリカ国民は、ヨーロッパ・中国では戦争が行われていたが、それらは遠い世界のことで、戦争に参加しないことを公約として選ばれた大統領がおり、日本に対する金融資産・資源の追い込みやハル・ノートのことなどは知らずに、平和に、そして日本に対して特別な悪感情をもたずに暮らしていたと言われている。しかし突如舞い込んだ真珠湾攻撃のニュースは、それまで8割近い米国民が参戦反対であったものを一気に変化させ、「リメンバー パールハーバー」のスローガンの下で全国的な参戦賛成が実現した。これは、戦前にローズベルト大統領が語っていた「私は分裂した国民を戦争に引き込むことはできない。私はこれを第1次世界大戦から学んだ。しかし、ウィルソン大統領は私に教えた。私は、合衆国が公然と参戦する場合は、国民が一致して参戦することを、確実に、非常に確実にしようと思っている」が実現されたのである。ちなみに、ローズベルト大統領は真珠湾攻撃の4ヶ月以上前の7月23日に、蒋介石政権と連携して日本本土を約500機の戦闘機や爆撃機で空爆する計画（JB355と呼ばれる）に承認のサインをしている。

<sup>\*10</sup>連合国の暗号解読努力が、外交暗号・海軍暗号に主に向けられていたためだと言われている。



なかっただろうとも言われている。ナヴォホ暗号が機密扱いを解かれたのは 1968 年のことであった。(ナヴォホ暗号の詳細は、例えば [27, 第 V 章] 参照)

第 2 次世界大戦が終わると、機械式暗号はすたれ、コンピュータを用いた暗号が主流になっていった。(コンピュータでは 2 進数が使われる。よって、平文は適当なコード表により数字列に変換され、その数字列が暗号化される。暗号の鍵も数字列である。)

注 3. 注 1 に述べた映画で、「チューリングマシンはコンピュータの原型とされている」という説明があった。実は、現在普通に使われているコンピュータを特徴付ける、プログラム可変計算機構の数学的意味を最初に論じたのが、1936 年のチューリングの論文 “On Computable Numbers, with an Application to the Entscheidungsproblem”, Proc. London Math. Soc. (1936), 230–265<sup>\*11</sup> であった。この論文でチューリングは、数が計算可能であるとは、その小数表現を機械で書き出せることと定義し、(単機能) チューリングマシンと呼ばれることになる、自分自身でプログラムを書き換え、それをまた実行する想像上の計算機 (プログラムであるとも言える) を考案した。そして、すべての単機能チューリングマシンの計算を模倣できるマシンとして、万能チューリングマシンと呼ばれることになるマシンのアイディアも与えている。万能チューリングマシンは、計算可能な (これは「単機能チューリングマシンで表される」と同意) すべての計算を行うことが可能であり、現在我々が使っているノイマン型とよばれるコンピュータはこれを現実化したものになっている<sup>\*12</sup>。

ノイマン型とはプログラム可変内蔵方式、つまりプログラムは、それが実行されるときに内部のメモリ中に記憶され、計算と同じ速さで取り出すことができ、しかも、そのプログラムの実行中に得られる実行結果によって、そのプログラム自身や実行過程を変更できるものを意味する。そして、ノイマン型は万能チューリングマシンと同等の計算能力をもっている。ノイマン型と呼ばれている理由は、フォン・ノイマン (アメリカの数学者。第 2 次世界大戦中は原爆開発に関わった) が、ペンシルバニア大学で開発されていた新型コンピュータ EDVAC (Electronic Discrete Variable Calculator) の開発にコンサルタントとして途中から加わり (ロスアラモスでの原爆開発の合間を縫って EDVAC の技術会議に出席していた)、その会議から得られた結論を自分流に総括した「草稿」が、1945 年にノイマンの単著として流布しまったことによる。ノイマンの、EDVAC 開発への貢献度は諸説ある。ただ、EDVAC の完成は遅れ、史上初のノイマン型実用的コンピュータの第 1 号の座は、1949 年にケンブリッジ大学のウィルクス<sup>\*13</sup> によって制作された EDSAC (Electronic Delay Storage Automatic Calculator) に譲ることになる<sup>\*14</sup>。チューリングの論文が出てから 13 年後のことであった。

<sup>\*11</sup> 「計算可能数とその決定問題 (Entscheidungsproblem) への応用」と題するこの論文の主目的は、万能チューリングマシンを考案することではなかった。「記号列が表す主張がヒルベルト公理系から証明可能であるか否かを決定する機械的な手順 (後にアルゴリズムと呼ばれることになる) は存在するか?」という問題に、一般的な形では解決不能という結論を与えることが主目的であった。その証明を行うために、チューリングマシンの概念が使われている。

なお、投稿直前にアロンゾ・チャーチ (アメリカの数学者) が同じ結論の論文を発表するが、チューリングの手法が革新的であり、しかもかなりチャーチのものと違うため、指導教官のニューマンは投稿を勧めた。論文は、チューリングの「計算可能性」の概念と、チャーチの「実効的計算可能性」が同値であることの証明が付け加えられて投稿された。(チューリングはこの後すぐの 1936 年 9 月から 1938 年 7 月にかけて、チャーチの居るプリンストン大学に博士留学し、指導を受けることになる。この時期、プリンストンにノイマンが在籍しており、チューリングを高く評価し、助手として残って欲しいと依頼している。しかし、チューリングはこの要請を断り、戦争直前のイギリスに帰った。)

<sup>\*12</sup> この論文が出た頃、コンピュータとはある特定の目的 (例えば、微分方程式を解くだけ。できることはそれだけ) のために設計される機械と思われていた。

<sup>\*13</sup> ウィルクスはノイマン型のコンピュータについて、1946 年の夏にペンシルバニア大学で行われた「電子式デジタル計算機設計の理論と技術」と題する講習会で、EDVAC の開発スタッフたちから直接学んだ。

<sup>\*14</sup> 小規模実験機としては、EDSAC の 1 年前の 1948 年の 6 月に、マンチェスター大学でノイマン型コンピュータ Baby Mark I が稼働している。これは、マックス・ニューマン (ケンブリッジ大学でチューリングの指導教官、ブレッチレー・パークでコロッセス制作メンバー) が始めたプロジェクトであり、マンチェスター大学に移ったチューリングも、1948 年 9 月から Mark I の商用化プロジェクトに参加し、プログラム面の責任を負っている。

当初のコンピュータは真空管を何万個も使う大型のものであり、コンピュータとそれを使った暗号は政府と軍部だけが利用できるものであった。しかし、1959年に集積回路ICが発明され、1960年代コンピュータはますます性能が上がり、価格は下落した。そして、民間企業もコンピュータを購入し、重要な通信を暗号化するために、コンピュータを利用し始めた。しかし、暗号文で通信を行うためには、送信元と送信先が同じ暗号化システムを使用する必要がある。このため、1977年米国商務省標準局(NBS)は、IBMから提案された暗号方式をDES(Data Encryption Standard)(デス)と名付け、これを商用暗号の標準システムとして公布した<sup>\*15</sup>。これは、転置と換字を適当に組み合わせた処理を16段も繰り返すもので、短時間による解読は不可能に近いとされ、長期間にわたり信頼できる暗号として使用されてきた<sup>\*16</sup>。しかし1994年、三菱電機の松井充(数学科出身)は、デスクトップワークステーション上で、50日処理して鍵を発見することができる、線形解読法と呼ばれる新しいテクニックを発表し、世界の暗号研究者を驚かせることになった<sup>\*17</sup>。そしてRSA Data Security社が行ったDES暗号解読コンテスト(1999年)では、distributed.netのグループが僅か22時間15分で解読している。このような経緯から、2000年にはDESに代わる米国政府標準の次世代暗号技術としてAES(Advanced Encryption Standard)の選定が行われ、その結果Rijindael(ラインダール)が選ばれ、2001年に公布された。この暗号はガロア体理論に基づいており、このため、いくつかの観点で安全性を数学的に証明することが可能である。この暗号は、現在でもいろいろな製品に組み込まれ、広く使われている。

実は、これまで紹介した暗号はすべて、現在「共通鍵暗号」とか「秘密鍵暗号」と呼ばれているものである。ここで、この用語を定義しておこう。

- 共通鍵暗号(秘密鍵暗号): 暗号化に使う「鍵」と復号に使う「鍵」が同じ暗号のこと。「鍵」が同じであるので、暗号の作り方が分かれば、復号の方法も当然分かる。よって、「鍵」は他人に知られないよう秘密にしなければならない。このことから「秘密鍵暗号」とも呼ばれる。1976年以前の暗号はすべてこの「共通鍵暗号」であった。

共通鍵暗号は、共通鍵を送信相手に予め極秘に連絡しておかなければならないので、その方法が問題になる。また、その費用が必要となる。そして、鍵が相手方から漏洩してしまうかもしれないという危険性もある。これらの問題の解決策として、公開鍵暗号(次の§IIIで解説する)が考案された。

注4. 逆に、共通鍵暗号の利点とは、鍵の長さが公開鍵暗号より短いため処理速度が高速であることである。(鍵の長さは、DESは56ビット、改良型のトリプルDESは56, 112, 168ビットのいずれか、AESは128, 192, 256ビットのいずれか。公開鍵暗号の場合、後で述べるように、安全性を保証するためには最低でも1024ビットは必要とされる。)

例えば、ソニーが開発した非接触型ICカードの技術方式FeliCaはDESやAESを用いており、特に高速処理が求められる自動改札機(ICOCA, Suica, PiTaPa)、いろいろな企業のビル入館のゲート、レジのアプリケーション(電子マネーEdy, nanako, WAON)に用いられている。本年度からIC化された大阪教育大学の学生証もFeliCaである。その開発秘話は[31]に詳しく書かれている。また、家庭用の無線LANでもAES等の共通鍵暗号が使われている。

<sup>\*15</sup>DESはIBMが提案した暗号ルシファーの改良版である。ただし、ルシファーの鍵の長さは128ビット、DESの鍵の長さは56ビット、と暗号の強度がわざと弱められている。この過程に、アメリカ国家安全保障局(NSA)(1952年に設立されたアメリカ最大の機密機関。長らく、その存在は公には否定されていた。世界中のどの組織よりも多くの数学者を雇っているとか、世界最大のコンピュータ購入機関であるとか、世界中の誰よりも多く電話の会話・通信を盗聴・傍受しているとか言われたり噂されたりしている。2013年に、局員であったエドワード・スノーデンが、個人情報収集の実態と手口を告発したことは記憶に新しい)が関わっていることは知られており、なぜ暗号の強度が弱められたのか? 公布当初から、さまざまな憶測・噂が飛び交った。

<sup>\*16</sup>鍵長が56ビットと短いこともあり、NSAにしか存在しないと言われる超高速コンピュータでは短時間で解読できていたのではないかと噂されている。

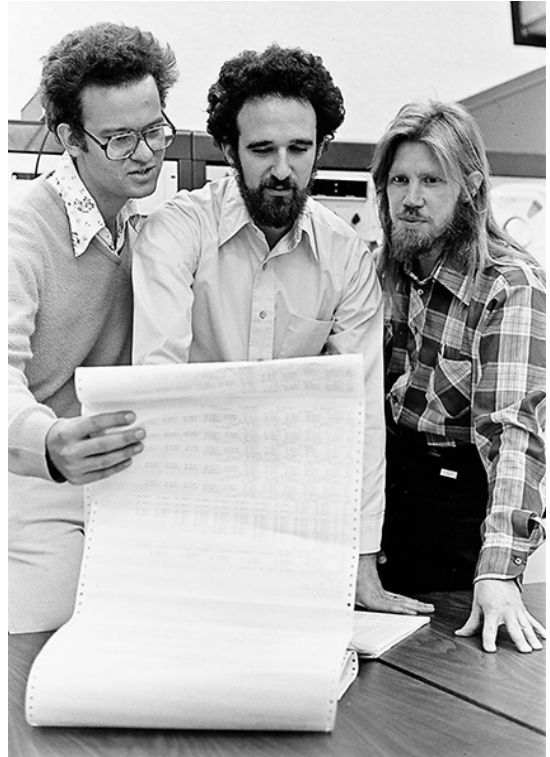
<sup>\*17</sup>三菱電機は、DES解読の技術を生かし、この後さまざまな信頼性の高い暗号システムを生み出していくことになる。

注 5. 共通鍵暗号の問題点の 1 つ「どうやって通信相手に密かに共通鍵を知らせるのか?」という問題は **鍵配送問題**と呼ばれ、歴史上のどの暗号でも悩みの種とされてきた。例えばエニグマ暗号では、毎月 1 回、毎日変更される鍵が書かれた本を、極秘にすべてのエニグマ暗号機の元に配達する必要があった。戦後の 1976 年に DES がアメリカでの商用暗号の標準システムと定められ、多くの企業が DES を使うようになり、しかもその頻度はどんどん増えていった。アメリカ政府は鍵の管理・配送を行う組織を作って、日々何トンもの鍵を配送していた。また、銀行は特別な配達人を使って鍵を配送していた。それらの費用は法外な額に膨れ上がり、財政を圧迫した。

この大問題を最初に解決したのは、公開鍵暗号のアイデアを生み出したディフィー、ヘルマン (次の §III 参照) と、マークル<sup>\*18</sup>の 3 人である<sup>\*19</sup>。公開鍵暗号のアイデアが生まれたのとはほぼ同時期の 1976 年のことであった。(現在では単にディフィー・ヘルマン鍵共有と呼ばれることが多い。)

ではそのアイデアを、A さんと B さんの間で共通鍵を決めたいときで説明しよう。大きな素数  $p$  と、 $p$  より小さい自然数  $m$  を  $m + p\mathbb{Z}$  が乗法群  $(\mathbb{Z}/p\mathbb{Z})^\times$ <sup>\*20</sup> の生成元<sup>\*21</sup>であるように取る (単位元ではない元でもいいが、生成元である方が安全性が高い)。この  $p$ ,  $m$  は A さんと B さんの間で予め決めておく。(第三者に知られても構わない。) さて、共通鍵を決める必要が生じたとき、それぞれが勝手に  $p-2$  以下の自然数 1 つ選ぶ。今、A さんが  $a$  を選び、B さんが  $b$  を選んだとすると、A さんは  $m^a$  を  $p$  で割った余り  $r_a$  を計算し、B さんは  $m^b$  を  $p$  で割った余り  $r_b$  を計算する。2 人は電話等で  $r_a$ ,  $r_b$  の情報交換をする。そしてさらに、A さんは  $r_b^a$  を  $p$  で割った余りを計算し、B さんは  $r_a^b$  を  $p$  で割った余りを計算する。これらは共に  $m^{ab}$  を  $p$  で割った余りであるので一致する。この一致した数を共通鍵として利用する。第三者は、 $p$ ,  $m$ ,  $r_a$ ,  $r_b$  を知り得る可能性がある。しかし、これらを知っていても  $a$ ,  $b$  は計算できない。よって共通鍵は分からない。

鍵配送は、この後に発見された RSA 暗号等の公開鍵暗号によっても可能であり、現在では公開鍵暗号によって行われることも多い。(§IV の「ハイブリッド暗号方式」参照)。ディフィー・ヘルマン鍵共有は 2 人の参加者が同時に協調動作する必要があるため、用途が制限される。



マークル (左), ヘルマン (中央), ディフィー (右)  
(スタンフォード大学のサイトから引用)

<sup>\*18</sup>鍵配送問題を研究していたカリフォルニア大学バークレイ校の大学院生で、この問題の重要性を教授に理解してもらえず、この問題を研究するヘルマンの存在を知って 2 人のところに合流してきた。

<sup>\*19</sup>3 人による鍵配送問題の共同研究の末、ある夜、このアイデアが舞い降りたのはヘルマンにであった。翌朝、このアイデアは早速 2 人に伝えられた。ディフィーは語っている。「マーティ (ヘルマン) が説明してくれた鍵交換システムは、歯ざりするほど簡単だったよ。彼の説明を聞きながら、同じ考えが何度も頭に浮かび上がったことに思い当たった。しかし結局、表面まで浮かび上がっては来なかったんだ。」

<sup>\*20</sup> $F$  を体とすると、 $F$  からその零元  $0_F$  を除いた集合  $F - \{0_F\}$  は、体  $F$  の乗法に関して群をなす。これを  $F$  の乗法群とよび  $F^\times$  で表す。ここでは、素体  $\mathbb{Z}/p\mathbb{Z}$  の乗法群  $(\mathbb{Z}/p\mathbb{Z})^\times$  を考えている。

<sup>\*21</sup>説明を簡略化するために有限群  $G$  で説明すると、 $G$  の元  $a$  は、 $G = \{a, a^2, a^3, \dots, a^{n-1}, a^n = 1_G\}$  (ここで、 $1_G$  は群  $G$  の単位元) となるある自然数  $n$  が取れるとき、群  $G$  の生成元であると呼ぶ。乗法群  $(\mathbb{Z}/p\mathbb{Z})^\times$  の場合、例えば  $p = 11$  とすると、 $(\mathbb{Z}/11\mathbb{Z})^\times = \{a + 11\mathbb{Z} \mid a = 1, 2, \dots, 10\}$  であり、 $a + 11\mathbb{Z}$  は 11 で割ると  $a$  余るすべての整数の集合を表し (それを 1 つの元と考えている)、例えば  $(6 + 11\mathbb{Z})^2 = 6^2 + 11\mathbb{Z} = 3 + 11\mathbb{Z}$ ,  $(6 + 11\mathbb{Z})^3 = 6^3 + 11\mathbb{Z} = 7 + 11\mathbb{Z}$ , ... であるので、このような計算を続けていくと、 $2 + 11\mathbb{Z}$ ,  $6 + 11\mathbb{Z}$ ,  $7 + 11\mathbb{Z}$ ,  $8 + 11\mathbb{Z}$  が  $(\mathbb{Z}/11\mathbb{Z})^\times$  の生成元であり、その他の元はそうではないことが簡単に確かめられる。

### III. RSA 暗号

公開鍵暗号とは、1976年にディフィーとヘルマン<sup>\*22</sup>によって発表された、暗号の新時代を開く画期的な暗号のアイデアである。ただし、彼らの述べていることは単にアイデアだけであり、具体的にどのようにすれば、それが実現できるかは述べられていない。

そのアイデアのポイントは、鍵を2つ使うこと。ひとつは公開鍵 (public key) と呼び、公開鍵サーバーなどに登録して誰でもダウンロードできるようにしておく。もうひとつは秘密鍵 (private key) と呼び、誰にも知られないように大切に保管しておく。公開鍵と秘密鍵はペアの鍵で、お互いに関係はあるが、公開鍵をいくら調べても、秘密鍵がどうなっているか分からない。

暗号化して情報を送ってもらいたいとき、公開鍵で暗号化して送ってもらう。鍵は公開されているので、誰でも暗号化し送信できる。暗号化された文書は、公開鍵では復号できない。これを復号できるのは秘密鍵だけ。誰でも暗号化して送信することができるけれど、復号して読むことができるのは秘密鍵を持っている者だけ。

逆に秘密鍵で暗号化すると、これは公開鍵でないと復号できない。しかも秘密鍵は1個しかなく、それを持っているのは本人だけ。よって、秘密鍵で暗号化された文書は、秘密鍵を知っている本人でないと作れない。これは「署名」と同じ効果がある。つまり、紙とペン・印鑑が使えないコンピュータネットワーク環境で「電子署名」が可能になるのである。(この電子署名は、現在では、情報社会を支える最重要基礎技術になっている。)

公開鍵暗号の概念が発表されて以来、多くの暗号研究者を巻き込んで公開鍵暗号実現化の研究が行われた。そして翌年の1977年に、マサチューセッツ工科大学の、リベスト、シャミア、エイドルマン<sup>\*23 \*24</sup>によって、実質的に公開鍵暗号の第1号であるRSA暗号が生み出された<sup>\*25 \*26</sup>。

<sup>\*22</sup> 2人とも暗号研究者。ディフィーはMITで数学を、ヘルマンはニューヨーク大学とスタンフォード大学で電気工学を学んだ。フリーのセキュリティ専門家であったディフィーは鍵配送問題に取り組んでいたが、同じ問題に取り組んでいる研究者の存在を耳にする。(不思議なことに、当時この問題に取り組んでいた研究者はごく小数であった。)それがスタンフォード大学教授のヘルマンであった。ディフィーはすぐにニューヨークからカリフォルニアまで車を飛ばしヘルマンに会いに行き、意気投合する。共同研究のため、便宜上ディフィーはヘルマンの研究室の院生になる。公開鍵暗号のアイデアはこの共同研究から生まれた。実際のアイデアはディフィーによって生み出されたとされている。

<sup>\*23</sup> 3人とも大学で数学を学んでいる。そして、リベストとシャミアは大学卒業後に計算機科学の分野に進んだ。3人の研究は、整数論の研究者であったエイドルマンが、たまたまりベストの研究室を訪ねたとき、リベストがディフィーとヘルマンの論文を手に興奮した様子でその内容を語ったことから始まる。その後シャミアが加わり、リベストとシャミアが次々に考え出した42個のアイデアに対して、エイドルマンがそれらの不備を指摘し続けた。リベストが思いついた43番目のアイデアが、現在RSA暗号と呼ばれるものである。

<sup>\*24</sup> エイドルマンは当初、RSA暗号の論文の共著者になることを辞退している。しかし、最終的には自分の名前を最後に書くという条件で共著者になった。これが、RSAがアルファベット順のARSではない理由である。エイドルマンはこの後、広い分野で先進的な研究を行った。例えば、「コンピュータ・ウィルス」という言葉はエイドルマンが命名したものであり、いち早くこの分野の研究を行った。また、植物の葉緑素を使った分子計算、DNA計算なども行っている。

<sup>\*25</sup> 現在のセキュリティ・システムの多くはRSAに基づいている。この功績によって3人は2002年のチューリング賞(計算機科学分野で革新的な功績を残した人物に贈られる、ノーベル賞と同クラスとされる賞)を受賞した。

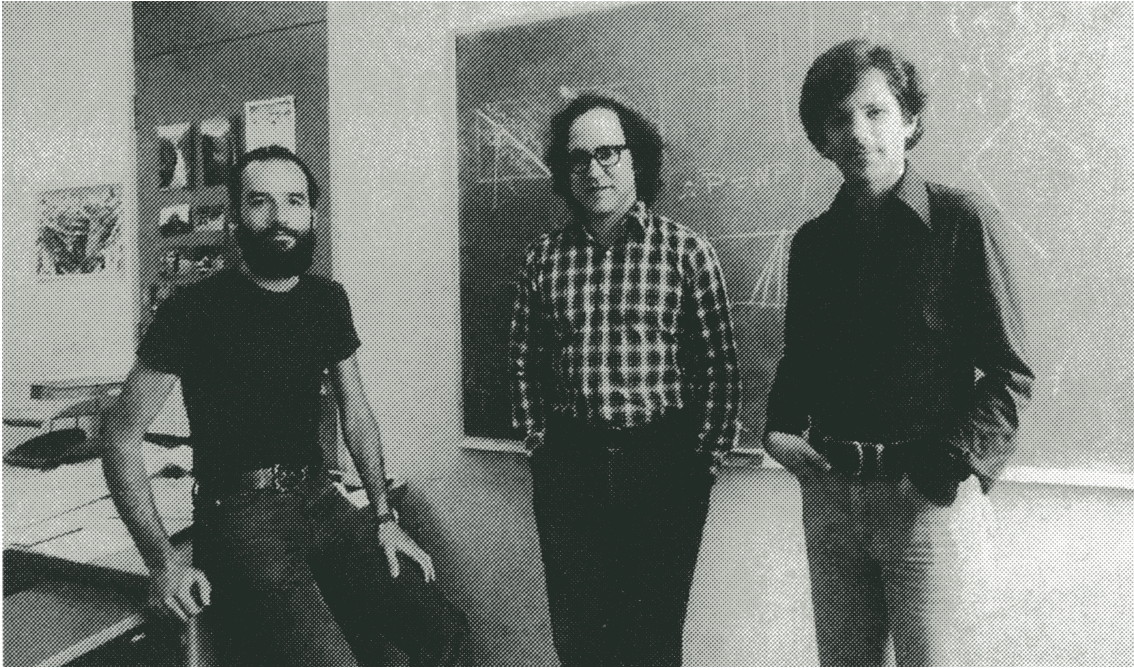
<sup>\*26</sup> 1997年以前、公開鍵暗号のアイデアの発見やRSA暗号の発見は、今述べた通りであると考えられてきた。しかし現在では、実はもう少し前に、別の研究機関の人々によって発見され、その事実は1997年まで秘密にされていたことが分かっている。その研究機関は、ブレッチレー・パークの後を継ぐ形で設立されたイギリスの最高機密機関、英国政府通信本部であった。

1969年頃、公開鍵暗号のアイデアは、スタッフの暗号作成家ジェイムズ・エリスにより発見され、それは上司に伝えられた。アイデアだけであった点もディフィー、ヘルマンと同じである。数学者でなかったエリスは、アイデアを実現する能力が自分がないことを悟り、アイデアのみを報告したのである。

アイデア実現化の問題は、1973年に加わった数論の研究者クリフォード・コックスに、新入りであるコックスの教育係に指名された研究員から軽い調子で伝えられた。そして、彼はRSA暗号と同じものを生み出すことになる。彼は述べている。「取りかかってから終わるまでに30分もかかりませんでした。やっていて楽しかったですね。『やあ、もらった問題が解けちゃったぞ』と思いました。」(コックスは、これが大問題であるという認識を持たず、軽い気持ちで取り組んでいる。これが30分で解決できた大きな理由ではないかと言われている。)

しかし、英国政府通信本部はこれらの事実を公表しなかった。もちろん、特許も取っていない。1980年代に入り、コン





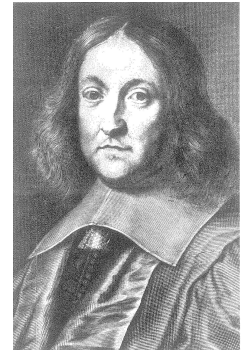
RSA 暗号誕生の頃の 3 人, 左からシャミア, リベスト, エイドルマン  
([35, p. 105] から引用)

そして, 公開鍵暗号現実化に決定的な役割を果たしたのは, 何と, 17 世紀フランスの数学者ピエール・ド・フェルマーが生み出した, 数に関する次のような簡単な定理であった.

フェルマーの小定理.

$p$  を素数,  $a$  を  $p$  の倍数ではない整数とすると,  $a^{p-1} - 1$  は  $p$  の倍数.

この定理は, 初等整数論と呼ばれる代数学の一分野の結果である. 整数論は, その理論の美しさゆえ, 19 世紀ドイツの数学者カール・フリードリヒ・ガウスが「数学は科学の女王であり, 数論は数学の女王である」と称え, 20 世紀イギリスの数学者 G.H. ハーディが著書『ある数学者の生涯と弁明』の中で, 「真の数学は, 戦争にはまったく役立たない. 数論を戦争に利用する方法はこれまで誰も見出してはいないのだ」と数論の美しさと極端な非応用性を誇り高く述べた分野であるが, その分野の古典的な定理が, 現代の情報社会を見えないところで支えているのである.



ピエール・ド・フェルマー  
(1607 年? - 1665 年)

では RSA 暗号の作り方を紹介しよう. 次のような自然数  $p, q, s, t$  を暗号の材料として用意する.

- (1)  $p, q$ : 異なる素数
- (2)  $s$ : 積  $(p-1)(q-1)$  と互いに素な自然数
- (3)  $t$ :  $st-1$  が  $(p-1)(q-1)$  の倍数となる自然数

コンピュータの能力が向上し, インターネットが生み出された後, RSA 暗号の特許は莫大な富をもたらすことになる. また, 公開鍵暗号や RSA 暗号発見の名誉も, 一般的にはディフィー, ヘルマン, シャミア, リベスト, エイドルマンに与えられている.

そして、2つの素数の積  $pq$  と自然数  $s$  を暗号の「公開鍵」として公表し、自然数  $t$  を「秘密鍵」として誰にも知られないようにしておく。また、 $p$  と  $q$  も誰にも知られないように秘密にしておかなければならない。

注 6. 秘密鍵  $t$  は  $(p-1)(q-1)$  より小さい自然数の範囲でただ1つだけ存在する (この後の §V で証明する)。このことは、実際の運用上では「秘密鍵は1個しかない」という意味合いをもつ。なぜならば、安全性を保証するため、 $p, q$  として、現時点では少なくとも  $pq$  が 309 桁以上 ( $2^{1024}$  は十進法で 309 桁であるので、 $pq$  が 1024 ビット以上と同意) になるように  $p, q$  を選ぶためである。

桁数の大きな素数  $p, q$  を見つける方法は整数論の研究者により研究されており、いい方法がある (例えば、[26] 参照)。そして、桁数の大きな2つの素数の積や、この後で説明する RSA 暗号の作業の途中に出てくる計算 (累乗等) は、すべてコンピュータで高速に計算できる<sup>\*27</sup>。また、 $s$  と  $(p-1)(q-1)$  が分かっているならば、秘密鍵  $t$  はすぐに計算される。しかし、公開鍵  $pq, s$  が分かっているとしても、これらから  $t$  をすぐに求めることはできない。もちろん  $p$  と  $q$  が分かれば、 $(p-1)(q-1)$  も分かり、 $t$  も分かる。しかし、桁数の大きな2つの素数の積  $pq$  が与えられても、コンピュータは、たとえそれが最新鋭のものでも、それを  $p$  と  $q$  に素因数分解するには天文学的な時間を要する。(これは、素因数分解を高速に行う上手い方法 (アルゴリズム) がまだ発見されていない<sup>\*28</sup> ことが原因である。) RSA 暗号は、このようなコンピュータの得手不得手を利用した暗号である。

ではこの「公開鍵」と「秘密鍵」を材料として、どのような手順で RSA 暗号の暗号化・復号が行われるのかを説明しよう。

## I. A さんは、私への通信内容を秘密したいとき。

大雑把に作業の流れを説明すると、A さんは、私の公開鍵  $pq, s$  を入手する<sup>\*29</sup>。そして、公開鍵を使って通信文 (平文) を暗号化する。通信文を受け取った私は、 $pq$  と秘密鍵  $t$  を使って復号する、となる。では、暗号化・復号の手順を説明しよう。

A さんは、送りたい文章をアスキー・コード等のコード表で数字列  $a$  に直し (数字列は、数字と見たとき、 $pq$  より小さくなるように、必要があれば幾つかに区切る。ここでは話を簡単にするため  $a < pq$  とする)、 $a$  を  $s$  乗し、さらにそれを  $pq$  で割り算したときの余り  $r$  を計算する。この  $r$  が暗号文である。これを私に送信する。

私は秘密鍵  $t$  を知っているので、受け取った暗号文  $r$  を  $t$  乗し、それを  $pq$  で割り算したときの余りを計算する。すると何と、その余りは元の数字列  $a$  となる !!! よって、アスキー・コード等で元の文章に戻せば A さんの文章が得られる。(復号される。)

実際にきちんと復号されることを証明する前に、簡単な練習問題で復号されることを確かめてみよう。

<sup>\*27</sup>もう少し丁寧に言えば、コンピュータで高速に計算することを可能にするアルゴリズム (解答手順を、単純な計算や操作の組み合わせとして明確に定義したもの) が存在する。詳細は、この後の注 7 参照。

<sup>\*28</sup>小さな素数から順番に割り算を行って余りが 0 かどうかを確認していくという原始的な作業の他に、一般数体ふるい法等のアルゴリズムが知られているが、2010 年に NTT と幾つかの研究機関が共同で 232 桁の合成数の素因数分解に成功しているレベルである。(このような研究は、RSA 暗号は何桁であれば安全なのかを検証する意味でも重要である。)

<sup>\*29</sup>もし、A さんと私がいつもメールでやりとりをしている間柄であれば、メールで公開鍵を送るように依頼すればよい。そうでなければ、A さんが私の公開鍵を検索し、公開鍵サーバー等から入手する。

## 練習問題.

2つの素数を  $p = 5$ ,  $q = 7$  とする. このとき  $pq = 35$ . 自然数  $s$  は  $(p-1)(q-1) = 4 \cdot 6 = 24$  と互いに素なものを取ればよかったので, 例えば  $s = 5$  を取ろう. 自然数  $t$  は  $st - 1$  が  $(p-1)(q-1) = 24$  の倍数であるものを取ればよかったので, 例えば  $t = 5$  を取ろう. このとき, 公開鍵は  $pq = 35$  と  $s = 5$ , 秘密鍵は  $t = 5$  となる. 今,

|      |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|
| (空白) | や | か | わ | は | あ | ず | さ | ま | た |
| 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

とひらがなと数字の対応をつけるとする.

このとき, 「かず」の暗号文を作成してみよう. まず, 「かず」を数字列に置き換えると 26 になる. さらに,

$$26^s = 26^5 = 11881376$$

11881376 を  $pq = 35$  で割った余りは

$$11881376 = 35 \times 339467 + 31$$

より 31. よって, 暗号文は 31.

次に, 暗号文 31 を復号してみよう.

$$31^t = 31^5 = 28629151$$

これを  $pq = 35$  で割ってみると

$$28629151 = 35 \times 817975 + 26$$

よって, 元の数字列は 26. つまり, 元の文は「かず」であることが分かる. 確かに, この例では正しく復号されている.

注 7. 実際に RSA 暗号を用いる場合,

$$26^s = 26^5 = 11881376$$

のような計算をそのまま行くと, 暗号文  $a = 26$  の部分や  $s = 5$  が大きな数の場合,  $a^s$  の桁数が非常に大きくなるため, コンピュータに負担がかかり, 高速に計算できない. したがって, 実際には次のような計算法 (アルゴリズム) を用いる.

$s = 5$  だと数が小さすぎて分かりにくいので,  $pq = 7 \cdot 11 = 77$ ,  $s = 53$  の場合で説明する. 53 を

$$\sum_{i=0}^n a_i 2^i \quad (\text{ただし, 各 } a_i \text{ は } 0 \text{ か } 1)$$

の形で表してみると (この表し方を **2進展開**と呼ぶ)

$$53 = 1 + 0 + 4 + 0 + 16 + 32$$

$$= 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5$$

したがって

$$a^{53} = a^{2^0} a^{2^2} a^{2^4} a^{2^5}$$

を計算すればいいことになる. これで, 53 乗まで計算しなくても, たかだか  $2^5 = 32$  乗までの計算でよいことになった. しかし, これでもまだ桁数が大きくて大変である. そして, 最終的に計算したいのは  $a^{53}$  を  $pq = 77$  で割った余りである. もうひと工夫しよう.  $a^{2^i}$  を 77 で割る計算を

$$a^{2^i} = q_i \cdot 77 + r_i \quad (\text{ここで, } 0 \leq r_i < 77)$$

と表す. このとき,



$$\begin{aligned}
 a^{53} &= a^{2^0} a^{2^2} a^{2^4} a^{2^5} \\
 &= (q_0 \cdot 77 + r_0)(q_2 \cdot 77 + r_2)(q_4 \cdot 77 + r_4)(q_5 \cdot 77 + r_5) \\
 &= r_0 r_2 r_4 r_5 + X \quad (\text{ここで, } X \text{ は } 77 \text{ のある倍数})
 \end{aligned}$$

と表される. これで,  $r_0, r_2, r_4, r_5$  だけ計算すればよいことになった. そして, これらは, わざわざ  $a^{2^i}$  を計算することなく, 帰納的に計算できる. 実際,

$$\begin{aligned}
 a^{2^{i+1}} &= a^{2^i \cdot 2} \\
 &= \left(a^{2^i}\right)^2 \\
 &= (q_i \cdot 77 + r_i)^2 \\
 &= r_i^2 + Y_i \quad (\text{ここで, } Y_i \text{ は } 77 \text{ のある倍数})
 \end{aligned}$$

であるので,  $r_{i+1}$  は  $r_i^2$  を 77 で割った余りであることが分かる. この方法を使うと,  $s$  が大きな数でも, 計算途中の桁数がそう大きくならずに  $a^s$  の計算が行える. このアルゴリズムは繰り返し自乗法と呼ばれている. 大きな数を扱う場合, このようなちょっとした工夫が計算時間を大幅に短縮する.

では次に, フェルマーの小定理を使って, 復号して元の  $a$  が得られることを証明してみよう. (この証明の部分は, 小学校専門科目「数学」では希望者に配布. オープンキャンパスの模擬授業や数学専攻での授業では全員に配布している.)

(証明) 何が行われているか, もう一度振り返ってみる.

- 送りたい文章  $a$  ( $a < p$  である数字列) の  $s$  乗  $a^s$  を計算する.
- 計算した  $a^s$  を  $pq$  で割り算して, その余り  $r$  を求める.  
( $a^s = mpq + r$  と表そう. もちろん  $m$  は商,  $r$  は余りなので  $0 \leq r < pq$ )
- 暗号文  $r$  の  $t$  乗  $r^t$  を計算する.
- 計算した  $r^t$  を  $pq$  で割り算して, その余り  $r'$  を求める.  
( $r^t = npq + r'$  と表そう. もちろん  $n$  は商,  $r'$  は余りなので  $0 \leq r' < pq$ )
- このとき  $r' = a$  (復号された)

そして, この手順の最後の部分  $r' = a$  が証明されればよい.

準備として,  $a^{st} - a$  は  $pq$  の倍数であることを証明しよう. そのためには,  $a^{st} - a$  が  $p$  の倍数かつ  $q$  の倍数であることを示せばよい. ( $\because p, q$  は互いに異なる素数だから.)

最初に  $a^{st} - a$  が  $p$  の倍数であることを, 次のような2つの場合に分けて証明しよう.

- (I)  $a$  が  $p$  の倍数のとき: このとき,  $a^{st}$  も  $p$  の倍数. よって,  $a^{st} - a$  もまた  $p$  の倍数.
- (II)  $a$  が  $p$  の倍数ではないとき:  $t$  の取り方より,  $st - 1$  は  $(p-1)(q-1)$  の倍数であった.

$$st - 1 = k(p-1)(q-1) \quad (\text{ここで, } k \text{ は整数})$$

と表そう. さらに,  $a$  は  $p$  の倍数ではないので, フェルマーの小定理より,  $a^{p-1} - 1$  は  $p$  の倍数. よって,

$$a^{p-1} - 1 = lp \quad (\text{ここで, } l \text{ は整数})$$

と表すとき,

$$a^{k(p-1)(q-1)} = (a^{p-1})^{k(q-1)} = (1 + lp)^{k(q-1)}$$

したがって, 2 項展開して,

$$a^{k(p-1)(q-1)} = 1 + X \quad (\text{ここで, } X : p \text{ のある倍数})$$

と表される. ゆえに,

$$\begin{aligned} a^{st} &= a^{1+k(p-1)(q-1)} \\ &= a a^{k(p-1)(q-1)} \\ &= a(1 + X) \\ &= a + aX \end{aligned}$$

よって,  $a^{st} - a = aX$ . つまり,  $a^{st} - a$  が  $p$  の倍数であることが示された.

同様の議論により,  $a^{st} - a$  が  $q$  の倍数であることも証明できる.

$\therefore a^{st} - a$  は  $pq$  の倍数.

$$a^{st} - a = X' \quad (\text{ここで, } X' : pq \text{ のある倍数}) \quad (*)$$

と表そう.

では, いよいよ証明してみる.

$r$  の導入の仕方より,  $a^s = mpq + r$  であったので,

$$r^t = (a^s - mpq)^t$$

したがって,

$$r^t = a^{st} + Y \quad (\text{ここで, } Y : pq \text{ のある倍数, つまり } pq \text{ の整数倍}) \quad (**)$$

と表される. ゆえに,

$$\begin{aligned} r' &= r^t - npq && (\because r' \text{ の導入の仕方より } r^t = npq + r' \text{ であったので}) \\ &= a^{st} + Y - npq && (\because (**)) \text{ より分かる.} \\ &= a + X' + Y - npq && (\because (*)) \text{ より分かる.} \end{aligned}$$

$\therefore r' = a \quad (\because 0 \leq r', a < pq, \text{ そして } X' + Y - npq \text{ は } pq \text{ の倍数だから})$

(証明終)

## II. 私は, 私の公開鍵を知っている A さんに, 署名した通信文を送るとき.

秘密鍵で暗号化することを (電子) 署名すると, そしてその暗号文を (電子) 署名 (文) と呼ぶ<sup>\*30</sup>. 大雑把に作業の流れを説明すると, 私は  $pq$  と秘密鍵  $t$  を使って署名文を作り, A さんは公開鍵  $pq$ ,  $s$  を使ってそれを復号する. これにより, 送り主の認証 (間違いなく私本人であることの確認) が行われる. では, 署名文の作成と復号の手順を説明しよう.

私は, 通信文を数字列  $a$  に直し ( $pq$  より小さい数字となるように, 必要があれば幾つかに区切る. ここでは話を簡単にするため  $a < pq$  とする),  $a$  を  $t$  乗 (秘密鍵乗) し, さらに  $pq$  で割り算したときの余り  $r$  を計算する. この  $r$  が署名文. これを A さんに送信する.

<sup>\*30</sup> 「(電子) 署名をする」「(電子) 署名 (文)」という用語は, ここで定義した RSA 暗号を用いた場合に限らず, 「(電子) 文書に対し, この文書は確かに私が書いたものであるということを証明できるなにか (をする)」という広い意味でも使われる.

Aさんは、私の公開鍵  $s$  を知っているのので、送られてきた署名文  $r$  を  $s$  乗し、さらにそれを  $pq$  で割り算したときの余りを計算する。すると、その余りは何と元の数字列  $a$  になる !!!

(証明) 既に「I. Aさんは、私への通信内容を秘密したいとき」は証明されているので、

$$(a^s)^t = a^{st} = a^{ts} = (a^t)^s$$

であることから分かる。

(証明終)

私の公開鍵  $s$  で復号できる署名文を作るには、この公開鍵とペアになる秘密鍵  $t$  を知っている必要がある。なぜなら、公開鍵  $s$  で復号される署名文を作るには、秘密鍵  $t$  を使わないと難しいし、秘密鍵  $t$  は  $(p-1)(q-1)$  より小さい自然数の範囲でただ1つしか存在しないから。そして、秘密鍵  $t$  を知っているのは私だけ。よって、この送信者は間違いなく私本人であると Aさんは判断できる。

講義では、以上の内容を解説した後で、次の問題に取り組んでもらった。

### 小テスト

練習問題と同じ暗号、つまり

公開鍵は  $pq = 35$  と  $s = 5$ 、秘密鍵は  $t = 5$

で、コード表も同じ

|      |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|
| (空白) | や | か | わ | は | あ | ず | さ | ま | た |
| 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

を使用するとき、次の問に答えよ。

- (1) 「やま」 を数字列に置き換えよ。
- (2) (1) の答を暗号化せよ。
- (3) (2) の答を復号し、「やま」 に戻ることを確認せよ。
- (4) 暗号文 18 を復号せよ。

## IV. 暗号使用の実際

ここでもう一度、共通鍵暗号と公開鍵暗号のメリット・デメリットをまとめてみよう。共通鍵暗号のメリットは、暗号化・復号の処理が高速に行える点であり、デメリットは鍵配送問題の存在、鍵の配布先からの漏洩問題、また通信先ごとに異なった鍵を準備する必要があり、その管理が大変な点などである。一方、RSA 暗号のメリットは、鍵の管理が楽なことである。実際、どの通信先にも同じ公開鍵を配布すればよく、秘密鍵を自分でキチンと保管できればよい。デメリットは暗号化・復号の処理速度が遅いことで、共通鍵暗号に比べ数百倍遅いと言われている。例えば、10M バイト程度の本文を暗号化する場合を考えみると (画像などであればこの程度の大きさにはすぐなる)、AES (128 ビット) が1秒間に 100M バイト程度処理できるとすると 0.1 秒で暗号化が完了するが、RSA なら 1 分程かかることになる。

このため、実際に暗号を使うときには、共通鍵暗号と公開鍵暗号の両方の長所を生かしたハイブリッド暗号方式を用いたり、署名するときには、ハッシュ関数<sup>\*31</sup> <sup>\*32</sup> を使って通信文から小さなサイズのハッシュ値を求め、そのハッシュ値だけを署名したりする工夫が行われている。次に、この2つの工夫について説明しよう。

### ● ハイブリッド暗号方式

Aさんが私にハイブリッド暗号方式で送信する場合で説明しよう。まずAさん(送信者)は共通鍵を作成する。そして、平文をその共通鍵を使って共通鍵暗号で暗号化する。と同時にその共通鍵を私の公開鍵で暗号化する。そして、その2つの暗号文を私に送信する。私は自分の秘密鍵で共通鍵を復号し、その共通鍵で暗号化されている通信文本体を復号して元の平文を得る。

なお、ハイブリッド暗号で使われる共通鍵は、通常1回使ったら次回以降は同じものを使わない。これにより鍵の漏洩の心配が無くなり、通信先ごとにいちいち鍵を管理する必要も生じない。1回の通信(セッション)に限って有効なことから、この共通鍵はセッション鍵と呼ばれる<sup>\*33</sup>。

注. 公開鍵暗号でまず共通鍵の交換を行い、その後、その共通鍵を使って共通鍵暗号通信を行うものもハイブリッド暗号方式と呼ばれる。後述するSSLでは、そのような使い方がされている。

### ● ハッシュ関数を用いた電子署名

私が、私の公開鍵を知っているAさんに、署名した通信文を送るときで説明しよう。あるハッシュ関数を使って、通信文(平文)のハッシュ値を求め、そのハッシュ値を署名し、それを元の通信文に添付してAさんに送る。Aさんは、署名文を復号してハッシュ値を得ると同時に、通信文のハッシュ値も計算して、その2つを比べる。もしその2つが一致していれば、その平文は間違いなく私が送信したものであることが分かる。しかも、通信文(平文)が改ざんされていないことも確認できる(脚注32参照)。

また、認証に関する問題点として、例えば「II. 私は、私の公開鍵を知っているAさんに、署名した通信文を送る」とき、Aさんが私の公開鍵と思っているものは、本当に私の公開鍵なのか?がある。つまり、悪意の第三者Mr.Xが私になりすましてMr.Xの公開鍵をAさんに送り、AさんはMr.Xを私だと思って、Mr.Xの署名文を私からの文章だと信じ込んだり、本来私に送信すべき情報をMr.Xに送信している可能性はないだろうか? また、私がAさんに送信した署名文を、私の公開鍵を知っているMr.Xが通信経路の途中で奪い取り、復号してその内容を知った後、それを改ざんしてMr.Xの署名を行いAさんに送信していないだろうか? いずれも、AさんがMr.Xの公開鍵を私の公開鍵であると思い込んでいることが原因である。したがって、これを解決するためには、通信相手を確認する手段があればよい。(特に電子商取引において重要である。)そして、その手段として、公開鍵の信頼性を確保するための公開鍵認証基盤(PKI: public key infrastructure)という仕組みが存在

<sup>\*31</sup>ハッシュ関数(要約関数とも呼ばれる)とは、任意の長さのデータを、小さな固定長のハッシュ値(要約値とも呼ばれる)に変換する、一方向性関数(簡単に計算できるが逆関数の計算は非常に困難である関数)である。 (「ハッシュ」は「細かく刻む」とか「ごちゃまぜにする」を意味する。)

ハッシュ関数の特徴を3つ挙げると、異なる2つのデータから同じハッシュ値が得られる可能性は非常に低い、ハッシュ値からデータが推測できない、同じハッシュ値をもつデータは簡単に作成できない、である。

ちなみに、署名に使われるハッシュ関数として、SHA-1 (Secure Hash Algorithm 1, ハッシュ値は160ビット)とその改訂版SHA-2が有名である。

<sup>\*32</sup>ハッシュ関数のその他の重要な使われ方として、改ざん防止がある。例えば、請求書を送信するとき、そのハッシュ値を添付して送信する。このとき、もし悪意の第三者が途中で請求書の内容を書き換えることがあっても、受信者が送られてきた請求書のハッシュ値を求めて、それが添付されているハッシュ値と一致しているかを確認することにより、容易に改ざんの有無が確認できる。

<sup>\*33</sup>鍵配送問題が解決されたことにより、セッション鍵を気軽に使うことができるようになった。そしてこれは、通信文を傍受していたNSAが暗号解読にかける経費を著しく増大させたと言われている。

する。公開鍵認証基盤では、信頼できる第三者 (TTP: Trusted Third Party) として認定された登録局 (RA: Registration Authority) と認証局 (CA: Certificate Authority) <sup>\*34</sup> が定められており、我々は次のようにして、それを利用することができる。

私は予め、パスポート、免許証、登記簿本、印鑑証明等の本人 (会社・組織) の確認のための書類を携帯して登録局に行き、私 (会社・組織) の公開鍵証明の申請を行う。登録局は私 (会社・組織) を審査・確認し、審査結果の責任を負う。審査結果は認証局に送られる。認証局は、あるハッシュ関数を使って私の公開鍵を含む証明内容 (平文) のハッシュ値を求め (証明内容の中に、どのようなハッシュ関数を用いたかも書かれている)、さらに認証局がそのハッシュ値を署名する。証明内容 (平文) にその署名文を付け加えたものが認証局が発行する電子証明書 <sup>\*35</sup> である。この証明書は私に発行される。

A さんに私の公開鍵を認証してもらいたいとき、私は A さんに認証局が発行した電子証明書を送信する。A さんは電子証明書を発行している認証局にアクセスして認証局の公開鍵を手に入れ、送られてきた署名文を、認証局の公開鍵で復号しハッシュ値を得る。と同時に、電子証明書の内容を指定されたハッシュ関数でハッシュ値に変換する。2つのハッシュ値が一致すれば、この電子証明書は改ざんされておらず、その内容を認証局が保証していることになる。これで、私の公開鍵が認証された。

ちなみに、通信参加者にあらかじめ安全にその公開鍵が配布されている認証局を、ルート認証局とよぶ。ルート認証局の公開鍵は、パソコンの OS や Web ブラウザに最初から組み込まれている。例えば、パソコンの OS である Windows 8 の場合、「コントロールパネル」から「ネットワークとインターネット」-「インターネットオプション」とクリックして開く「インターネットのプロパティ」というウィンドウで、「コンテンツ」タブをクリックして「証明書」ボタンをクリックし、開いたウィンドウの中の「信頼されたルート証明機関」タブをクリックすると、電子証明書のリストが現れる。そして電子証明書の中から、その機関 (ルート認証局) の公開鍵を取り出すことができる。さらに、ルート認証局以外にも中間認証局と呼ばれる認証局が多数存在し、いずれもあるルート認証局に自身の公開鍵を渡し、そのルート認証局から証明書を発行してもらうことで信頼性を保つ (その公開鍵が認証される)、さらに別の認証局はルート認証局に証明書を発行してもらった認証局から証明書を発行してもらい、…… と、上位の認証局から証明書を発行してもらうことにより、階層的にその信頼性を保っている <sup>\*36</sup>。

では、このセクションの残りの部分で、クレジットカード、インターネット・ショッピングにおける RSA 暗号の使用実態例と、メールの暗号化通信を簡単に実現するフリー・ソフトウェア PGP について紹介しよう。

## ● クレジットカードについて

2014 年、経済産業省は、東京オリンピックが行われる 2020 年までに、国内で流通するすべての

<sup>\*34</sup> 現在、民間の第三者認証機関は、RA と CA の両方の業務を行っているところがほとんどである。

<sup>\*35</sup> 2001 年に定められた「電子署名及び認証業務に関する法律」通称「電子署名法」によって、電子証明は紙の文書における署名や押印と同等の法的効力をもつとされている。

<sup>\*36</sup> 認証局に依頼する手間や経費なしに、なりすましを防ぐ、より身近な手段を 2 つ紹介しておく。(これらは、この後で紹介する PGP 等で用いられている。)

1 つ目は、信用の輪と呼ばれる方法である。これは、信頼の置ける公開鍵をもつ人に、第三者の公開鍵を署名してもらうことにより、その第三者の公開鍵を信頼の置けるものとみなしていく方法である。

2 つ目はハッシュ関数を用いる方法である。公開鍵のハッシュ値 (鍵指紋 (fingerprint) と呼ばれる) がもらった名刺等へ書き込まれていたり、あるいは直接本人から電話等で教えてもらったとき、それを公開鍵サーバから取ってきた公開鍵 (長いので名刺には書けないし、電話で聞き取るのも難しい) のハッシュ値と比較することにより、公開鍵を認証する方法である。

クレジットカードの IC カード化を表明した。IC カードとは、IC チップと呼ばれる集積回路を内蔵し、専用のリーダを介して端末等と通信を行うカードである。これまでの磁気ストライプのものは、記録された情報が簡単にスキミング<sup>\*37</sup>されてしまうが、IC チップは記憶されている秘密情報の外部からの読み取りが難しく、このため偽造カードを作るのも困難である。そして、IC チップが埋め込まれたクレジットカードを使って店舗で買い物をする場合、クレジットカードの真偽のチェックが、RSA 暗号を使って行われている。その仕組みはさまざまなものが存在するが、例えば次のようなものである。

カード発行会社はカード毎の公開鍵と秘密鍵を作成し、公開鍵を認証局に送る。認証局はカード会社から送られてきたカードの公開鍵を署名し、それをカード会社に送り返す。カード会社は、認証局から送られてきた署名文とこのカードの秘密鍵を、カードの IC チップに記憶させておく。

カード使用者が、加盟店で買い物をしてカード決済するとき、IC カードを端末に差し込むと、端末は乱数をカードに送る。カードは乱数をカードの秘密鍵で署名し、認証局が署名したカードの公開鍵と一緒に送り返す。加盟店の端末には認証局の公開鍵が入っているので、その公開鍵でカードの公開鍵を取り出し（この時点で、復号されたカードの公開鍵は、認証局によって認証された公開鍵であることになる）、さらにこの公開鍵を使って、カードが署名した乱数を復号する。これが元の乱数と一致していれば、このカードは偽造されたものではなく、正規のカードあることが分かる。（この後、使用者が正規のカードの持ち主であるかを、使用者に暗証番号を入力させて確かめる段階に入り、それが済むとやっと買い物の金額等、この買い物に関する段階に進む。）

#### ● インターネット・ショッピングについて

インターネットが使われ始めた頃は、単に静的なページが配布されるだけであった。しかし 1994 年、クレジットカードでの買い物等の電子商取引やオンライン銀行などの金融取引を、インターネットで安全に実行したいという企業の要求を実現するため、当時の最有力ブラウザ・ベンダーであった Netscape Communication 社が、インターネット上で情報を暗号化して送受信する仕組みである SSL (Secure Sockets Layer: セキュア・ソケット・レイヤー) を開発した<sup>\*38</sup>。SSL は何度も改訂され、現在でも最も広く使われている<sup>\*39</sup>。SSL のおかげで、ネットショップの会員 ID やパスワード、氏名、住所、メールアドレス等の個人情報、クレジットカードの情報が守られ、インターネット・ショッピング等を安全に行うことが可能となっている。

SSL にはハイブリッド暗号方式が用いられる。その手順は次の通りである。利用者が SSL が利用されている店舗のウェブページ (URI が http ではなく https<sup>\*40</sup> のウェブページ。鍵のマークが表示される) に接続すると、どの公開鍵暗号と共通鍵暗号を使って通信を行うのかを、まずウェブページのサーバー (店舗のコンピュータ上で動作するプログラム) と利用者のブラウザ (利用者のコンピュータ上で動作するプログラム。Internet explorer 等) の間で決定し、その後、そのウェブページのサーバーの電子証明書 (もちろんある認証局の発行) がウェブページのサーバーから利用者のブラウザに送られてくる。利用者のブラウザには、主な認証局の公開鍵が予め入っているので、それを使って電子証明書の改ざんとサーバーのなりすましを検証する。検証結果が正しければ、そのサー

<sup>\*37</sup>磁気記録情報を不正に読み出して、コピーを作成し使用する犯罪行為。「スキマー」と呼ばれる、カード情報を読み取る装置を使って情報を複製する。

<sup>\*38</sup>Netscape Communication 社の主力製品であったブラウザ Netscape Navigator に実装された。その後、多くのブラウザで使われ、事実上の業界標準になった。Microsoft 社が、OS である Windows98 に、ブラウザ Internet explorer を標準搭載するようになったため、Netscape Navigator はその後苦戦することになる。

<sup>\*39</sup>SSL から派生した、TLS (Transport Layer Security: トランスポート層セキュリティ) も使われており、順次 SSL は TLS に移行していくだろうと思われる。ただ、SSL が普及しているので TLS も含めて SSL と呼ばれることが多い。

<sup>\*40</sup>https は HyperText Transfer Protocol over Secure Socket Layer (SSL) を略したもの。



バの電子証明書は認証局から発行された正当な証明書であることが立証され、それが証明しているサーバー（ウェブサイト）とその公開鍵が確認できたことになる。次に、利用者のブラウザは適当に乱数を発生させ、通信本体に使う共通鍵暗号用の鍵を作成し、それをサーバーの公開鍵で暗号化してサーバーに送信する。サーバーはそれをサーバーの秘密鍵で復号する。これで両者は共通鍵を手に入れた。後はその共通鍵を使って、共通鍵暗号 (DES 等) で通信を行う。

### ● PGP について

クレジットカードを使うときやインターネット・ショッピングをするときなど、知らぬ間に RSA 暗号を使っていることは既に述べた。しかし、自主的に RSA 暗号を扱うことはないだろうと思っている人も多いのではないだろうか。なぜなら、RSA 暗号の理屈が分かって、大きな 2 つの素数  $p, q$  を見つけるのも大変だし、さらに大きな数  $(p-1)(q-1)$  に対して  $s, t$  を見つけたり、手順通り計算を進めるのは大変そうだから。

しかし実は、誰でも簡単に RSA 暗号を使ったハイブリッド暗号方式で暗号化してメールを送信したり、電子署名できるソフトウェアが存在する。しかもこれは無料でダウンロードできるのである<sup>\*41</sup>。それを使うと、簡単な操作で自分用の公開鍵・秘密鍵やセッション鍵が自動的に作成され、作成した公開鍵を、自分のウェブサイトや公開鍵サーバ等で公開したり、他人に送信したり、名刺に載せるためにその鍵指紋を確認したり、秘密鍵（中味は複雑な数字列で扱いにくい）を指定したキーフレーズで呼び出せるようにしておいて、自分の公開鍵で暗号化されたメールが届いたら、簡単な操作で瞬時に復号してくれたり、他の人の公開鍵をメーラーに登録しておき、その人にメールを送る場合に、その公開鍵を呼び出して瞬時にハイブリッド暗号方式で送信することができる。

そのようなソフトウェアで、最初に作られ、今でも最もよく使われているものが PGP (Pretty Good Privacy) である。これは物理学とコンピュータ科学を学んだフィル・ジーマーマンによって 1991 年にインターネット上にフリー・ソフトウェアとして公開され、アメリカ政府との 1990 年代の法廷論争<sup>\*42</sup>を経て、現在は標準的なソフトウェアとして認識されるようになっている。

## V. もう少し詳しく説明してみよう

講義では、必要最小限の知識で RSA 暗号を理解してもらうため、フェルマーの小定理だけで解説をおこなった。しかし、素数  $p, q$  に対して、 $(p-1)(q-1)$  という数が出てきていることから、実はリベスト、シャミア、エイドルマンの頭の中には、フェルマーの小定理の拡張形であるオイラーの定理 (1760 年) があったに違いない。事実、RSA 暗号は、オイラーの定理を用いて解説されることが一般的である。この章では、オイラーの定理の視点から、RSA 暗号を再考してみよう。

まずは、フェルマーの小定理を再登場させる。

### フェルマーの小定理

$p$  を素数、 $a$  を  $p$  の倍数ではない整数とすると、 $a^{p-1} - 1$  は  $p$  の倍数。

本学・数学教育専攻の学生は、2 回生後期・代数学 C の剰余環のところ、素数  $p$  に対し剰余環  $\mathbb{Z}/p\mathbb{Z}$  が体をなすことを学んだ後、その系としてこの定理を学んでいる。その折の証明は次のような、実に短いものである。

<sup>\*41</sup>ただし、対応しているメーラーである必要はある。また商業用に利用する場合は有料。

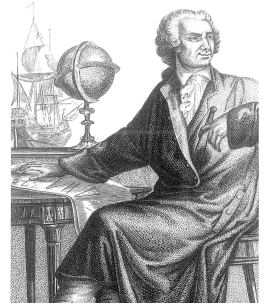
<sup>\*42</sup>アメリカ政府は、暗号化ソフトウェアを軍需品に指定しており、国務省の許可なしに輸出することを禁じていた。そして、ジーマーマンはインターネットを介して軍需品を輸出した武器商人として告発されたのである。3 年間の審査の末、1996 年になってアメリカ政府は不起訴を決定する。その背景には、1994 年から 1996 年にかけて、ジーマーマンが電子フロンティア協会のパイオニア賞をはじめとする多数の賞を受賞して、すっかり有名人になったことや、ネットワーク・セキュリティに対する一般の関心が、数年前に比べて格段に高まっていたことがあるのではないかと噂されている。

(証明) 乗法群  $(\mathbb{Z}/p\mathbb{Z})^\times$  の位数は  $p-1$  であり, その任意の元  $a+p\mathbb{Z}$  (ここで,  $a \in \mathbb{Z}$ ) に対し  $(a+p\mathbb{Z})^{p-1} = a^{p-1} + p\mathbb{Z}$  は  $\mathbb{Z}/p\mathbb{Z}$  の単位元  $1+p\mathbb{Z}$  に一致する. このことから,  $a^{p-1} - 1 \in p\mathbb{Z}$ . つまり  $a^{p-1} - 1$  は  $p$  の倍数である.

(証明終)

ここでの素数  $p$  を任意の自然数  $n$  に拡張した結果が, 次に紹介するオイラーの定理である. そこにはオイラー関数と呼ばれる概念が使われている. まずこれを定義しよう.

$n$  を自然数とせよ.  $1, 2, \dots, n$  の中で,  $n$  と互いに素なものの個数を  $\varphi(n)$  で表し, オイラー関数とよぶ. つまり,  $\varphi(1) = 1, \varphi(2) = 1, \varphi(3) = 2, \varphi(4) = 2, \varphi(5) = 4, \varphi(6) = 2, \varphi(7) = 6, \dots$ . 特に,  $n$  が素数  $p$  であるときを考えると,  $1, 2, 3, \dots, p-1$  がすべて  $p$  と互いに素なので  $\varphi(p) = p-1$  であることが分かる.



レオンハルト・オイラー  
(1707 年 - 1783 年)

これを用いて, フェルマーの小定理は次のように一般化される.

#### オイラーの定理

$n$  を自然数,  $a$  を  $n$  と互いに素な整数とすると,  $a^{\varphi(n)} - 1$  は  $n$  の倍数.

(証明)  $\mathbf{X}$  を  $n$  と互いに素な  $1$  以上  $n-1$  以下の自然数の集合とすると, 剰余環  $\mathbb{Z}/n\mathbb{Z}$  の単元群<sup>\*43</sup>  $U(\mathbb{Z}/n\mathbb{Z})$  は  $\{a+n\mathbb{Z}\}_{a \in \mathbf{X}}$  に他ならない. ( $a$  と  $n$  が互いに素であることは, 整数  $k, l$  で  $ka+ln=1$  を満たすものが存在することと同値であることから分かる.) よって,  $U(\mathbb{Z}/n\mathbb{Z})$  の位数は  $\mathbf{X}$  の数の数, つまり  $\varphi(n)$  である. したがって,  $\mathbf{X}$  の任意の元  $a$  に対し,  $(a+n\mathbb{Z})^{\varphi(n)} = a^{\varphi(n)} + n\mathbb{Z}$  は  $U(\mathbb{Z}/n\mathbb{Z})$  の単位元  $1+n\mathbb{Z}$  に一致する. ゆえに,  $a^{\varphi(n)} - 1 \in n\mathbb{Z}$ . つまり,  $a^{\varphi(n)} - 1$  は  $n$  の倍数.

(証明終)

そして, オイラー関数は次の定理の (1) ように具体的に計算でき, これを用いると次の定理の (2) (積公式) が得られる.

#### オイラー関数の公式

(1)  $n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$  を自然数  $n$  の素因数分解とすると,

$$\varphi(n) = n \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_r}\right)$$

と表される.

(2) 自然数  $m, n$  に対し, これらが互いに素であることと,

$$\varphi(mn) = \varphi(m) \varphi(n)$$

が成り立つことは同値である.

証明. まず,  $n = p^e$  ( $p$ : 素数) の場合を考えよう.  $\forall a \in \{1, 2, \dots, n\}$  に対して,  $a$  と  $n$  は互いに素ではないことと,  $a$  が  $p$  の倍数であることは同値である. そして,  $\{1, 2, \dots, n = p^e\}$  の中で  $p$  の倍数は,

$$p, 2p, 3p, \dots, pp^{e-1} = p^e = n$$

<sup>\*43</sup>環  $R$  に対し, 乗法に関する逆元をもつ元を単元と呼ぶ. そして,  $R$  の単元全体の集合は  $R$  の乗法に関して群をなす. この群を単元群と呼び, 通常  $U(R)$  で表す. 特に  $R$  が体のとき,  $U(R)$  は乗法群  $R^\times$  に他ならない.

の  $p^{e-1}$  個だけである. よって,  $\varphi(p^e)$  は  $p^e$  個の元からこれらを除いた残りの元の数であるので,

$$\varphi(p^e) = p^e - p^{e-1} = p^e \left(1 - \frac{1}{p}\right)$$

次に, 一般の  $n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$  のときを考えよう. このとき, 写像

$$\mathbb{Z}/n\mathbb{Z} \rightarrow (\mathbb{Z}/p_1^{e_1}\mathbb{Z}) \oplus (\mathbb{Z}/p_2^{e_2}\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/p_r^{e_r}\mathbb{Z}), \quad a+n\mathbb{Z} \mapsto (a+p_1^{e_1}\mathbb{Z}, a+p_2^{e_2}\mathbb{Z}, \dots, a+p_r^{e_r}\mathbb{Z})$$

は環同型. したがって,

$$\begin{aligned} \varphi(n) &= |U(\mathbb{Z}/n\mathbb{Z})| \\ &= |U((\mathbb{Z}/p_1^{e_1}\mathbb{Z}) \oplus (\mathbb{Z}/p_2^{e_2}\mathbb{Z}) \oplus \cdots \oplus (\mathbb{Z}/p_r^{e_r}\mathbb{Z}))| \\ &= |U(\mathbb{Z}/p_1^{e_1}\mathbb{Z}) \oplus U(\mathbb{Z}/p_2^{e_2}\mathbb{Z}) \oplus \cdots \oplus U(\mathbb{Z}/p_r^{e_r}\mathbb{Z})| \\ &= |U(\mathbb{Z}/p_1^{e_1}\mathbb{Z})| |U(\mathbb{Z}/p_2^{e_2}\mathbb{Z})| \cdots |U(\mathbb{Z}/p_r^{e_r}\mathbb{Z})| \\ &= \varphi(p_1^{e_1}) \varphi(p_2^{e_2}) \cdots \varphi(p_r^{e_r}) \\ &= \left(p_1^{e_1} \left(1 - \frac{1}{p_1}\right)\right) \left(p_2^{e_2} \left(1 - \frac{1}{p_2}\right)\right) \cdots \left(p_r^{e_r} \left(1 - \frac{1}{p_r}\right)\right) \\ &= n \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_r}\right) \end{aligned}$$

(2)  $m, n$  の素因数分解を  $m = p_1^{e_1} \cdots p_r^{e_r}$ ,  $n = q_1^{f_1} \cdots q_s^{f_s}$  と表す. このとき,

$m, n$ : 互いに素な自然数  $\iff \{p_1, \dots, p_r\}$  と  $\{q_1, \dots, q_s\}$  には重なりがない.

$$\iff \varphi(mn) = mn \left(1 - \frac{1}{p_1}\right) \cdots \left(1 - \frac{1}{p_r}\right) \left(1 - \frac{1}{q_1}\right) \cdots \left(1 - \frac{1}{q_s}\right)$$

$$\iff \varphi(mn) = \varphi(m) \varphi(n)$$

(証明終)

この公式から, 異なる素数  $p, q$  に対し,

$$\varphi(pq) = \varphi(p) \varphi(q) = \left(p \left(1 - \frac{1}{p}\right)\right) \left(q \left(1 - \frac{1}{q}\right)\right) = (p-1)(q-1)$$

であることが分かる. したがって, RSA 暗号に出てくる  $(p-1)(q-1)$  とは, 積  $pq$  のオイラー関数値  $\varphi(pq)$  に他ならないことになる.

以上の基礎知識の下で, 次を証明しよう.

- (1) 数字列  $a$  に対し,  $a^{st}$  を  $pq$  で割ると, 余りは  $a$  (RSA 暗号の証明)
- (2) RSA 暗号において, 公開鍵  $s$  に対し, 秘密鍵  $t$  は  $(p-1)(q-1)$  以下の自然数の範囲でただ1つである.  
 $\left( \begin{array}{l} \text{実際に運用されるとき, } p, q \text{ はとても大きな数であるので, このことは公開鍵 } s \\ \text{に対して, 秘密鍵 } t \text{ は事実上ただ1つだけであることを意味している.} \end{array} \right)$
- (3) 逆に,  $s, t$  が定まっているとき,  $t$  の満たしている条件「 $st-1$  が  $(p-1)(q-1)$  の倍数」を満たす  $s$  は,  $t$  に対して  $(p-1)(q-1)$  以下の自然数の範囲でただ1つである.  
 $\left( \begin{array}{l} \text{実際に運用されるとき, } p, q \text{ はとても大きな数であるので, このことは秘密鍵 } t \\ \text{に対して, 公開鍵 } s \text{ は事実上ただ1つだけである, つまり RSA 暗号は電子署名} \\ \text{の役割を果たせることを意味している.} \end{array} \right)$

ただし、オイラーの定理を用いて証明できることを示すのが目的であるので、(1) の証明は、 $a$  と  $pq$  が互いに素の場合だけを証明する。(数字列  $a$  が  $pq$  と互いに素ではない場合、つまり  $a$  が  $p$  や  $q$  の倍数である場合は、既に表示した証明で済ませばよい.)

(証明) まず、次の (i), (ii), (iii) が成り立つことを示そう.

(i)  $st = 1 + \varphi(pq)u$  を満たす整数  $u$  の存在が、 $s, t$  の取り方から分かる.

(ii)  $s + \varphi(pq)\mathbb{Z}$  と  $t + \varphi(pq)\mathbb{Z}$  は、単元群  $U(\mathbb{Z}/\varphi(pq)\mathbb{Z})$  において互いに逆元の関係にある.

(iii)  $a^{\varphi(pq)} = 1 + pqv$  を満たす整数  $v$  の存在が、オイラーの定理から分かる.

(i) の証明：自然数  $t$  は、 $st - 1$  が  $(p-1)(q-1) = \varphi(pq)$  の倍数になるように取った. このことから、

$$st + \varphi(pq)\mathbb{Z} = 1 + \varphi(pq)\mathbb{Z} \quad (*)$$

が分かる. つまり、 $u$  は存在する.

(ii) の証明：自然数  $s$  は、 $(p-1)(q-1) = \varphi(pq)$  と互いに素な自然数として取った. これは

$$s + \varphi(pq)\mathbb{Z} \in U(\mathbb{Z}/\varphi(pq)\mathbb{Z})$$

を意味している. そして、(\*) は、 $U(\mathbb{Z}/\varphi(pq)\mathbb{Z})$  において、

$$(s + \varphi(pq)\mathbb{Z})(t + \varphi(pq)\mathbb{Z}) = 1 + \varphi(pq)\mathbb{Z} \quad (: U(\mathbb{Z}/\varphi(pq)\mathbb{Z}) \text{ の単位元})$$

を意味している. つまり、(ii) は成り立つ.

(iii) の証明：今、 $a$  は  $pq$  と互いに素と仮定しているので、オイラーの定理より  $a^{\varphi(pq)} - 1$  は積  $pq$  の倍数. よって  $v$  は存在する.

$$\begin{aligned} (1) \quad a^{st} &= a^{1+\varphi(pq)u} && (\because (i)) \\ &= a^1 (a^{\varphi(pq)})^u \\ &= a(1 + pqv)^u && (\because (iii)) \\ &= a + X && (\text{ここで、} X \text{ は } pq \text{ のある倍数}) \end{aligned}$$

$\therefore a^{st}$  を  $pq$  で割った余りは  $a$  である. ( $\because a < pq$ )

(2), (3) (ii) と逆元の一意性から成り立つことが分かる.

(証明終)

## VI. その他の公開鍵暗号

公開鍵暗号は、未だに RSA 暗号が最も普通に使われる公開鍵暗号であるが、その他にも存在する. その中からエルガマル暗号と楕円曲線暗号を簡単に紹介しよう.

### ● エルガマル暗号

1984 年にエジプト人暗号学者 Taher Elgamal により発明された暗号.  $\mathbb{Z}/p\mathbb{Z}$  を素体,  $s + p\mathbb{Z}$  を乗法群  $(\mathbb{Z}/p\mathbb{Z})^\times$  の生成元 ( $p$  が素数であることから、乗法群  $(\mathbb{Z}/p\mathbb{Z})^\times$  は巡回群になり、このような  $s$  は常に存在する),  $t + p\mathbb{Z} \in (\mathbb{Z}/p\mathbb{Z})^\times$  とするとき、 $(s + p\mathbb{Z})^m = t + p\mathbb{Z}$  を満たす自然数  $m$  を求める効率的なアルゴリズムは未だに発見されていない ( $m$  を求める問題は離散対数問題と呼ばれる). しかし、自然数  $m$  が与えられているとき  $(s + p\mathbb{Z})^m$  は高速に計算できる. エルガマル暗号はこの事実を利用した暗号である.

暗号の手順を説明しよう. 次の自然数  $p, s, m$  を準備する.  $p$  は非常に大きな素数,  $s$  は  $p-1$  以下 (必要ではないが  $s+p\mathbb{Z}$  が乗法群  $(\mathbb{Z}/p\mathbb{Z})^\times$  の生成元であることが望ましい),  $m$  は  $p-2$  以下とする. そして,  $s^m$  を  $p$  で割った余り  $r$  を計算し,  $(p, s, r)$  を公開鍵,  $m$  を秘密鍵とする.

平文を数値化した  $a$  (ただし,  $a < p$  とする) に対して, 任意に選んだ  $p-1$  以下の自然数  $t$  を用いて次のように計算した 2 つの数の組  $(c_1, c_2)$  が暗号文となる.

$$\begin{cases} c_1: s^t \text{ を } p \text{ で割った余り} \\ c_2: a r^t \text{ を } p \text{ で割った余り} \end{cases}$$

次に, 暗号文  $(c_1, c_2)$  を受け取ったとする.  $s+p\mathbb{Z} \in (\mathbb{Z}/p\mathbb{Z})^\times$  (: 乗法群) であるので  $c_1 \neq 0$ . つまり,  $1 \leq c_1 \leq p-1$  である. よって,  $p$  が素数であることから,  $c_1$  と  $p$  は互いに素であることが分かり, 整数  $u, v$  が  $u c_1 + v p = 1$  を満たすように取れる. このとき,  $u^m c_2$  を  $p$  で割った余りを計算すると, これは元の  $a$  である.

(注. ここでの割り算は, 割られる数を整数に拡張したものを意味している.)

(証明)  $s^m = pq + r$ ,  $s^t = pq_1 + c_1$ ,  $a r^t = pq_2 + c_2$  と表すことにする. このとき,

$$\begin{aligned} u^m c_2 &= u^m (a r^t - p q_2) && (\because a r^t = p q_2 + c_2) \\ &= u^m (a (s^m - p q) - p q_2) && (\because s^m = p q + r) \\ &= a u^m s^{m t} + X && (\text{ここで, } X: p \text{ の倍数, つまり } p \text{ の整数倍}) \\ &= a u^m (p q_1 + c_1)^m + X && (\because s^t = p q_1 + c_1) \\ &= a u^m c_1^m + Y && (\text{ここで, } Y: p \text{ の倍数}) \\ &= a (1 - v p)^m + Y && (\because u c_1 + v p = 1) \\ &= a + Z && (\text{ここで, } Z: p \text{ の倍数}) \end{aligned}$$

$\therefore u^m c_2$  を  $p$  で割った余りは  $a$  ( $\because a < p$ )

(証明終)

### ● 楕円曲線暗号

1985 年にワシントン大学のニール・コブリッツと IBM のビクタ・ミラーがほぼ同時期に別々に考案した暗号. エルガマル暗号を, 乗法群  $(\mathbb{Z}/p\mathbb{Z})^*$  ではなく, 標数が素数  $p$  である有限体  $K$  上の楕円曲線から得られる有限群の巡回部分群に適用して得られる暗号. 楕円曲線とは,  $p \geq 5$  以上のとき

$$y^2 = x^3 + ax + b \quad (\text{ただし, } a, b \in K \text{ は } 4a^3 + 27b^2 \neq 0 \text{ を満たす})$$

と表される曲線のことで,  $x \mapsto \infty$  のとき  $y \mapsto \infty$  と考えて  $0 := (\infty, \infty)$  もこの曲線の点と考える. このとき,

$$E(K) := \{ (x, y) \in K^2 \mid y^2 = x^3 + ax + b \} \cup \{0\}$$

は, 加法公式と呼ばれる公式で与えられる加法により  $0$  を零元とする有限加法群になる. 乗法群  $(\mathbb{Z}/p\mathbb{Z})^*$  は  $p$  に対してただ 1 つであるが, 楕円曲線から得られる有限群は  $a, b$  の取り方でさまざまに変化するため, 安全な暗号を作るために必要な  $p$  の大きさがずっと小さくて済む. 具体的には,  $(\mathbb{Z}/p\mathbb{Z})^*$  なら十進法で 300 桁が必要なら, 楕円曲線なら 50 桁くらいでよい. そのため, 高速に計算することができるのが楕円曲線暗号の利点である. この利点のため, 将来性をもった公開鍵暗号であると言われている.

## VII. 講義に対する受講生の反応

この講義で学んだことや感じたことを自由に記述してもらったところ、下記のような意見が寄せられた。(括弧内は学生の所属専攻名。)

- 数学が私たちの生活にどれほど深く関わっているのかを改めて感じた。(教育科学専攻)
- 普段の生活で数学を意識することはほとんどありませんが、毎日使っているネットも数学によって成り立っていることが改めて分かりました。(特別支援教育教員養成課程)
- これまで数学を学んでもあまり意味がないと思っていたが、意外と数学を活用する機会がおおいなと感じた。この驚きを子どもに伝えることで、算数や数学に少しでも興味を持ってもらえるような実践をしたいと思った。(特別支援教育教員養成課程)
- 数学の応用は私たちの生活に欠かすことのできないものだった。(保健体育専攻)
- コンピュータを使って暗号を解読するので、暗号の作り方はとても複雑だろうと予想した。しかし、計算自体は面倒くさいが、仕組みはとても簡単であり、目を疑った。また、コンピュータにも得手不得手があり、不得意なものは数百年かかるということにも驚いた。(理科教育専攻)
- 暗号は難解なものだと今まで考えていたが、意外に簡単に計算できるのを感じた。(社会科教育専攻)
- 暗号という言葉の存在や意味はもちろん知っていたが、こんなに深かったのかと感心した。やはり、「深くやればやるほど、おもしろいのだ」と再確認した。(教育科学専攻)
- 暗号は単なる子どもの遊びや推理小説のトリックに用いられるものではなく、意外と身近に存在し、役に立つものであることが分かった。(教育科学専攻)
- RSA 暗号は、仕組みは簡単なのに解読されないという、すごい暗号だと思いました。(書道教育専攻)
- RSA 暗号は、コンピュータの得手不得手を利用した暗号であるというのは面白いものだと思う。こういったことを研究する数学者について調べることも興味深いことだと感じた。(教育科学専攻)
- 昔から推理小説を読むのが好きだったので、文字を数字列に置き換えて暗号化したり、逆にそれを復号して文字に直したりすることで探偵になった気がして楽しかった。(国語教育専攻)
- こんな数字の使い方をしたのははじめてだったし、数がいろんな分野で活躍していたことも勉強になりました。(美術教育専攻)
- まさか数式を使って暗号を作ったりしているなんて思わなかったのが驚きました。コンピュータの不得意な部分を利用した暗号を作るなんて、やっぱり人間の脳もすばらしいと思いました。(書道教育専攻)
- 鍵が上手い具合に計算と結びついて、すごいなーと感じた。(幼稚園教員養成課程)
- こんな暗号を考えつくなんてすごいなあと思いました。(教育科学専攻)
- 暗号の難しさが分かったが、そうした先人の知恵に助けられて、私たちが今の生活をできていることを思うと感動した。(教育科学専攻)
- 日頃受けている数学の授業とは違った類の話でしたので、講義を聞いていてとても興味をもつことができました。前回の授業(確率)もおもしろかったのですが、小学生相手でも使えるのは今回の方だと思います。(教育科学専攻)
- こんな感じで暗号化したり、それを復号するのはどこかかっこいいと思いました。いつか、この授業を知らない誰かにどや顔で説明して、このようなやりとりをする機会があれば、ぜひそれをして、秘密結社の一員になったつもりになりたいと思いました。難しくても面倒だけでも楽しかったです。(国語教育専攻)
- 暗号の仕組みをある程度理解できたし、暗号の歴史的背景はとても興味深いものであった。戦時下において、暗号は重要な役割を果たしていたことが分かったし、その歴史をひも解くことで、当時の歴史にも言及できると思った。(教育科学専攻)
- 今、私たちが使っている暗号が、どのような歴史的背景を持つのか、どのような仕組み、工夫点を持つのかというお話は興味深かったです。(国語教育専攻)
- 数学の暗号というテーマから歴史の話まで広がって面白かったです。大戦前から解読されていたかもという話を聞いて、とてもヘンな気持ちになりました。(美術教育専攻)
- エニグマ暗号機など、戦争中に暗号が使われている話が興味深かったです。技術に勝る国が戦争にも勝つのだなあと思い知らされました。(特別支援教育教員養成課程)

- シーザー暗号やRSA暗号などは知っていましたが、実際に自分で計算して解読したことはなかったので、面白かったです。また、暗号は人によって考えられたのだと改めて実感し、すごいなと思いました。普段何も気にしていなくても、インターネットやカードなど、身の回りにはたくさんの暗号があふれていることに気付きました。(教育科学専攻)
- 練習問題の解説を聞いても、いまいちピンとこなかったのですが、このテストで実際に解いてみることで、何となくですが、RSA暗号を体験できたような気がします。解いているときは、クイズを考えているような感じで楽しかったです。ワクワク感もあって面白い問題だなと思いました。(国語教育専攻)
- 説明を聞いているときは、自分とはかけ離れた話だったのでよく分からなかったが、計算してみると案外簡単でした。(国語教育専攻)
- 恥ずかしながら、自分の頭では理屈を理解することはできませんでしたが、最後のテストで、作業だけはできた気がしました。それで、すごいなというのを(漠然とですが)感じました。(国語教育専攻)
- 暗号はあまり身近でなかったのが難しかったけれど、練習問題などで実際に解いてみると、意外に分かりやすく理解が深まりました。膨大な量の暗号がコンピュータで実際に使われているんだと思うと、私たちの生活に暗号は実は欠かせないんだと思いました。(幼稚園教員養成課程)
- (4)の問題を解き終えて、「やま」-「かわ」ああ!! と思ってしまいました。(国語教育専攻)
- 暗号が解けたときはちょっとうれしかったです。(特別支援教育教員養成課程)
- 暗号化したものを復号して元の形に戻ったときは少し感動しました。(幼稚園教員養成課程)

## VIII. 最後に

現時点で、RSA暗号は十分な安全性をもった信頼に足る暗号であるとされている。そして、暗号の歴史に於いて、現在は安全な暗号が存在する時代であると言える。しかし、暗号の歴史は、安全な暗号が存在する時代と、それが破られ存在しない時代が交互に訪れている。つまり、たまたま我々は安全な暗号が存在する時代に生きており、それにより、知らないところで情報技術に支えられて、日々の生活が安全に営まれているだけかもしれないのである。実は、量子コンピュータと呼ばれるものが完成されたなら、RSA暗号は(そして現在使われているすべての暗号は)もう使い物にならなくなると言われている<sup>\*44</sup>。今使われているコンピュータは古典物理学の法則に従って作動している。一方、量子コンピュータは量子物理学の法則に従って作動するコンピュータである。量子コンピュータは量子物理学の法則のおかげで一度に複数の処理が可能で、完成されれば、現代のスーパーコンピュータで数百年かかる計算を数秒で行うとも言われている。しかし、量子コンピュータをそのような桁違いの処理速度を持つものとするならば、現時点では量子コンピュータであると言えるものは存在していないし、そのようなコンピュータの実用化はまだ20年から50年かかるだろうという専門家が大半を占める。また同時に、量子論を用いた量子暗号と呼ばれる新しい暗号も研究されており、量子コンピュータ完成までに実用化されるかが焦点となっている。安全な暗号が常に存在し、社会の安定が保たれるため、実は暗号の研究は現代社会にとって最も重要なもののひとつなのである。

世界の暗号技術は莫大な費用をつぎ込んで秘密裏に研究されており、日々進化していると思われる。英国政府通信本部で、実は世界で最初に公開鍵暗号のアイデアを生み出していたエリスは、暗号にまつわる秘密主義に関して次のように書き残している。「暗号に関する秘密を公開することが許されるのは、秘密にしてもこれ以上利益はないことが明らかになった後に、ただ歴史的正確さを期すためでしかないのである。」つまり、我々が(知っているところや)知らないところで新しい暗号が開発され、我々が(知っているところや)知らないところで暗号が破られ、我々が(知っているところや)知らないところで暗号は改良される。この繰り返しが行われている。今、我々が安全であると考えて使っている暗号技術も、実はどこかの国の秘密情報機関ですでに破られている技術なのかもしれないのである。

<sup>\*44</sup>1994年にAT&Tベル研究所の数学者ピーター・ショアが量子コンピュータのプログラムの定義に成功しており、量子コンピュータが完成されれば、因数分解を高速に行い、RSA暗号を解読できることは分かっている。



謝辞. 山口大学理工学研究科の菊政勲教授には、著者の再三に渡る暗号に関する著者の質問について丁寧に回答・解説して頂き、さらに、この論文を通読して、問題点の指摘や有益なアドバイスを頂きました。また、大阪教育大学・教養学科・情報科学講座の藤井淳一教授にもこの論文を通読して頂き、有益なアドバイスを頂きました。心からお礼申し上げます。

### 参考資料

- [1] 秋光 淳生, 川合 慧, 情報ネットワークとセキュリティ, 2010, 放送大学教育振興会
- [2] 江口 朴郎 編集, 世界の歴史 14 – 第一次大戦後の世界, 1962, 中央公論社
- [3] 江崎 道朗, コミンテルンとルーズヴェルトの時限爆弾, 2012, 展転社
- [4] S.C. コウチーニョ, 林 彬 訳, 暗号の数学的基礎-数論と RSA 暗号入門, 2001, シュプリングー・フェアラーク
- [5] Elgamal, T., “A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, IEEE Transactions on Information Theory IT-31 N.4 (1985), 469-472
- [6] シムソン・ガーフィンケル 著, 山本和彦 監訳, PGP 暗号メールと電子署名, 1997, オライリー・ジャパン
- [7] 原勝洋, 北村新三, 暗号に敗れた日本, 2014, PHP 研究所
- [8] Hilton, P., “Cryptanalysis in World War II-and Mathematics Education, The Mathematics Teacher 77 No.7 (1984), 548-552
- [9] 広田厚司, エニグマ暗号機, 2004, 光人社
- [10] 本田 元, 新技術で決済が変わる! 図解カードビジネスのしくみ, 2013, 中央経済社
- [11] 星野 力, 誰がどうやってコンピュータを創ったか?, 1995, 共立出版
- [12] デーヴィッド・カーン 著, 秦 郁彦, 関根 英雄 訳, 暗号戦争 – 日本暗号はいかに解読されたか, 1967, 早川書房
- [13] ルドルフ・キッペンハーゲン, 赤根 洋子 訳, 暗号攻防史, 2001, 文藝春秋
- [14] N. コブリッツ, 櫻井 幸一 訳, 数論アルゴリズムと楕円暗号理論入門, 1997, シュプリングー・フェアラーク
- [15] 久米原 栄, 三上 信男, ネットワーク 超入門講座 セキュリティ編, 2010, ソフトバンククリエイティブ
- [16] 倉山満, 大間違いの太平洋戦争, 2014, KK ベストセラーズ
- [17] ジョン・マコーミック 著, 長尾 高弘 訳, 世界でもっとも強力な 9 のアルゴリズム, 2012, 日経 BP 社
- [18] スコット・マッカートニー 著, 日暮 雅通 訳, エニアック – 世界最初のコンピュータ開発秘話, 2001, パーソナルメディア
- [19] 宮地 充子, 楕円曲線暗号, 数理解析研究所講究録 1098 (1999), 138-146
- [20] 能澤 徹, コンピュータの発明, 2003, テクノレビュー
- [21] 大井 孝, 欧州の国際関係 1919-1946 – フランス外交の視角から, 2008, たちばな出版
- [22] 大駒 誠一, コンピュータ開発史-歴史の誤りをたずねる「最初の計算機」をたずねる旅-, 2005, 共立出版
- [23] 太田 和夫, 國廣 昇, ほんとうに安全? 現代の暗号, 2005, 岩波書店
- [24] チャールズ・ペゾルド 著, 井田 哲雄, 鈴木 大郎, 奥居 哲, 浜名 誠, 山田 俊行 訳, チューリングを読む, 2012, 日経 BP 社
- [25] Rivest, R.L., Shamir, A., Adleman, L.M., “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Comm. of ACM 21,2 (1978), 120-126
- [26] ジョセフ・H・シルバーマン, 鈴木治郎 訳, はじめての数論 原著第 3 版, 2007, ピアソン・エデュケーション
- [27] サイモン・シン, 青木薫 訳, 暗号 ロゼッタストーンから量子暗号まで, 2001, 新潮社
- [28] 高田 伸彦, 南 俊博, 情報ネットワーク教科書, 2014, 東京電機大学出版局
- [29] 高岡 詠子, チューリングの計算理論入門, 2014, 講談社
- [30] アンドリュー・S・タネンバウム, デイヴッド・J・ウエザロー 著, 水野忠則, 相田 仁, 東野輝夫, 太田 賢, 西垣正勝, 渡辺尚 訳, コンピュータネットワーク-第 5 版, 2013, 日経 BP マーケティング
- [31] 立石 泰則, フェリカの真実 – ソニーが技術開発に成功し、ビジネスで失敗した理由, 2010, 草思社
- [32] ハミルトン・フィッシュ 著, 渡辺 惣樹 訳, ルーズベルトの開戦責任: 大統領が最も恐れた男の証言, 2014, 草思社
- [33] チャールズ・A・ピーアド 著, 開米潤, 阿部直哉, 丸茂恭子 訳, ルーズベルトの責任 (日米戦争はなぜ始まったか) 上・下, 2011, 2012, 藤原書店
- [34] ジョージ・モーゲンスター 著, 渡邊明 訳, 真珠湾-日米開戦の真相とルーズベルトの責任, 1999, 錦正社
- [35] 若山正人 編, 技術に生きる現代数学, 2008, 岩波書店
- [36] H.O. ヤードレー 著, 近現代史編纂会 編, 平塚 枢緒 訳, ブラック・チェンバー – 米国はいかにして外交暗号を盗んだか-, 1999, 荒地出版社
- [37] 結城 浩, 暗号技術入門-秘密の国のアリス, 2003, ソフトバンクパブリッシング