数学入門公開講座

昭和58年７月26日（火）から８月４日（木）まで

日 / 時間	7月 26日(火)	27日(水)	28日(木)	29日(金)	30日(土)	31日(日)	8月 1日(月)	2日(火)	3日(水)	4日(木)
13:15～14:45	一松	一松	一松	一松	休講		中西	中西	中西	中西
14:45～15:00	休憩						休憩			
15:00～16:30	宇敷	宇敷	宇敷	宇敷			松浦	松浦	松浦	松浦

主催　京都大学数理解析研究所

講師及び内容

1. 暗号の数理 (6時間)

京都大学数理解析研究所教授 一松 信

近年電気通信や計算機内部の情報保護のために、暗号の活用が情報科学の重要な課題となり、数学的に興味ある問題が続出している。こうした動向をふまえて、特に近年急激に発展してきた数学上の話題を中心に、計算機犯罪に対する対策を含めて論ずる。

2. カオスとフラクタル (6時間)

京都大学教養部助教授 宇敷重広

簡単な数学的アルゴリズムから生成されるにもかかわらず、極めて複雑な構造をもつものとして、カオスやフラクタルがある。その生成のメカニズムとダイナミクスについて実例をあげて解説する。

3. グラフの理論と素粒子 (6時間)

京都大学数理解析研究所助教授 中西 襄

線とその端点から構成される図形で、線の接合関係以外の相違を無視して得られる概念をグラフという。この単純な対象を研究するのがグラフの理論であるが、その体系は深遠かつ巧妙であり、応用範囲は極めて広い。ここでは、素粒子物理学において現れるファインマン・グラフへの応用について、面白そうな話題をいくつかとりあげてみたい。

4. 接図形と無限小解析 (6時間)

京都大学数理解析研究所教授 松浦重武

勝手な平面図形、空間図形（一般には、n次元空間の部分集合）の任意の点における接図形の定義を述べる。図形の滑らかさについては、何も仮定しない。

図形が関数のグラフになっているときには、それらがどのようなものになるかは、無限小解析の問題となる。

時間割

日 / 時間	7月 26日(火)	27日(水)	28日(木)	29日(金)	30日(土)	31日(日)	8月 1日(月)	2日(火)	3日(水)	4日(木)
13:15～14:45	一松	一松	一松	一松	休講		中西	中西	中西	中西
14:45～15:00	休憩						休憩			
15:00～16:30	宇敷	宇敷	宇敷	宇敷			松浦	松浦	松浦	松浦

昭和58年度(第7回)数学入門公開講座テキスト(京都大学数理解析研究所, 昭和58年7月26日～7月29日開催)

1. 暗号の数理 (6時間)

京都大学数理解析研究所教授 一松 信

1983, JULY 26, 27, 28, 29 13:15-14:45

1983年7月

数理解析研究所　数学入門講座

暗号の数理

一松　信

1. 暗号の必要性　2
2. 暗号の種類と小史　8
3. 非対称暗号系　16
4. DES　26
付録 数学的基礎事項　35

1. 暗号の必要性

1.1 暗号の定義

広義には暗号という語は, 秘密通信と同意に使われるが, 狭義には, 次のような定義がふさわしい.

暗号とは, 関係者以外に通信内容が知られないようにするため, 通信文をその"構成単位"の列とみて, 一定の算法による変換を施し, その算法を知っている正規の受信者は, その内容を復元して読むことができるが, それを知らない第三者には意味がとれないようにする手法 (cryptography), あるいはそれによって作られた通信文 (cipher text) をいう.

ここで"構成単位"とは, 必ずしも通例の文字 (アルファベットなど) と限らず, 符号 (code) 化した数字, モールス符号の・と—とスペース, テレタイプコードの0, 1のビット, などをも含める. その具体例は, 2章以降に述べる.

上記の変換には, 当事者のみのとりきめに基く秘密のパラメタが含まれるのが普通で, これを鍵 (key) という. 現在の暗号は, 原則として, その秘密保持をもっぱら鍵にたよっている.

送信者が通信文 (平文 ; plane text) を暗号文に変換することを暗号化 (encipher) といい, 正規の受信者がそれを平文に戻すのを復号 (decipher) という. これに対して, 正規

の受信者でない第三者が、いろいろな手法で原文を推定する作業を解読 (to break) という。これらの用語は混用される場合も多いが、この講義では、このように使い分ける。

1.2 暗号の必要性

昔から仲間うちの秘密を独占したいという欲望があるらしく、4000年以上前のエジプトのガラス玉製造法の文書は、暗号で書かれているという。その後軍事用の通信に暗号が使われてきたことは、いうまでもない。

近年、通信が文書を物品として送達することから、次第に電気通信によって、情報のみを送るようになってきた——電信、電話、ラジオ、テレビ、テレックス、ファクシミリ---。電気通信は原則として、"盗聴"可能であり、その内容を秘密に保つには、暗号化しかない。さらに計算機に多量のデータが蓄積されるにいたって、例えば個人に関する情報を、みだりに関係者以外に見せないようにすることが、重要になってきた。それはまた改竄(かいざん)防止のためにも、必要である。

暗号は本来通信の秘密を保つ手法であった。しかしこれをうまく活用すると、情報圧縮と署名の操作が可能になるので、その応用も重要である。(署名の具体的な技法は、3章で解説する。) また部分的な誤り検出・訂正可能性もある。

このような目的で使われる暗号は，秘密保持性もさることながら，次のような性質が要請される．

(i) 自動化可能であること．鍵の入力など以外に，変換はすべて自動的に行われるべきである．

(ii) 通信文がなるべく短くなること

(iii) 能率がよいこと．できれば on-line で処理できること．

(iv) 装置そのものを改造せずに，鍵の変更など入力データの変更のみで，秘密保持ができること．

このような要請には，歴史的なるくの暗号は十分でなく，新しい工夫が必要になる．((2章参照))

他方暗号解読技術は，簡単な場合には，一種の知的レクリエーション(?)であるし，古代文字の解読に活用された例も，少数ながら存在する．ただし暗号解読技術は，もとの文章が何語で書かれているかが既知でないと，ほとんど無力な点が，古代文字解読への応用にあたって，大きな制約である．

単に秘密保持という意味だけならば，後述の"一回限り使い捨て無限乱数"による暗号は，原理的に解読不能という究極の強度をもっている．しかしこれには運用上の難点が多い．近年の(実用上の)暗号は"原理的には解読可能だが，その操作が事実上実行不可能なほどの費用と時間を要する"という方向をねらっているようである．

1.3. 計算機犯罪

計算機犯罪 (computer crime) とは新語であるが、広義には計算機に対して、または計算機を使う犯罪すべてを意味する。しかし近年は、この意味をもっと限定し、例えば計算機自体を盗んだり、物理的に破壊したりする犯罪を除外する傾向にある。

計算機犯罪の完全な分類は困難だが、大ざっぱにいって、次のように分けられそうである。

(i) 金銭または物品の横領 —— 手口はさまざま。例えばキャッシュカード偽造、注文先の変造、利子切捨て分を秘密口座に振込み、大金を一時借金(期限内に返済予定)…

(ii) 情報そのものを盗む、または改竄する。—— 手口はほぼきまっているが動機は多様: 産業スパイ、特定個人の中傷、不利記録の消去…

(iii) 混乱を起す目的で、偽情報の入力、プログラムの改竄

(iv) 計算機の不正使用、タクシーの"エントツ"流の手口

(v) ソフト的な器物破損・妨害

(vi) カンニング的利用

(vii) ミスによる公害(悪意のない犯罪)、余りに下手すぎて、他人に迷惑をかけるプログラム

…… 等々。

この種の議論を余り詳しくすると、"犯罪教授"になるので、これ以上述べない。アメリカのウェルズ・ファーゴ銀行の事件な

ど，2年以上にわたって，2100万ドルも盗まれたのがわからなかった（捕ったのは，犯人の操作ミスから）というのは，計算機犯罪という以上に，銀行の管理自体の問題と思う。

計算機犯罪は，発覚し難く，調査に専門知識がいり，既成の法律で処罰し難く，社会的影響が大きい，など，新手の犯罪として，大きな問題になりつつある。

計算機犯罪を防ぐ最大のきめ手は，人間関係の改善らしいが，精神論だけでは完全な解決にはならない。暗号も一つの手段にすぎず，けっしてそれで完全でも万能でもないが，ある場合には，防御の有力な手法になる。このような面からも，暗号が情報保護の一手法として注目されている。

1.4 主鍵の管理

以下に述べるように，暗号には数学の応用される場面が多いが，その他にも関連しておもしろい問題が数多い。一例として主鍵（master key）の管理問題をあげよう。

主鍵は一連の数字である（それを変化して，いくつかの鍵を生成する）。これを盗まれたら，全部の情報を盗まれてしまう。他方それを紛失したら，すべての情報が読めなくなってしまう。主鍵のコピーを作れば，盗まれるおそれを生ずるし，コピーしないでおけば，紛失の危険性を生ずる。

では主鍵を, 例えば7つの部分に分割し, そのうちの3つまでわかっても, 一意的に復元できないが, 4つがわかればどの4部分でも正しく復元できるようにできないものか? それができれば, 1つや2つ盗まれても, すぐに鍵の変更ができるし, 1つや2つ紛失しても復元可能になる.

そのような方法として, いくつかの案が提唱されている. 実用になりそうなのは, 百五減算方式 (いくつかの素数 $p_1, \cdots, p_n$ で割った剰余 $r_1, \cdots, r_n$ を記録しておく方法) と, Lagrange の補間多項式を利用する方法である. ――― (詳細は当日余裕があったら述べる) 聽講の方々も, よい案を考えついたら, お教えいただきたい.

この例は, 正に何がどういう所で役に立つかわからない, という話の典型例かもしれない. また数学の思いがけない応用例として, 注意すべきかもしれない.

2. 暗号の種類と小史

2.1 暗号の種類

- 換字式 (substitution) -- 主流. 例「黄金虫」
- 転置式 (transposition) - 補助手段, 併用 例「謎の暗号」
- 挿入式 (insertion) -- 現在では補助手段のみ 例「グロリア・スコット号」

- 逐字方式 -- 換字に限る. かつての機械暗号
- 小ブロック式 -- 現在の主流
- 大ブロック式 -- 転置暗号など

境界はあいまい. 電子装置が"記憶"をもったために, 実用になった.

- 対称暗号(慣用暗号) -- 暗号化と復号の鍵が同一
- 非対称暗号(公衆鍵暗号) -- 暗号化と復号の鍵が別. 次章で解説予定

換字式の分類: (併用方式を除く)

- 単文字換字 例 Caesar 暗号, 「黄金虫」
- 二文字換字 例 Playfair 暗号
- 周期的多表式 例 Vigenère 暗号
- 流れ暗号 -- 平文そのものをずらせて鍵とする.
- 機械(擬似)乱数 例 ENIGMA
- 完全無限乱数 -- ホットラインなど特別重要な秘密通信

2.2 単換字暗号

Caesar 暗号 -- アルファベット順に一定間隔だけずらす. Aを0, Bを1, --, Zを25 とした26進法の加算とみ

て，Aのうった先を鍵字とする．ただし

$M+K=C$（通常の方式），M平文，K鍵，C暗文に対して，$K-M=C$ とする方式を，特にBeaufort式ということがある．この方式では復号も $K-C=M$ と同じ方式であって，機械化が容易という利点がある．

「黄金虫」の暗号（別掲）は，単文字換字（各文字を別の文字または記号に換える）の典型例である．この解読には，各記号の頻度分布やよく現れる文字列を利用する．「黄金虫」の暗号は，まず英語であることを確認した上で，最も多い8をeと，7回も現れる；48をtheと推定し，それらを復元した上で，順次文字を推定してゆく．2文字の頻度分布などを補助に使えば，さらに容易に解ける．

情報理論によれば，英語の単文字換字暗号は，原理的には20字ほどの文があれば解読可能である．文章が分ち書きされていれば，なお容易である．現在では，単文字換字の状態にまで還元された暗号は，解読されたも同然とされている．

この解読を防ぐため，eやtなどよく現れる文字には，複数個のコードを割りあてる"同音式"が工夫されたが，機械化が困難なためもあって，現在では使われていない．歴史的には，1401年に同音式暗号が使われた記録がある．

換字暗号の例 (演習問題)

```
53‡‡†305))6*;4826)4‡.)4‡);806*;48†8¶
FHXXGHYFVVICKNOWIVNXMVNXVKOYICKNOGOQ

60))85;;]8*;:‡*8†83(88)5*†;46(;88*96
IYVVOFKKDOCKRXCOGOHAOOVFCGKNIAKOOCJI

*?;8)*‡(;485);5*†2:*‡(;4956*2(5*-4)8
CZKOVCXAKNOFVKFCGWRCXAKNJFICWAFCPNVO

¶8*;4069285);)6†8)4‡‡;1(‡9;48081;8:8
QOCKNYIJWOFVKVIGOVNXXKUAXJKNOYOUKORO

‡1;48†85;4)485†528806*81(‡9;48;(88;4
XUKNOGOFKNVNOFGFWOOYICOUAXJKNOKAOOKN

(‡?34;48)4‡;161;:188;‡?;
AXZHNKNOVNXKUIUKRUOOKXZK
```

図2・1 ポーの『黄金虫』の中の暗号文。各行の上段はポーの暗号文，下段は英字に置き換えた暗号文。なお，同じ符号を同じ文字に置き換えるという原則さえ変えなければ，この二つの暗号文相互のあいだに質的な違いはまったく起こらない(「数理科学」1975年12月号の長田順行の記事による)。

南無阿弥陀仏、南無阿弥陀仏、阿弥陀仏、
南無阿弥陀仏、阿弥陀仏、無陀、陀、
南無阿弥陀仏、南無仏、陀、無阿弥陀、
無陀、南仏、南陀、無弥、
無阿弥陀仏、南無阿陀、阿弥、
無阿弥、南陀仏、南阿弥陀、阿陀、
南弥、南無弥仏、無阿弥陀、
南無弥陀、南弥、南無弥仏、
南無弥陀、南無陀、南無阿、阿陀仏、
無阿弥、南阿、南阿仏、陀、南阿陀、
南無、無弥仏、南弥仏、阿弥、
南無陀仏、阿弥陀、無陀、
南無阿弥陀、阿陀仏、

図2・2 江戸川乱歩の『二銭銅貨』に出てくる暗号文

カナ文字を点字で表現し，6個の穴を南無阿弥陀仏の1文字に対応させている。

図2.3

Sample ciphertext. 多表式暗号の例 ——は長文反復

ZHYME ZVELK OJUBW CEYIN CUSML RAVSR YARNH CEARI UJPGP VARDU
QZCGR NNCAW JALUH GJPJR YGEGQ FULUS QFFPV EYEDQ GOLKA LVOSJ
TFRTR YEJZS RVNCI HYJNM ZDCRO DKHCR MMLNR FFLFN QGOLK ALVOS
JWMIK QKUBP SAYOJ RRQYI NRNYC YQZSY EDNCA LEILX RCHUG IEBKO
YTHGV VCKHC JEQGO LKALV OSJED WEAKS GJHYC LLFTY IGSVT FVPMZ
NRZOL CYUZS FKOQR YRYAR ZFGKI QKRSV IRCEY USKVT MKHCR MYQIL
XRCRL GQARZ OLKHY KSNFN RRNCZ TWUOC JNMKC MDEZP IRJEJ W

2.3 多表式暗号

二文字ずつ換字する暗号例として, Playfair (この名は異論がある) が19世紀に使われたが, 現在では余り使われないので解説を略す. ——作り方の規則性から, 一たん部分的に解読されると, いもづる式に全部が解読される危険性が大きいせいである.

そのために考えられたのが多表式——文字ごとに鍵字を変えて別々の変換をする方式——である. 最も簡単なのは, Vigenère (この名も異論が多い) 暗号とよばれる方式で, 鍵字を定め, それを反復して使い, 平文の各文字を鍵字に応じて, Caesar 暗号式に変えてゆくものである:

例.

H O W A R E Y O U T O D A Y	← 平文
E N G L A N D E N G L A N D	← 鍵字
L B C L R R B S H Z Z D N B	← 暗号文

上記の例でもわかるように, 暗号文中の同一文字が, 平文で別の文字に対応するから, 単に Aがいくつ, Bがいくつ, ... と統計をとっても意味がない.

かつては, この種の暗号は, 鍵がわからなければ解読不能と考えられていたが, 現在では, 長文の暗号文があれば解読できる. その基礎は, W. Kasiski (1863年; プロシアの将校) による周期推定法である. もしも鍵字の長さnの倍数だけ離

れて，同一の文字列（必ずしも一つの単語である必要はない）があれば，それは同じ暗号文中の文字列になる．もちろん偶然の一致もありうるが，統計的に5文字以上の反復，または4文字の4回以上の反復は，同一の平文の文字列とみなしてほぼ誤りのないことが確かめられている．したがってそのような長文字列の反復がみつかれば，周期nが推定できる．周期nがわかれば，n文字ごとにとった列は単文字換字になっているから，その頻度分布から解読できる．ときには鍵字の文字列が推定でき，容易に解読できることもある．

p.10の下段は，D.E.R. Denning 女史の著書(末尾)にある例である．下線を引いた11文字の3回反復から，この周期が51と72の最大公約数3であることが，ただちにわかる．3文字ごとにとった文字の頻度分布から，鍵字が容易に推定できる．このあとは聴講者の演習問題としておく．

もし長文字列の反復がみつからないときには，鍵字の長さnの範囲（たとえば5～10）が既知のことが多いので，n文字ごとに区切った各列の文字の分布を調査すると，それからnがいくつであるのが一番もっともらしいかが推定できる．そのような試行錯誤を若干行うと，解読できる場合が多い．現在では，この種の短周期の多表式暗号は，単文字換字暗号より若干多い手数で解読できるのが普通である．

2.4 流れ暗号, 無限乱数など

流れ暗号とは, 最初に鍵字をおき, 以後もとの平文をずらせてそれを鍵字とする方式である.

例　　T H E T R E A S U R E I S B U R I E D　←平文

最初の鍵→ H A M, T H E T R E A S U R E I S B U R　←鍵

A H Q M Y I T J Y R W C J F C J J Y U　←暗号文

この方式は一時アメリカで盛に使われた。しかし平文そのものが鍵であるため, よく現れる文字同志のくせが生じやすく, 一部分が解読されると, いもづる式に全部が解読できるなどの欠点があり, いまでは使われない.

第二次大戦頃に盛に使われた暗号機械の多くは, 比較的簡単な規則から, 長い周期の鍵字列を自動的に作りだし, 暗号化, 復号を自動的に行う装置であった。この種の機械は適切に使えば, かなりの強度をもつ暗号になるが, 何といっても"簡単な規則"でできる鍵字列は, 完全な乱数列にはなりえず, そのくせが解読の手がかりになったようである。この種の研究が, 今日の計算機による"擬似乱数の生成"とつながりがあるのかどうか不明である。ドイツのENIGMA式の暗号機械は, 記憶がないため, 誤りの訂正が不可能な点と, 電子計算機による解読が可能になったため, 現在ではもう使われなくなったが, 方式としては, 優れた着想だったといってよい.

日本の"紫暗号"の解読成功は、もちろんFriedmanの天才に負うが、使用の失敗(鍵の設定不備)もあったように感ずる。

この方式を押し進めると、乱数の鍵字列を無限長とし、一回限りの使い捨てとすれば、鍵字がわからない限り、原理的に解読不能となる。これが今日まで最高の機密を要する通信に使われている方式である。

しかしこの方法では、平文と同じ長さの乱数(鍵字)列が必要である。これを確実に送達し、管理、廃棄などを実行するには、多大の費用と手間を要する。そのため、この方式は真にそれを必要とするホットラインのような場合に限定されている。

もともと乱数表は、実験計画、品質管理、シミュレーション実験用などのための研究道具であるが、いつしか暗号用のほうが有名になり、スパイ道具の一つとでも思っている方が多いようなのが遺憾である。

2.5 転置暗号その他

転置暗号も歴史が古い。19世紀にはかなり広く使われたし、(1回)日本軍も一時採用したことがある。しかし20世紀前半には、(記憶をもった電子装置が普及するまで)機械化が困難なこと、一文字の脱落や重複が混乱を生じやすいこと、それだけ

では解読されやすいこと，などの理由で，ほとんど使われなくなった．もっとも換字と転置という相異なる方式を併用することによって，著るしく強度が高まることが，理論的にも実験的にも知られていた．

近年電子装置の発達により，転置操作（scramble — かきまぜ；という語がよく使われる）が機械化されるに及び，ふたたび換字と併用した転置暗号が脚光をあびてきた．その典型例がDES（第4章）である．*)

挿入式暗号は，Holmesもののグロリア・スコット号にあるのがよい例だが，文章が長くなるのと，それだけでは強度が弱いため，現在ではほとんど使われない．言葉遊びのような隠字詩（acrostico）は一種の挿入式暗号であり，二銖銅貨の暗号（図2.2）にも，隠字句の"どんでん返し"が含まれているが，これらは補助的な利用であろう．ただし，空白や句読点も特別のコードとして平文中にかえたり，字数の合わない空白にQ, X, Zなど意味のない文字を埋めるのは，一種の補助的な挿入式暗号と考えることができる．

*)(付記) 考えようによっては，ビデオテープの録画記録は，巨大な転置暗号ともみられる．

第3章 非対称暗号系

3.1 非対称暗号系の考え方と特長

"非対称暗号系"というのは, 1979年頃につけられたらしいが, この考えは, 1976年 Hellman と Diffie が発表したもので, 若干の誤解があったものの"暗号史の分水嶺"といってよい着想である. それは"計算量の理論"の最も直接的な応用でもあった.

そもそもの考えは, 暗号化の鍵Kと復号の鍵K'とを別にするということである. これにより, 次のような利点を生ずる. (その詳しい意味については, 以下に順次解説する)

1° n人が相互に自由に通信し合う場合, 慣用暗号系では人間の対の数 $n(n-1)/2 \sim n^2/2$ だけの鍵を要するのに対して, 非対称暗号系では, 各人につき2個, 計2n個ですむ. したがってネットワーク状の通信に適する.

2° 暗号化の鍵は公表してよい. このため鍵の配布が(一部で誤解されているように不必要ではないが)容易になる.

3° 方式によっては, "署名"が可能である.

4° その他慣用暗号系では不可能な特色がある.

もっともよいことずくめではない. 以下のような欠点がある.

1° 原理的に解読可能であり, 秘密保持力をもっぱら計算の複雑度にたよっている. このために思いがけない抜け穴が

ありうる.

2° 本質的に非能率的である. 現在最も有用と考えられているRSA系で, 専用のチップを使っても, 1ms/bit程度である. これはDESより千倍も遅く, on-lineでは使えない.

そのため, 現在では, 次のような用途が考えられている.

1° 場所が固定している組織同志の間の, 特別重要な, 正当性の証明をも要する文書の通信用.

2° DESを利用するとき, その鍵の送達などのプロトコール用

3° "電話でジャンケンをする", といった交渉用

最後の"電話でジャンケン"とは, 自分の手を先にいったら相手に裏をかかれる, という場合, いかにして互いに自分の意向を相手に伝えると同時に, それに影響されないままの相手の意向を知る, といった相互情報交換を公正に行うかという課題である. 完全ではないが, 非対称暗号系によって, ある程度それが可能になるという提案がある. さらに進んで"不誠実な相手と交渉して, だまされる確率をなるべく小さくする方法"といった研究もある. この確率を0にはできない. 任意のをより小さくすることはできるが, それにはとてつもなく長い時間を必要とする.

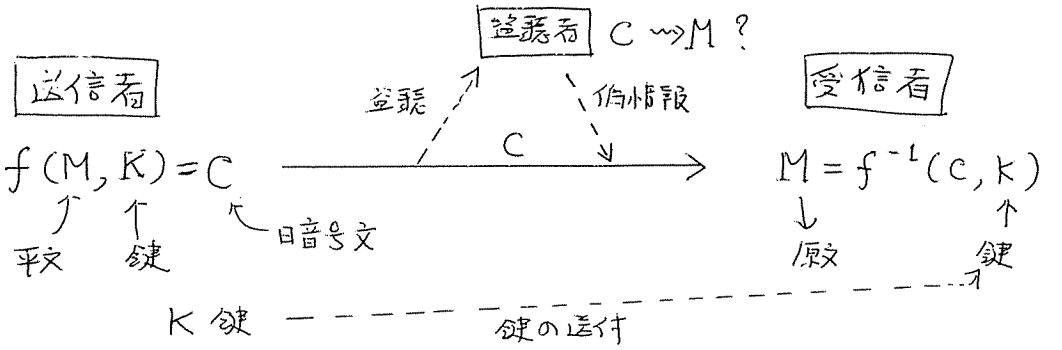


図 3.1 . 慣用暗号系

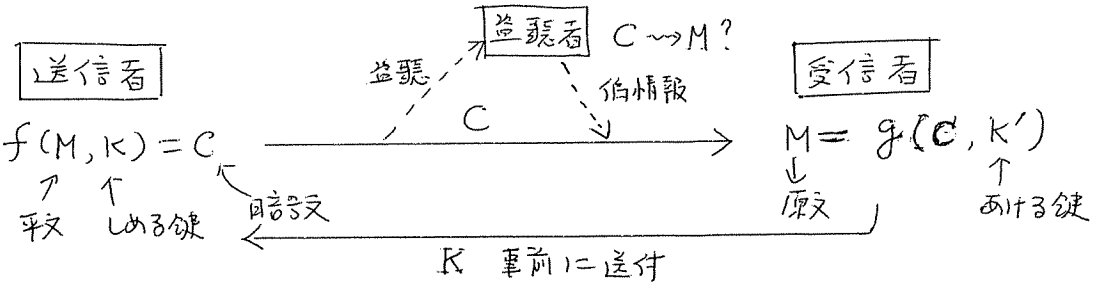


図 3.2 非対称(公開鍵)暗号系

$g(f(M,K),K') \equiv M$
[半可逆性]

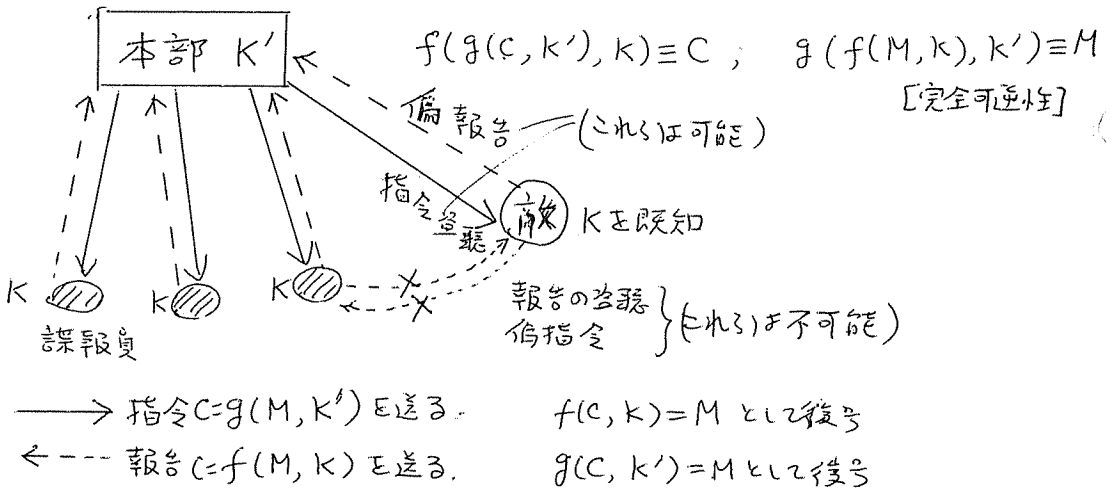


図 3.3 非対称暗号系の効用のモデル

3.2 ディジタル署名

電気通信の最大の難点は，署名（通信文の正当性の証明）が不可能なことである。この難点の解決案を提示したという点でも，非対称暗号系の価値は高い。

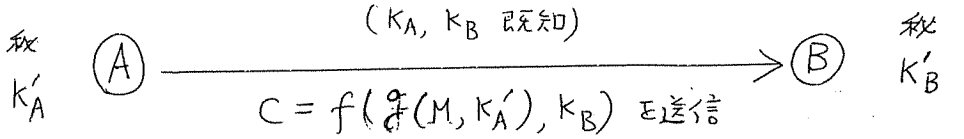
それは次のようにする。このためには，復号用の鍵K'と暗号化の鍵Kとが完全に双可逆である必要がある。

まず秘密の復号用の鍵K'をもっている人が，公開状を発するには，K'による復号化の操作を平文に施す。暗号化(!)の鍵Kをもっている人は，それを暗号化することによって，平文を復元できる。この暗号文はK'がなくては作れない。

完全に双可逆な方式でなくても，秘密保持を断念して，署名のみをすることは可能な場合がある。しかしある場合には，署名をつけると，秘密保持が不可能になることが証明されている。その意味で完全に双可逆なことが望ましい。

通例の署名つき秘密通信は，図3.4のようにして行う。ここで後日第三者（裁判所など）への証明ができることが本質的である。この変形として，図3.5のような方式もある。この（しかも自分の秘密の復号用の鍵を提出せずに）のほうが，場合によっては有利なことがある。

実用上には，すべてをこのように変換するのは大変なので，要約や一部分のみをこうすることも多い。

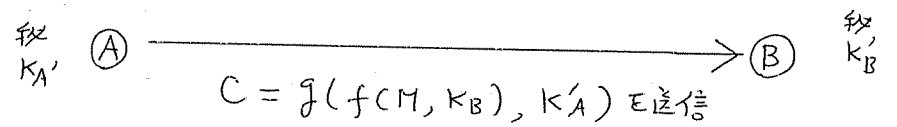


Bの復号：$M = f(g(C, k'_B), k_A)$

第三者への証明：　提出　$X = g(C, k'_B)$, M

検証　$M' = f(X, k_A)$,　$M = M'$?

図3.4　署名の方法



Bの復号：　$M = g(f(C, k_A), k'_B)$

第三者への証明：　提出　C, M

検証　$X = f(M, k_B)$,　$X' = f(C, k_A)$

$X = X'$?

図3.5　署名の他の方法

3.3 Pohling 暗号

上記のように，非対称暗号はいろいろの特長があるが，問題は具体的な算法である．当初 Hellman-Diffie が提案した方式はいずれも（逆算法が容易に可能なので）実用的でなかった。以下に述べる Pohling 暗号もそうである．それにも拘らず，これを述べるのは，後述の RSA が，ある意味でこの方式の"一寸した改良"だからである．

まず平文を適当に区切って，数字にコード化する．— 例えば A は 01，B は 02，…，Z は 26，スペースは 00 など．したがって平文はある大きな素数 p より小さい正の整数 a である．$p-1$ と互いに素な指数 e をとり，a の e 乗：

$$c \equiv a^e \pmod{p}$$

を暗号文とする．— この計算は，付録 2 のように高速にできる。復号するには，適当な d を求めて，暗号文 c の d 乗：

$$a \equiv c^d \pmod{p}$$

をとればよい．具体的には，付録 1 の方法で，$ue+v(p-1)=1$，$u>0$ である u を作ると，Fermat の小定理（付録 3）から

$$a^{ue} = a^{1-v(p-1)} \equiv a \pmod{p}$$

となるので，$d=u$ とればよい．この計算は容易にできるので，これでは暗号にならない．ただ d と e とは完全に可逆で，署名が可能である（もっとも偽造も容易にできる．）

3.4 RSA 暗号

現在非対称暗号系として, 最も注目されているのは, これである. この名は発案者 Rivest, Shamir, Adleman (MIT) の頭字である.

Rivestらは, Pohling 暗号の mod としてとる数を, 2つの大きな素数 p, q の積であるれに改めた. e は $p-1, q-1$ と互いに素 (したがって $\varphi(n)$ と互いに素) にとる. この"一寸した改良"で, 強度は飛躍的に高まった.

復号はやはり適当な d をとって, $a \equiv c^d \pmod n$ とする. Fermat の小定理 (付録3) により, $ue + v\varphi(n) = 1$, $u > 0$ であるものがあり

$$a^{ue} = a^{1-v\varphi(n)} \equiv a \pmod n$$

だから, $d = u$ とすればよい. この計算は $\varphi(n) = (p-1)(q-1)$ を知れば, 付録1の方法で高速にできる.

しかしながら, 大きな (数十桁以上) 整数れが素数であるか否かは高速度に判定できるが, 素数でないことがわかったれの素因数を求めるには, (一般に) 極めて長い時間を要する. この劇的な差が RSA 暗号の秘密である. $\varphi(n)$ を求めるには, れを素因数分解しなければならない. もし直接の方法があったとすると, $\varphi(n) = n - (p+q) + 1$ がわかれば, $p+q$ と $n = pq$ がわかるから, 2次方程式を解いて, p と q と

がわかることになり、事実上素因数分解と同じことになる。したがってeとnとを公表しても、dおよびnの素因数p, qを秘密にしておく限り、それらを求めようとしても、大変な時間がかかって実用にならないというわけである。

なおdとeとは完全に対称だから、これで署名もできる。AからBへ署名つきの文書を送る場合、$n_A < n_B$ならば図3.4の方式を、$n_A > n_B$ならば図3.5の方式をとるとよい。

例.(マイコンなどで確めて下さい)

p=14192599, q=7117387, n=101014219618813

e=127, d= 530258258839 (この計算法は末尾参照)

a=19051212021282 (平文:SELL 1982年 12月02日)

c=72263729504163.

3.5 RSA暗号のアキレス腱

RSA暗号は、現在まで提案された非対称暗号系のうち、最も優れたものであり、すでに一部で実用化されているとか、そのための装置が市販されているとか伝えられている。しかし、思いがけぬ弱点も多く指摘されている。そのうちの主なものをあげる。

1° 高速度の計算機で強引にnの素因数分解ができる可能性がある。$p \fallingdotseq q$ならば、高速度に求める方法もある。その

ため, p, q を100桁位にし, $p-q > q$ であるように選ぶ必要がある.

2° 文の初めや終りなどのきまり文句を暗号化して比較することにより(平文推定法), 一部分読みとくことが可能な場合がある. もっとも普通の暗号では, それから鍵を求めて他の部分の解読が進められるが RSAでは, それができない.

3° 暗号文 c と n とのGCDが1でなければ, n の素因数がわかる. p, q が大ならその様な確率は極めて小さいが, まったくありえない事態ではない.

4° $p-1, q-1$ が(2以外に)小さい約数や共通な約数を多くもっていると, 比較的小さい m で $a^m \equiv 1 \pmod{n}$ になることがある. そのときには, $c, c^2, c^3, \cdots$ を作ってゆくと, その中に a が現れる場合がある. これを防ぐには, (比較的早い時期に) (素数 p, q の選定に工夫がいる.

5° もしも同一の n により, 同一の平文を互いに素な e_1 と e_2 とによって暗号化した2通の文が入手できれば, その平文 a は容易に復元できる. (G.J.Simmons, Cryptologia, vol. 7, no. 2, 1983, pp.180-182). その方法

$$a^{e_1} \equiv c_1 \pmod{n} \qquad a^{e_2} \equiv c_2 \pmod{n}$$

とする. $GCD(e_1, e_2) = 1$ なので, $ue_1 + ve_2 = 1$ である u, v が求められる(付録1). 一方が正, 他方が負である. い

また $u>0$, $v<0$ とする。さて c_1 と n とが互いに素でなければ, 3°によってこの暗号体系そのものが破られる。c_2 と n とが互いに素なら, $c_2 \cdot c_2^{-1} \equiv 1 \pmod{n}$ である c_2^{-1} が求められる(付録1)。そこで mod n について次の式を計算する。

$$c_1^{u} \cdot (c_2^{-1})^{|v|} = a^{e_1 u} \cdot a^{e_2(-(-v))} = a^{e_1 u + e_2 v}$$
$$\equiv a \pmod{n}$$

これによって平文 a が復元できる。

例. 前節の例を e_1 とする。

$e_2 = 149$, $c_2 = 33411612782369$

$127\,u + 149\,v = 1 \rightarrow a=-61,\ b=52.$ (上の c_1 と c_2 とを交換)

$c_1^{-1} = 69196277075228,$

$(c_1^{-1})^{61} \times c_2^{52} = 19051212021282 \pmod{n}$

上記の 4°, 5° はたまたまその文章が読まれたとしても, 秘密の鍵 p, q, d がわかってしまうわけではないから, その暗号が以後使えなくなるわけではない。しかし 5°の例は, $n=pq$ を共用にして, e と d の組のみを変更する方式では, 秘密が保ち難いことを示している。その他実際の運用にあたって, 問題点が数多く指摘されている。

4章 DES

4.1 DESの沿革

DESとは Data Encryption Standard (データ暗号化標準)の略字で, アメリカ国務省が商用暗号の標準として推奨しているものである. 歴史的には, IBMが1973年に発表したLUCIFER(大天使; 魔王; 暁の明星 ── 金星暗号と訳した人がある)暗号を改良したものである. (その概略は図4.2)

それは文字をbit単位列とし, 以下に述べるような複雑かつ巧妙な換字と転置の操作を併用した暗号である. 発信者と受信者が同一の鍵を使う慣用暗号系であるが, 主鍵から16個の鍵の列を生成して, それぞれの変換に使うので, 主鍵が変れば, 暗号文はまったく変化し, 鍵を推定する以外には解読不能といわれている. 操作は複雑だが, 高速度に自動的に変換でき(最高で1μs/bit位), 相当長文の暗号文も短時間で送信できるなど, 利点が多い.

反面鍵が56ビットと短いため, 全数検査で解読可能という批判もある((4.5)). しかし主鍵の送達・管理などに注意すれば, 少くともここ当分の間, 比較的軽度の秘密文書の送達用として, 最も適した暗号系と思われる. 安全を期すためには, 相異なる鍵による二重暗号化も提唱されている.

4.2. DESの算法(1) 全体の流れ図

DESの流れ図は，図4.1のとおりである．

1° 全体を64ビットずつにとる．但し8ビットは奇偶検査用なので，実質56ビット（7ビットコードで8文字分）である．主鍵56ビットを用意する．

1° まず図4.5(a)のような初期転置（IP）を行う．これは図4.2のような配線盤でできる．

2° 64ビットを32ビットずつの L_i, R_i（左と右）に分け，$L_{i+1}=R_i$ とする．他方 R_i と鍵 K_{i+1} とから，後述の f により $f(R_i, K_{i+1})$ を作り，これと L_i とのexclusive or（1と0は1；0と0，1と1は0）をとって R_{i+1} とする．

3° 2°の操作を，鍵 K_i を変更しつつ，16回反復する．

4° 最後の段階では，左右を交換せず（或は一度上のようにしてから左右を入れかえ），IPの逆置換（図4.5(b)）を施して出力とする．

復号はこれを逆に施すのであるが，結果的に，鍵を逆順に $K_{16}, K_{15}, \cdots, K_1$ の順に使うとして，他は上記と同じ操作でよい．（4°のように最後に交換しないのは，このためである．）後述のように，f の操作は1対1でない（$R_i \to f(R_i, K_{i+1})$ は1対1とは限らない）が，復号には支障を与えない点に注意する．

DESの暗号化の流れ図

FIGURE 41 DES enciphering algorithm.

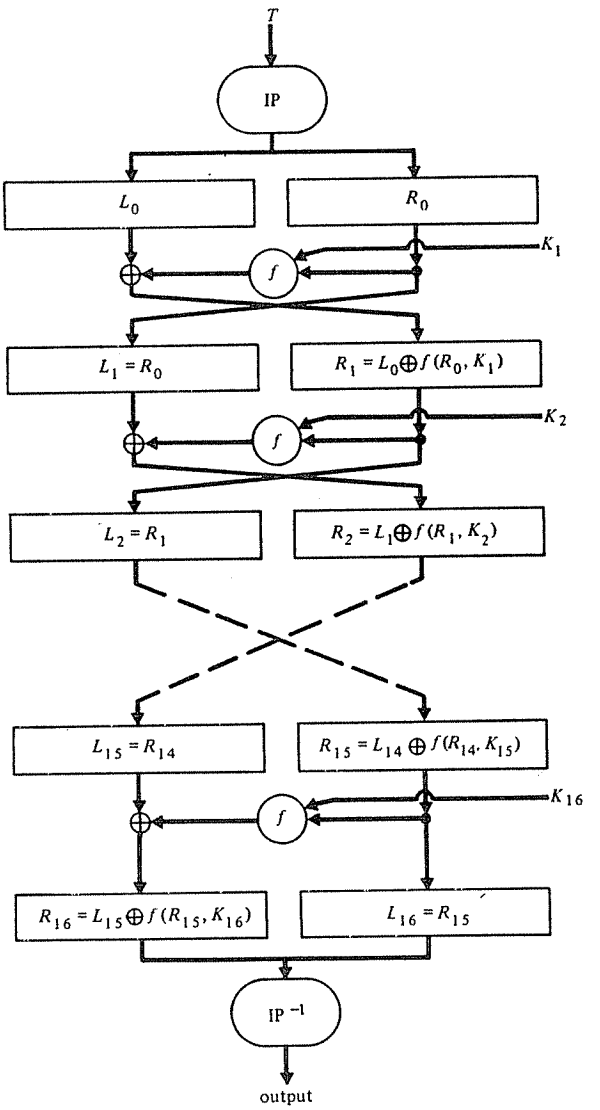


図2) 転置・換字の流れ図

FIGURE 1. Substitution-permutation cipher.

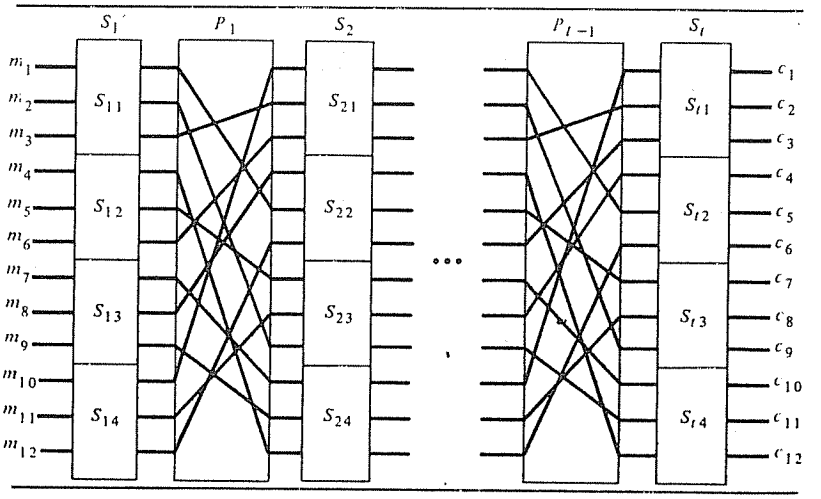


図4.3 — Selection functions (S-boxes). S箱の算法

Row	Column 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

4.3 DESの算法(2) fの計算法

fの計算は次のように行う。以下R, Kの添字を省略する。

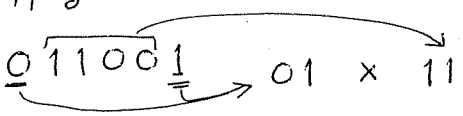
1° 鍵は次々に主鍵から作りだされる48ビットの列である。まずR(32ビット)を図4.6のように一部を重ねて拡大し, 48ビットとする(E).

2° EとKとのexclusive orをとり, それを6ビットずつ8個の$S_1, \cdots, S_8$に分ける。

3° S_iをS-boxとよばれる転置にかけて, 各4ビットの出力とし, 合せて32ビットにする。(攪乱)

4° この32ビットに図4.7のような転置操作を施して, fとする。

S-boxは図4.3のとおりである。この表は, 次のように読む。S_iの入力6ビットを, 両端の2ビットと中央の4ビットに分ける



それらをそれぞれ二進法で0～3, 0～15の数と読み, 図4.3のS_iの対応する左側の行番号と列番号の交点の数値を求める(上の例では1と12で, 答は9 — 1001である)。それを二進4ビットに直したものがS-boxの出力である。

この表から, 1ビット違っても, 出力がまったく変ることがわかる。なおS-boxの作成算法は秘密にされている。

DESの関数fの計算法

FIGURE 4.4 Calculation of $f(R_{i-1}, K_i)$.

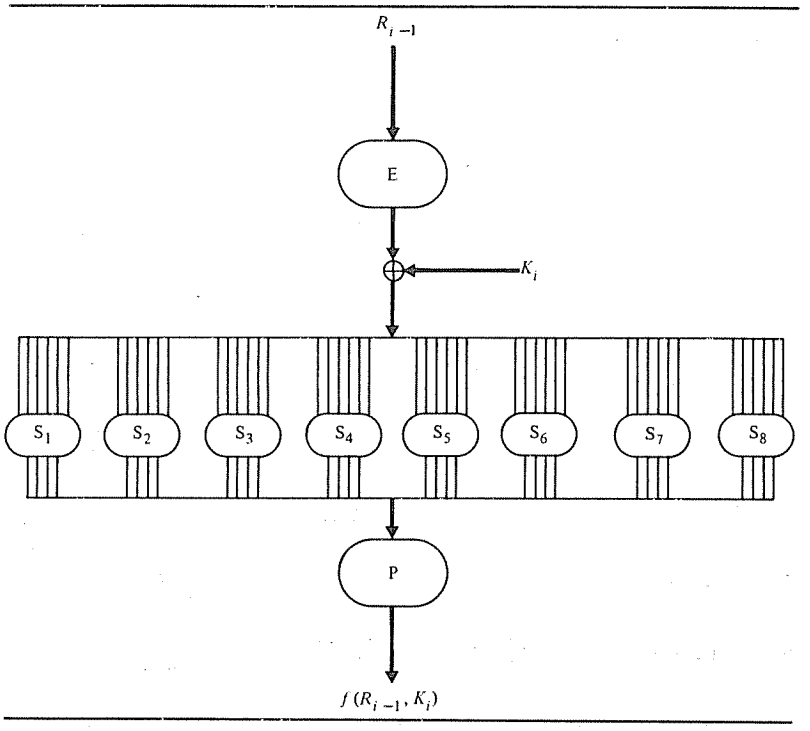


図4.5 (a) Initial permutation IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

図4.5 (b) Final permutation IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

図4.6 Bit-selection Table E.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

拡張算法

図4.7 Permutation P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

転置

§4. DESの算法(3) 鍵の生成

DESの逐字の鍵$K_1, \cdots, K_{16}$は、次のようにして56ビットの主鍵Kから作りだされる(図4.8)

まず56ビットのKを図4.9のPC-1によって転置する。次にそれを28ビットずつのC_0, D_0に分け、次々に図4.11のような数だけ(1または2)左へ巡回シフトをする。16回目には28ビット一巡してもとに戻る。そのうち図4.10にあるような48ビット分を、その順序にとりだして(置換PC-2)、それをK_1とする。以下これを反復する。

復号のときには逆に図4.11の数を下から右に巡回シフトして、PC-2を施せば、K_iがK_{16}から逆順に作りだされる。

もしも最初のマスターキーKがすべて0とかすべて1ならば、K_iはすべて0とかすべて1になる。しかしまずこのような鍵は使われないだろう。またたとえK_iがすべて0とも(テスト用は別として)ても、前記の算法に従うと、平文はまったく変って暗号文になり、容易に解読できないだろう。

ただ鍵の作りだし方が、比較的単純な算法なので、もしも中間のK_iの一つがわかれば、他の鍵の推定が比較的容易に可能になるかもしれない。

図4.8 鍵の生成法

Key schedule calculation.

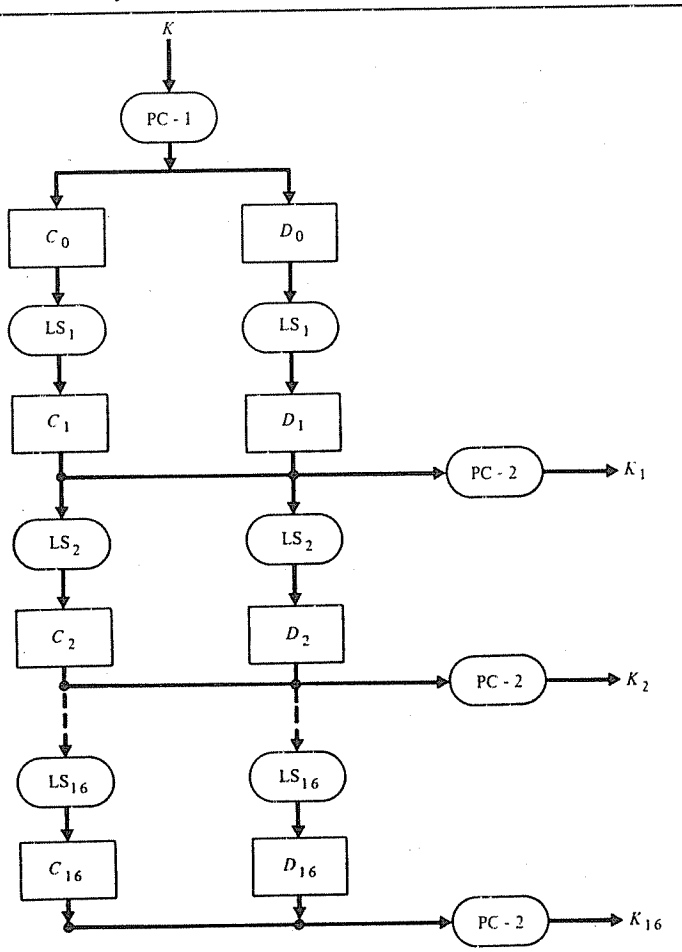


図4.9

Key permutation PC-1.

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

図4.10

Key permutation PC-2.

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

図4.11

Key schedule of left shifts LS.

Iteration i	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

4.5 DESの解読と批判

これまでの実験によって、DESの解読は、たとえ平文と暗号文を入手しても、鍵Kを定めることは困難であり、事実上鍵がわからない限り、解読不可能といわれている。

しかし一方鍵が56ビットしかないため、全数検査で解読可能という意見もあり、いろいろのかんぐりもされている。もっとも"百万個専用チップを並べる"という構想は、円滑な運用面に難があり、その人の評価が甘すぎるという再批判もある。そのため鍵を112ビットにするか、せめて別の鍵で再度変換するとよいという提案もある。

DESの内容は一応公表されているが、置換IP,及びS-boxの表を変更した機械を作ることによって、解読が困難になるかもしれない。もっともそういう点を変更するのは、秘密保持を専ら鍵にたよるという趣旨には反することになる。

いずれにせよ、DESは現在の電子装置の発達によって誕生した暗号機械の粋であり、当分の間標準的な装置として広く使われるであろう。能率の点はともかく、上記の算法に基き、マイコンなどでDESを実現するプログラムを作るのは、プログラミングの絶好の演習課題と思う。

付録　数学的基礎事項

1. 互除法と $um+vn=d$ の計算

$n_1=m$, $n_2=n$ とし以下互除法

$$n_{i-1}=q_i \times n_i + n_{i+1} \qquad (n_{i+1}<n_i)$$

を反復すれば, 有限回で $n_{l+1}=0$ となり, $n_l=G.C.D(m,n)=d$ である. このとき $u_i m+v_i n=n_i$ $(i=1,2,\cdots,d)$ である列 u_i, v_i は, 次の算法でえられる.

初期条件　$u_1=1$, $v_1=0$, $u_2=0$, $v_2=1$

反復　$u_{i+1}=u_{i-1}-q_i \times u_i$, $v_{i+1}=v_{i-1}-q_i \times v_i$

このとき, $u_l m+v_l n=d$ である. ここで $m \geqq n$ である必要はないことに注意する. また q_i は使い捨てでよく, u_i, v_i は2段階前まで保存すればよい. したがって次のプログラムでよい. (次ページにFORTRANによるプログラムを示す)

入力　m,n;

初期値　u0←1; v0←0; u1←0; v1←1;

repeat [q←m ÷ n (整商) 余り r;

w←u0 - q×u1 ; u0←u1; u1←w;

w←v0 - q×v1 ; v0←v1; v1←w;

m←n; n←r]; until r=0;

出力　n (GCD), u0 , v0

注. GCD(m,n)=1 が既知なら, r=1 で止めて, u1, v1 を出力すれば1段階早くなる.

(注 入出力の4は端末機を示す)

```
00100    C        GCD AND COEF. BY EUCLID ALGORITHM
00200    1        READ (4,100) IP,IQ
00300    100      FORMAT (2I10)
00400             IF (IP .LE. 0) STOP
00500             M=IP
00600             N=IQ
00700             J0=1
00800             J1=0
00900             K0=0
01000             K1=1
01100             IC=0
01200    10       K=M/N
01300             L=J0-J1*K
01400             J0=J1
01500             J1=L
01600             L=K0-K1*K
01700             K0=K1
01800             K1=L
01900             L=M-N*K
02000             M=N
02100             N=L
02200             IC=IC+1
02300             IF (L .NE. 0) GO TO 10
02400             WRITE (4,200) IP,IQ,M,J0,K0,IC
02500    200      FORMAT (1H0,2I10,6H  GCD=,I6/1X,2I10,8X,I4)
02600             GO TO 1
02700             END
```

応用例　$GCD(m,n)=1$ のとき　$u \equiv m^{-1} \pmod{n}$.
もし $u<0$ なら, $u+n$ を作ればよい.

Lamé の定理　上記の互除法は, m,n の小さい方の十進桁数を L とすると, 最大限 $5L$ 回で完了する. (極めて能率的!)

2. 累乗の計算

$s=a^m \pmod{n}$ の計算は, m を二進法で

$$m=m_\ell m_{\ell-1} \cdots m_1 m_0 \quad (m_i \text{は} 0 \text{か} 1)$$

と表示すれば, $a \to a^2 \to a^4 \to a^8 \to \cdots$ と作って, $m_i=1$ である桁に対応する値だけを掛けてゆけば求められる. この際 a^{2^q} を完全に作らずとも, すべて $\pmod{n}$ で計算してよい. したがって　$a \times b \pmod{n}$ を計算するサブルーチン　$prod(a,b,n)$ を用意しておけば, 次のプログラムで計算できる.

入力　m=$m_\ell m_{\ell-1} \cdots m_1 m_0$;　a (保存)

```
        s ← 1; b ← a;
for k ← 0(1)ℓ  do
 [ if  m_k = 1  then s ← prod (s,b,n);
        b ← prod (b,b,n)] ;
```

出力 s

prod(a,b,n) は $a \times b$ を n で割って余りをとればよい

が, もしも $n=2^q-k$ で, k が比較的小さいならば, $a\times b$ を二進法で求めて, $c\times 2^q+d$ と分割し(下から q 桁目でわって, 上が c, 下が d) $c\times k+d$ を作ればよい. これがふたたびあふれる可能性があるが, 再度同じ操作を施せば, 三度目には必ず 2^q より小さくなる. n より小さく正規化するには, n と比較して, 大きければ n をひく必要があるが, この操作は最後に行うので十分である. この方法は, 特に $k=1$ (Mersenne 素数) の場合, 極めて有用である.

注 上記の二進分解による算法は, 必ずしも a^m をうるための乗算回数最小の算法ではない. a^m を求めるために必要な最小乗算回数 $f(m)$ はわかっていないし, 計算量の理論のとてつもなく難しい問題の一つである. ただ上記の算法による $2\lceil \log_2 m\rceil$ 回は, 最小回数ではないが, 最小回数の 1.XX 倍程度であり, 簡単で一般性があり, 高能率であるために, 極めて有用な算法である.

3 Fermat の小定理

正の整数 n に対し, $1, 2, \cdots, n-1$ のうち, n と互いに素 (GCD が 1) である数の個数を, Euler の函数 (totient function) といい, $\varphi(n)$ で表す. n を $p_1^{e_1}\cdots p_r^{e_r}$ と素因数分解したとき,

$$\varphi(n) = p_1^{e_1-1}(p_1-1) \cdots p_r^{e_r-1}(p_r-1)$$

である。特にnが素数であることと、$\varphi(n)=n-1$であることとは同値である。上の式は、日本の和算家がEulerよりも早く発見している。

次の事実は周知と思うが、厳密には互除法(1節)によって証明される定理である。

1° a, bがnと互いに素ならば、abもnと互いに素である。特にabが素数pで割り切れれば、a, bの少くとも一方はpで割り切れる。

2° a, b, cがnと互いに素で、$a \equiv b \pmod n$ならば、$ca \equiv cb \pmod n$である。

Fermatの小定理(の一般化)。aがnと互いに素ならば、$a^{\varphi(n)} \equiv 1 \pmod n$。特にnが素数pなら、aがpの倍数でないとき
$a^{p-1} \equiv 1 \pmod p$, $a^p \equiv a \pmod p$

略証 nと互いに素な数を$c_1, c_2, \cdots, c_{\varphi(n)}$とする。これらにaを乗ずれば、$ac_1, ac_2, \cdots, ac_{\varphi(n)} \pmod n$はすべて相異なり、nと互いに素だから、$c_1, \cdots, c_{\varphi(n)}$の置換である。これらを乗じて $c_1 \cdots c_{\varphi(n)} \equiv a^{\varphi(n)} c_1 \cdots c_{\varphi(n)}$; 上記の1°, 2°から $a^{\varphi(n)} \equiv 1 \pmod n$」

これにより、$\varphi(n)$が既知なら、$a^{-1} \pmod n$は、$a^{\varphi(n)-1} \pmod n$として作ることもできる。

付記　RSA 暗号で, 指数の対 e, d は, $ed \equiv 1 \pmod{\varphi(n)}$ でなくても, $ed \equiv 1 \pmod{l}$, $l = L.C.M.(p-1, q-1)$ でよい. この l を大きくするため, $p-1, q-1$ は2以外に公約数をもたない対であることが望ましい.

また l 或は $\varphi(n)$ が大きすぎて, d の計算が(直接に)困難なときには, 次のような方法がある. 1°の方法で, まず

$$eu_1 + (p-1)v_1 = 1, \qquad eu_2 + (q-1)v_2 = 1.$$
$$w_1(p-1) + w_2(q-1) = \delta \ (= G.C.D.(p-1, q-1))$$

である u_i, v_i, w_i を求める. このとき $u_1 - u_2$ は必ず δ の倍数 δk である. $u_1 - u_2 = k w_1(p-1) + k w_2(q-1)$ であるから,

$$d = u_1 - k w_1(p-1) = u_2 + k w_2(q-1)$$

ととればよい. この d が負ならば, l (または $\varphi(n) = (p-1)(q-1)$) を加えて正にする. もし $k w_1, k w_2$ が大きくなりすぎたら, それぞれ $\bmod (q-1), (p-1)$ で還元した数で置き換える.

[参] 長田順行他, 暗号, 数理科学増刊, 1982年5月

一松 信, 暗号の数理, 講談社, 1980

一松 信監修, データ保護と暗号(仮題) 日本経済新聞, 1983年8月刊行予定

D.E.R.Denning, Cryptography and Data Security, Addison-Wesley, 1982.