暗号の種類とその構造

２年Ｂ組　　森　宇宏
２年Ｃ組　　佐藤　圭
２年Ｃ組　　清水　悠平
指導教諭　　川口　慎二

１．要約

サイエンス研究会数学班は、暗号理論と数学の関係について、参考文献[1]を輪読しながら、研究活動を行っている。本稿では、「シーザー暗号・スキュタレー暗号・単一換字暗号・DES・AES・公開鍵暗号」の６種類の暗号について、その暗号の歴史や(数学的)構造、暗号化と復号化の手順、またそれぞれの有する特徴を紹介する。

キーワード　　暗号化、復号化、鍵、平文、解読、シーザー暗号、スキュタレー暗号、単一換字暗号、DES、AES、公開鍵暗号

２．研究の背景と目的

現在は、生活のいたるところにコンピュータが用いられ、実に様々な情報がやり取りされている。氏名や住所といった個人情報を保護するため、また、インターネットを介した金銭や物品の受け渡しにおける安全性の確保のため、データを暗号化し、必要以上にさらさないことが求められている。このような情報の管理は古代でも同様であり、戦争や外交などの機密情報を交換するために、様々な暗号が考案され、そのたびに解読されるという歴史を辿ってきた。現在でも、より安全性の高い暗号を開発するために、日々研究と改良が重ねられている。

そのような暗号の基礎理論には、整数の剰余や楕円曲線といった数学的概念が利用されている。本稿では、種々の暗号の構造や解読方法を紹介しながら、そこに隠れた数学的な概念を紹介する。

３．研究内容

日本語で用いられる文字（ひらがなやカタカナなど）でも、アルファベットと同じように暗号のために用いることができるが、分かりやすくするために、本稿ではアルファベットのみを使って説明する。なお、小文字（a, b, c, …）で平文のアルファベットを表記し、大文字（A, B, C, …）で暗号文のアルファベットを表記することにする。

まず、暗号理論の基本事項についてまとめておこう。一般に、情報をやり取りする際に、送り手を**送信者(sender)**、受け取り手を**受信者(receiver)**という。送信者から送られた情報を受信者が受け取るまでに、悪意ある第三者がその情報を知ることがある。このような悪意ある第三者を**盗聴者(eavesdropper)**と呼ぶことにする。

送信者は、情報が盗聴者に漏れないように、伝えたい情報を受信者のみが理解でき

るように形を変えて送ることにします。このとき、元々の情報を**平文(plaintext)**、形を変えた文章を**暗号文(ciphertext)**といい、このような処理のことを**暗号化(encrypt)**という。受信者は、送られてきた暗号文を、送信者とあらかじめ決めておいた規則によって再変換することで、最初の情報を手に入れることが可能になる。このあらかじめ決めておいた規則を**鍵(key)**といい、その処理を**復号化(decrypt)**という。

数学や情報科学では、問題を解決する手順のことを**アルゴリズム(algorithm)**という。暗号理論でも、「**暗号化のアルゴリズム**」や「**復号化のアルゴリズム**」という言い方で、暗号化や復号化の手順を指すことがある。さらに、暗号化・複合化のアルゴリズムをまとめて**暗号アルゴリズム**という。

３－１．シーザー暗号

シーザー暗号(Caesar cipher)とは、平文で使っているアルファベットを一定の文字数だけずらすことにより暗号化を行う。暗号の中でも、アルゴリズムが単純であり、誰にでも暗号化が容易にできる。

いま、アルファベットを３文字ずらすことによって暗号化を行うとする。この場合、平文のａは３文字の先のＤに暗号化され、ｂはＥに、ｃはＦに、以下同様に続けていき、ｖはＹに、ｗはＺにそれぞれ暗号化される。さらに、ｘはアルファベットを一周してＡに暗号化される。同様に、ｙはＢに、ｚはＣにそれぞれ暗号化される。図１を見ると、アルファベットを「ずらす」様子がわかるだろう。

■シーザー暗号の暗号化

ここで、秘密にしたい情報を angou（暗号）だと仮定する。この単語を秘密にしたまま受信者に届けるとする。このとき平文は

angou

の５文字である。

下のように平文を３文字ずつずらし、暗号化する。

a→D
n→Q
g→J
o→R
u→X

これで、「angou」という平文が「DQJRX」という暗号文に変換できたことになる。「angou」という単語なら意味はわかるが、「DQJRX」では意味がわからない。

シーザー暗号では「アルファベットの文字をずらす」という操作が「暗号化のアルゴリズム」に相当し、ずらす文字数が「鍵」に相当する。この例では、鍵は３である。

■シーザー暗号の復号化

さて、受信者が、DQJRX という暗号文を受け取ったとする。暗号文のままでは意味は分からないので、復号化して平文に戻す作業を必要とする。シーザー暗号の復号化は暗号化のときと同じ鍵を使い、逆方向にずらす操作を行う。上の例なら、３文字だけ逆方向にずらせばいいのである。つまり、

D→a
Q→n
J→g

R→o

X→u

というように復号化する。

これで元の平文「angou」に戻すことができた。このとき、3 という鍵は送信者と受信者との間で前もって決めておき、共有しておく必要がある。

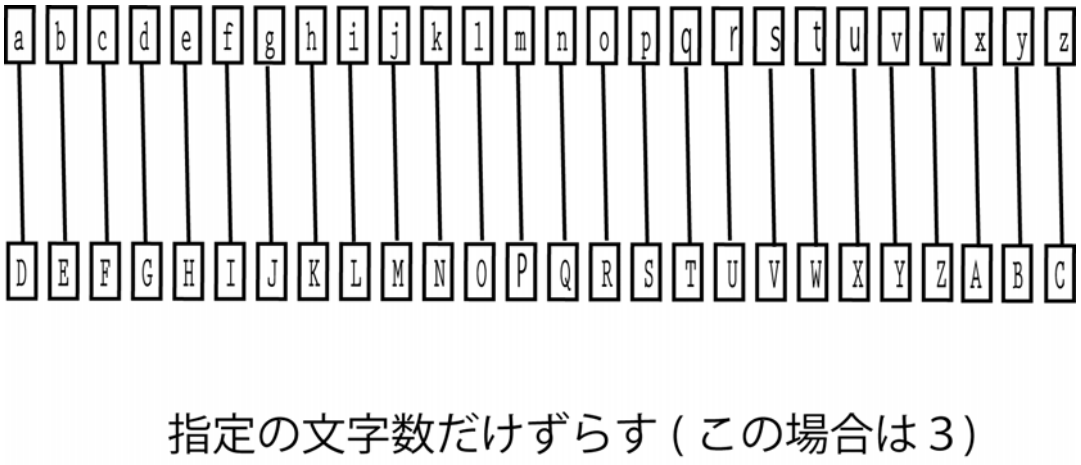

図１　シーザー暗号の仕組み

■シーザー暗号の解読

送信者と受信者以外の人間が、送信者の送った暗号文を知り、その平文を解読したいとする。

DQJRX→鍵 0 で復号化→dqjrx

DQJRX→鍵 1 で復号化→cpiqw

DQJRX→鍵 2 で復号化→bohpv

DQJRX→鍵 3 で復号化→angou

DQJRX→鍵 4 で復号化→zmfnt

DQJRX→鍵 5 で復号化→ylems

DQJRX→鍵 6 で復号化→xkdlr

DQJRX→鍵 7 で復号化→wjckq

DQJRX→鍵 8 で復号化→vibjp

DQJRX→鍵 9 で復号化→uhaio

DQJRX→鍵 10 で復号化→tgzhn

DQJRX→鍵 11 で復号化→sfygm

DQJRX→鍵 12 で復号化→rexfl

DQJRX→鍵 13 で復号化→qdwek

DQJRX→鍵 14 で復号化→pcvdj

DQJRX→鍵 15 で復号化→obuci

DQJRX→鍵 16 で復号化→natbh

DQJRX→鍵 17 で復号化→mzsag

DQJRX→鍵 18 で復号化→lyrzf

DQJRX→鍵 19 で復号化→kxqye

DQJRX→鍵 20 で復号化→jwpxd

DQJRX→鍵 21 で復号化→ivowc

DQJRX→鍵 22 で復号化→hunvb

DQJRX→鍵 23 で復号化→gtmua

DQJRX→鍵 24 で復号化→fsltz

DQJRX→鍵 25 で復号化→erksy

となり、このうち意味の通じるものを探すと、「鍵＝3」のとき angou を見つけることができる。これで解読ができたといえる。このように、考えられるすべての鍵のパターンを試みて、その中から鍵と平文を見つける解読方法を**ブルート・フォース・アタック(brute-force attack)**または**全数探索(exhaustive search)**という。

このように、暗号としての構造が単純なシーザー暗号は、送信者と受信者以外でも

解読が可能になってしまう欠点がある。

３－２．スキュタレー暗号

スキュタレー暗号(Scythia Cipher)は、古代ギリシアの都市国家スパルタで用いられていた暗号で、「スキュタレー」とは「棒」という意味である。最も古くから存在する暗号として知られている。

■スキュタレー暗号の暗号化

①長い帯状のもの（紙など）を長い棒（鍵になる棒）に巻きつける。
②巻きつけた鍵となる棒に、縦一列に伝えたい文（平文）を書く。
③平文とは関係のない文字を、同じように縦一列に紙が埋めつくされるまで書く。

■スキュタレー暗号の解読方法

暗号文の書かれた帯状の文を、暗号化で使った棒と太さが同じ棒に巻き、できた文の中から、意味の通じる部分を見つける。

暗号文を送る人と受ける人はお互いに同じ太さの棒を持っていなければいけない(この場合、この同じ太さの棒が鍵となる)。

３－３．単一換字暗号

平文のアルファベットを鍵の数だけ「ずらす」ことによって暗号化したものがシーザー暗号だった。しかし、このように規則的に変換しなくとも、アルファベット 26 文字それぞれに１対１の対応関係があれば、どんな対応関係でも暗号文が作れそうである。このように、平文を構成するアルファベットを別のアルファベットに変換する暗号のことを**単一換字暗号(simple substitution cipher)**という。単一換字暗号の対応表の例を図２に示す。

■単一換字暗号の暗号化

単一換字暗号の暗号化は平文を１文字ずつ換字表にしたがって変換していく作業の繰り返しである。

例えば angou という単語を暗号化してみましょう。図２より

a→D
n→A
g→C
o→O
u→X

となり、暗号文は DACOX になる。

■単一換字暗号の復号化

単一換字暗号の復号化は、暗号化のときに使った換字表を使い、暗号化の逆を行う作業である。単一換字暗号の復号化のときには暗号化のときに使った換字表が必要なので、送信者と受信者は同じ換字表をお互いに持っておく必要がある。この換字表が単一換字暗号の「鍵」なのである。

■単一換字暗号の鍵空間

「angou」をシーザー暗号（鍵＝3）で暗号化すると、「DQJRX」になる。一方、単一換字暗号（鍵は図２の換字表とする）で暗号化すると「DACOX」になる。　これらはどちらも意味が分からないが、この暗号文を見ただけで、シーザー暗号と単一換字暗号のどちらで解読すればよいかは分からない。シーザー暗号はブルート・フォース・アタックで解読できたが、単一換字暗号はブルート・フォース・アタックで

解読することは難しい。なぜなら、単一換字暗号の方がシーザー暗号よりもはるかに多くの鍵の候補があるからである。それを確かめるため、鍵の総数を計算する。

単一換字暗号の平文で使われるアルファベット a にたいしては、A, B, C, … ,Z の 26 文字（26 通り）のいずれかが対応する。b 対しては a を暗号化したものを除外した 25 文字（25 通り）が、以下同様に考えると c には 24 文字（24 通り）のいずれかが対応する。よって単一換字暗号の鍵の総数は、26!

=403,291,461,126,605,635,584,000,000 個であることがわかる。これだけ多いとブルート・フォース・アタックで調べるのは困難である。たとえば 1 秒間に 10 億個の鍵というスピードで調べたとしても、全ての鍵を調べ終えるのに 120 億年以上の時間がかかってしまうからである。平均すると、正しい鍵が見つかるまでの時間は約 60 億年なので、単一換字暗号の解読に対するブルート・フォース・アタックは良い方法ではない。このように、ある暗号に対して、使うことのできる「すべての鍵」の集合のことを**鍵空間(keyspace)**という。

■頻度分析による単一換字暗号の解読

ブルート・フォース・アタックで単一換字暗号を解読することは困難であることが分かった。そこで発明されたのが**頻度分析**と呼ばれる暗号解読法である。頻度分析とは、暗号文中の文字の頻度と他の一般的な文章の文字の頻度と調べて（英語の文章などの場合、頻度の高い単語としては

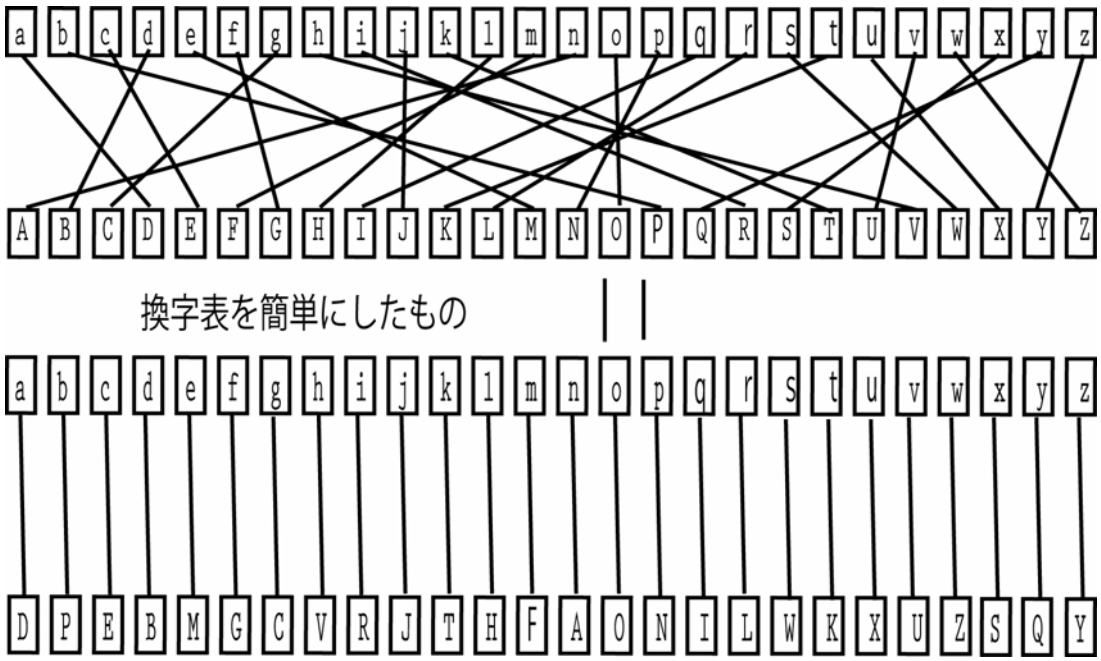

図２　単一換字表

the や in などで、アルファベットでは A や T など）その２つを比較し、暗号文を解読する方法である。この頻度分析を使えば単一換字暗号も解読することができる。

３－４．DES

DES (Date Encryption Standard)とは 1997 年にアメリカ合衆国の連邦情報処理標準規格（FIPS）に採用された対称暗号である。世界中の政府や銀行で用いられてきたが、コンピュータの進歩のため、現在ではブルート・フォース・アタックにより解読できるようになってしまった。RSA 社が行っている DES の鍵を見つけるコンテスト（DES challenge）の結果を見てみよう。

・1997 年 DES challenge Ⅰ　　96 日
・1998 年 DES challenge Ⅱ－1　41 日
・1998 年 DES challenge Ⅱ－2　56 時間
・1999 年 DES challenge Ⅲ
　　22 時間 15 分

でそれぞれ解読されている。DES による暗号文は短時間で解読できるようになったため、過去の暗号文を復号化する以外 DES を用いるべきではない。

■DES の暗号化プログラム

DES は 64 ビットの平文を 64 ビットの暗号文に暗号化する**対称暗号アルゴリズム(symmetric cryptgraphy algorithm)**である。

ここで、対称暗号アルゴリズムを説明するために、準備を行う。いままで、文字列の暗号について説明してきたが、現代はコンピュータによって情報がやり取りされることが多い。コンピュータでは、0 と 1 が並んだ**ビット列(bit sequence)**でいろいろな情報を表現する。例えば、文字も

m→01101101　t→01110100

のように８個の 0 と 1 の列としてあらわされる。このように、文字をビット列に対応させることを**符号化(encoding)**という。上のルールによる文字とビット列の対応は**ASCⅡ**と呼ばれている。

ビット列の処理において、**XOR (exclusive or)**、日本語でいう「**排他的論理和**」という計算規則がある。1 ビットの情報に対して XOR を施すと、

0 XOR 0 = 0　　1 XOR 0 = 1
0 XOR 1 = 1　　1 XOR 1 = 0

となる。0 を偶数、1 を奇数と見たとき、通常の和「+」のパリティ(偶奇)に一致するので、「XOR」の部分を「⊕」という記号で書く場合もある。さらに、２つの情報

A：01001100　　B：10101010

に対して XOR を施すと、A XOR B は、

$$A : 01001100$$
$$B : 10101010$$
$$A \oplus B : 11100110$$

となる。XOR には位上がりはないため、この規則を施せば、(A XOR B) XOR B = A であることが容易に確かめられるであろう。

さて、DES の暗号化アルゴリズムについて説明しよう。DES は、64 ビットの平文をまとめて 64 ビットの暗号文へと暗号化する。この 64 ビットごとのまとまりを**ブロック(block)**と呼ぶ。7 ビットおきに、1 ビットずつエラー検出用の情報が入るため、鍵のビット長は実質 56 ビットといえる。一般に、ブロック単位で処理を行う暗号アルゴリズムを**ブロック暗号(block**

cipher)と呼ぶため、DES はブロック暗号の一種になる。

DES で 1 度に暗号化できるのは、64 ビットだけであるが、それより長いビット長の平文を暗号化するためには DES による暗号化を繰り返す必要がある。この繰り返しの方法を**モード(mode)**という。

■DES の構造

DES の基本構造は、Horst Feistel が作った、**ファイステルネットワーク(Feistel network)**などと呼ばれるものが使われている。この構造は DES だけでなく多くのブロック暗号にも採用されている。ファイステルネットワークは、**ラウンド(round)**と呼ばれる暗号化の 1 ステップを何度も繰り返すことで暗号化を行う。DES の場合はラウンドを 16 回繰り返す。

図 3 にしたがってファイステルネットワークの 1 ラウンドを説明する。

①文字を ASCII に変換する。
②ASCII を左右均等に分ける。
③右をそのまま暗号文に挿入する。
④右をラウンド関数 f に送る。
⑤ラウンド関数 f は、右と**サブ鍵(subkey)**を使って、ランダムに見えるビット列を計算(XOR)する。
⑥得られたビット列と左との XOR を計算した結果を暗号化された左とする。

しかしこれでは「右」はまだ暗号化されていない。そこで次のラウンドのときは、出力の右と左を入れ替え、さらに異なるサブ鍵を用いて処理を行う。

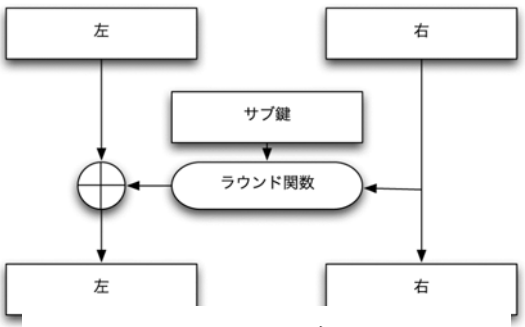

図 3　ファイステルネットワーク
（1 ラウンド分)の仕組み

複数のラウンドを持つファイステルネットワークを行う際も基本は同様であるが、ラウンドとラウンドの間では左右を入れ替えて、最終ラウンドの後では、左右の入れ替えを行わない点が異なる。

ところで、ファイステルネットワークで復号化を行うにはどうしたらよいだろうか。例えば、ファイステルネットワークの 1 ラウンド分の出力を、同じサブ鍵のファイステルネットワークにもう 1 度入れると、ラウンド関数 f がどんな関数であっても、正確にもとにもどるのである。これは XOR の性質（同じ数同士の XOR は 0 になること）から説明できる。

つまり、ファイステルネットワークの復号化は、サブ鍵を使う順番をラウンドごとに逆にするだけでできるのである。ファイステルネットワークの構造そのものは、暗号化と復号化で全く違いがない。

■ファイステルネットワークの性質

(1)ファイステルネットワークは、好きなだけラウンド数を増やすことができる。ラウンド数をいくら増やしても、復号化できなくなる心配はない。

(2)ラウンド関数 f にどんな関数を使っても良い。ラウンド関数 f は、復号化のことを考慮することなく、どんなに複雑にしてもよい。

ファイステルネットワークは、暗号化のアルゴリズムの中から「暗号の本質的な部分」をラウンド関数 f として抽出したことになる。ファイステルネットワークを使

えば､必ず復号化できることが保証される。暗号アルゴリズムを作ろうという場合は、複雑なラウンド関数 f を考えるよう努力すればよいのである。

３－５．AES（Rosalinde）

AES (Advanced Encryption Standard) とはこれまで標準であったDESに代わって新しい標準となる対称暗号アルゴリズムである。世界中の企業や暗号学者がAES の候補として多数の暗号アルゴリズムを提案しましたが、2000 年に Rijndael という対象アルゴリズムが AES として選定された。

■Rijndael (ラインダール)とは

Rijndael はベルギーの研究者 Joan Daemen と Vincent Rijmen が設計したブロック暗号アルゴリズムで2000年に次世代の標準暗号アルゴリズム AES として選定された。

Rijndaelのブロック長は128ビットで、鍵のビット長は 128 ビットから 256 ビットまで 32 ビット単位で選択することができる（ただし、AES の規格上では、鍵長は 128, 192, 256 ビットの 3 種類だけである)。

■Rijndael の暗号化と復号化

Rijndael は DES と同じく、複数のラウンドから構成されている。図５に 1 ラウンド分の概略を示しました。

DES ではファイステルネットワークという基本構造が使われていましたが、Rijndael ではファイステルネットワークではなく **SPN 構造(SPN structure)**という構造が使われている。

①Rijndael の入力ブロックは 128 ビット、すなわち 16 バイトである。16 バイトの入力に対してそれぞれ１バイトごとに **Sub Bytes** という処理が行われる。Sub Bites というのは、１バイトの値（0~255 のいずれかの値）を添え字とし､256 個の値を持っている換字表から１個の値を得るという処理である。

②Sub Bites の次に行われるのが、**Shift Rows** という処理である。これは、Sub Bites の出力をバイトごとに混ぜこぜにする処理である｡図５の線をたどって行けば､規則的に混ぜているのがわかる。

③Shift Rows の次に行われるのが、**Mix Columns** という処理である。これは、4 バイトの値をビット演算により別の 4 バイトの値に変換する処理である。

⑤最後に Mix Columns の出力とラウンド鍵との XOR をとる **AddRoundKey** という処理を行う。

これで Rijindael の１ラウンドが終わる。実際は以上のラウンドを 10～14 回繰り返すことになる。

以上の構造をみると､１回のラウンドで入力のすべてのビットを暗号化している。１回のラウンドで入力の半分のビットしか暗号化しないファイステルネットワークに比べて､ラウンド数を少なくできるメリットがある。また、SubBites はバイトごとに、ShiftRows は行ごとに、MixColumns は列ごとに並列処理ができるというメリットもある｡図６に Rijndeal の１ラウンド分の復号化を示す｡この図から、

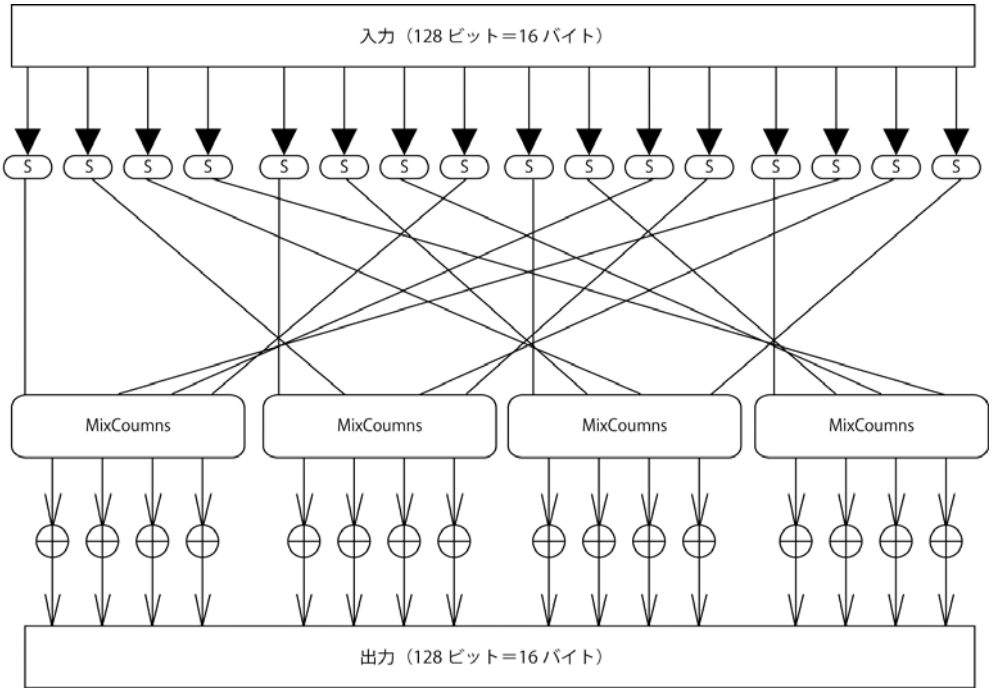

図５　Rijndael の暗号化(１ラウンド)

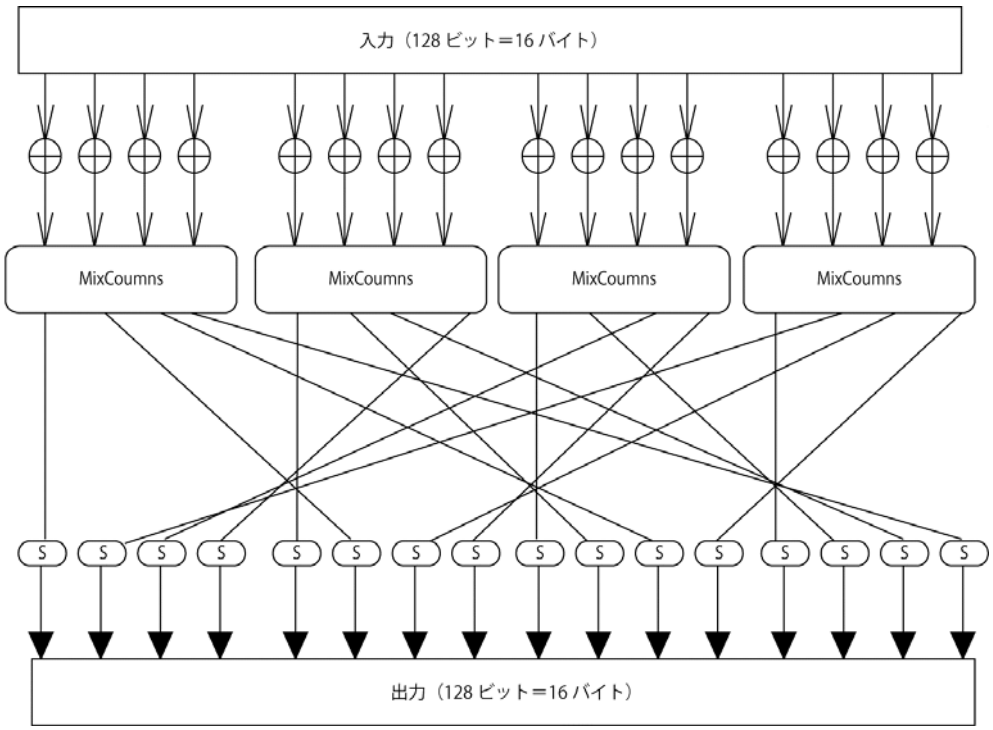

図６　Rijndael の復号化(１ラウンド)

SubBytes
ShiftRows
MixColumns

のそれぞれに対して、

InvSubBytes
InvShiftRows
InvMixColumns

という逆の処理(逆演算)が用意されていることがわかるだろう。これは、Rijndael は、ファイステルネットワークのように、１つの構造で暗号化と復号化を行うことができないことに起因する。

■Rijndeal の解読

Rijndeal に対しては今までにない新しい種類の攻撃が懸念されている。Rijndeal のアルゴリズムでは、平文から暗号文を作り出す計算を数式で表現することが可能なのである。これは過去の暗号アルゴリズムにはなかった特徴である。もしも Rijndeal の数式を数学的な操作によって解くことができるなら、Rijndeal を数学的に解読できることになる。これは、過去になかった攻撃の可能性を示唆している。

ただし、これはあくまで可能性の話であり、Rijndeal に対する有効な攻撃は、現在のところ見つかっていない。

３－６．公開鍵暗号

■鍵配送問題

対称暗号を使おうとするとすぐに**鍵配送問題(key distribution problem)**にぶつかる。鍵配送問題とは、たとえばメールを送る際に他人に平文を知られないように暗号化した文章を送っても、受信者が復号化の鍵を持っていなければ受信者は平文を理解することはできない。この問題を解決するために鍵を暗号文と一緒に送ると、このメールを他人が見ていた場合、簡単に暗号化されてしまう。つまり、「鍵を送らなければならないが、送ってはいけない」という状況が生じるわけである。

このような鍵配送問題解決策はいくつかあり、その中の一つに公開鍵暗号がある。

公開鍵暗号とは「暗号化の鍵」と「復号化の鍵」の２つの鍵を用いた暗号アルゴリズムである。送信者は「暗号化の鍵」を使ってメッセージを暗号化し、受信者は「復号化の鍵」を使って暗号文を復号化しである。公開鍵暗号を使う場合はこの２つの鍵をきちんと理解しておく必要がある。ここで「暗号化の鍵」と「復号化の鍵」を区別するとつぎのようなことがわかる。

(1)送信者が必要なのは「暗号化の鍵」だけである。
(2)受信者が必要なのは「復号化の鍵」だけである。
(3)盗聴者に知られて困るのは「復号化の鍵」だけである。
(4)「暗号化の鍵」は盗聴者に知られてもかまわない。

したがって、受信者が「復号化の鍵」を手元においておき「暗号化の鍵」だけを送信者におくると「復号化の鍵」を配送する必要が無くなり、鍵配送問題を解決できる。

公開鍵暗号における「暗号化の鍵」は一般に公開することができる。このような鍵のことを「**公開鍵(public key)**」と呼ぶ。公開鍵は受信者にメールで渡そうが、web で公開しようが構わない。盗聴者に公開鍵が盗聴されることを気にすることもない。

一方、「復号化の鍵」は絶対に公開してはいけない。このような鍵を「**プライベート鍵(private key)**」と呼ぶ。プライベート鍵は、他人に見せたり、渡したりしてはいけない。プライベート鍵は自分の通信相手にさえも知らせてはいけない。

公開鍵とプライベート鍵は２本で一対になっているため、これらの鍵のことを「**鍵ペア(key pair)**」と呼ぶ。公開鍵で暗号化した暗号文は、その公開鍵とペアになっているプライベート鍵でなければ復号化できない。鍵ペアをなしている２本の鍵は、互いに密接な数学的関係がある。このため、公開鍵とプライベート鍵をそれぞれ個別に作ることはできない。

公開鍵暗号の利用者は、他人に公開しても良い公開鍵と、自分だけが使うプライベート鍵からなる鍵ペアを作成する。

■公開鍵暗号の歴史

1976 年、Whitfield Diffie と Martin Hellman により公開鍵暗号のアイディアが発表された。公開鍵暗号の具体的なアルゴリズムは提示されなかったが、暗号化の鍵と復号化の鍵を分けることと、公開鍵暗号がどのような特性を備えているべきかが示された。

1977 年、公開鍵暗号の具体的なアルゴリズムとして、Ralph Merkle と Martin Hellman による**ナップサック暗号**が作られました。この暗号アルゴリズムは特許がとられましたが、後になって安全ではないことがわかりました。

1978 年、３人の研究者 Ron Rivest, Adi Shamir, Leonard Adleman は公開鍵暗号アルゴリズムの１つとして現在よく知られている **RSA (Rivest-Shamir-Adleman)** を発表しました。RSA は、現在の公開鍵暗号において事実上の標準といえる。

■公開鍵を使った通信の流れ

それでは、公開鍵を使った通信の流れを見ていこう。公開鍵の通信では受信者が先に動き出す。

①受信者は公開鍵・プライベート鍵の鍵ペアを作る。
②受信者は、自分の公開鍵を送信者に送る。
③送信者は、受信者の公開鍵を使ってメッセージを暗号化する。
④送信者は、暗号文を受信者に送る。
⑤受信者は、自分のプライベート鍵を使って、暗号文を復号化する。

図７を見ると送信者と受信者の間で何が行われているかを確かめてもらいたい。流れているのは公開鍵と暗号文だけなので、盗聴者は暗号文を解読することはできない。盗聴者は公開鍵を持っているかもしれないが、公開鍵は「復号化の鍵」ではなく「暗号化の鍵」なので復号化することはできない。

■公開鍵暗号でも解決できない問題

公開鍵暗号によって鍵配送問題は解決できた。しかし、これですべての問題が解決されたわけではない。入手した公開鍵が、本当に正しい公開鍵であるかどうかを判断する必要があるからである。これは公開鍵の**認証(authentication)**の問題である。また、公開鍵暗号は対称暗号に比べて処理速度が何百倍も遅いという問題もある。

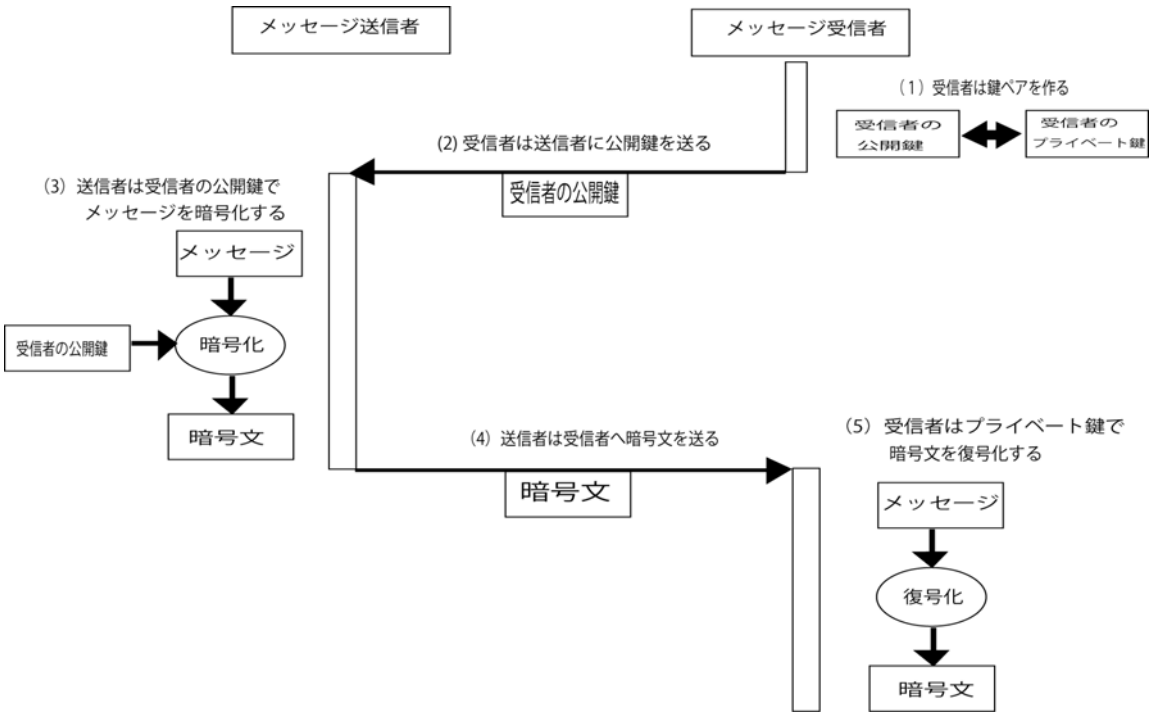

図７　公開鍵暗号における暗号化と復号化の流れ

４．考察

シーザー暗号は、アルファベットを「ずらす」数が、スキュタレー暗号では、棒の太さが鍵である。スキュタレー暗号では、もとの巻紙で一定の間隔で平文の文字が並ぶことになる。例えば、７文字の間隔で平文を構成する文字が並んでいる場合、巻紙の端から文字に番号を付けていくと、７で割った余りが等しい番号の文字を読んでいけばよいことになる。一般に、２つの整数 m,n を整数 a で割った余りが等しいとき、m と n は **a を法として合同(equivalent modulo a)**であるといい、

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

と表す。つまり、スキュタレー暗号には、「整数の合同」という数学的概念が利用されている。

５．今後の課題

本年度は、参考文献[1]を輪読し、６種類の暗号について、その暗号アルゴリズムと特徴を調べてきた。しかし、これらの暗号の背景にある数学についてはまだ十分に考察できていない。今後はそのような数学についてもさらにまとめていきたい。

６．参考文献

[1]「暗号技術入門―秘密の国のアリス」結城浩、SoftBank Criative (2003)
[2]「暗号の科学」、熊谷直樹、すばる舎、(2007)

７．謝辞

顧問の川口先生には、研究活動において様々なアドバイスをいただきました。ありがとうございました。