暗号化方式とその活用について

野呂颯太[[1]](#footnote-0)

2023年08月07日

## はじめに

この高度情報化社会において、我々は常に情報漏洩などのセキュリティリスクにさらされている．

暗号化に対する知識を深めその性質を理解し正しく利用することで安全に情報を保存する手段を手に入れることができると考えている．

本論文はその暗号化技術を知り、情報漏洩などのセキュリティリスクに備えるためのものである．

# 1. 導入

暗号化とはデータを安全に保存するためのアルゴリズムでありその必要性は時代が進むに連れ高まってきている．

暗号化方式にまつわる記録は非常に古くから存在している．

つまり暗号化技術というものは昔から多くの人間が取り組み研究を行ってきたものと言うことができる．

その数多くのアルゴリズムの中から有名である暗号化方式を選ぼうと思う．

以上のことから我々は、以下の代表的な分類の暗号化方式を研究していく[1]．

* 換字式暗号
* 転置式暗号
* 共通鍵暗号
* 公開鍵暗号

# 2. 換字式暗号

換字式暗号とは平文を文字ごとに置換する暗号化方式である．

換字式暗号は単純な暗号化方式であるため、初心者でも簡単に理解することができる．

換字式暗号には以下の種類が存在する[2]．

* シーザー暗号[[2]](#footnote-1)
* アフィン暗号
* ヴィジュネル暗号[[3]](#footnote-2)
* ボーフォート暗号
* ROT13

換字式暗号については代表的なシーザー暗号について調べていく．

シーザー暗号は非常に簡単な仕組みで暗号化を行っている．

その暗号化の方式は文字列を3つ左へずらすという非常に簡単なものである(図 1)．

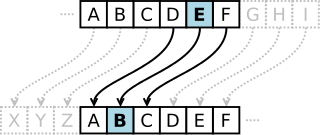


図 1 : シーザー暗号の解説図[3]

実際にこの暗号化方式を使用して暗号化を行っていこうと思う．

まず変換を容易にするため変換テーブルを作成する(図 2)．

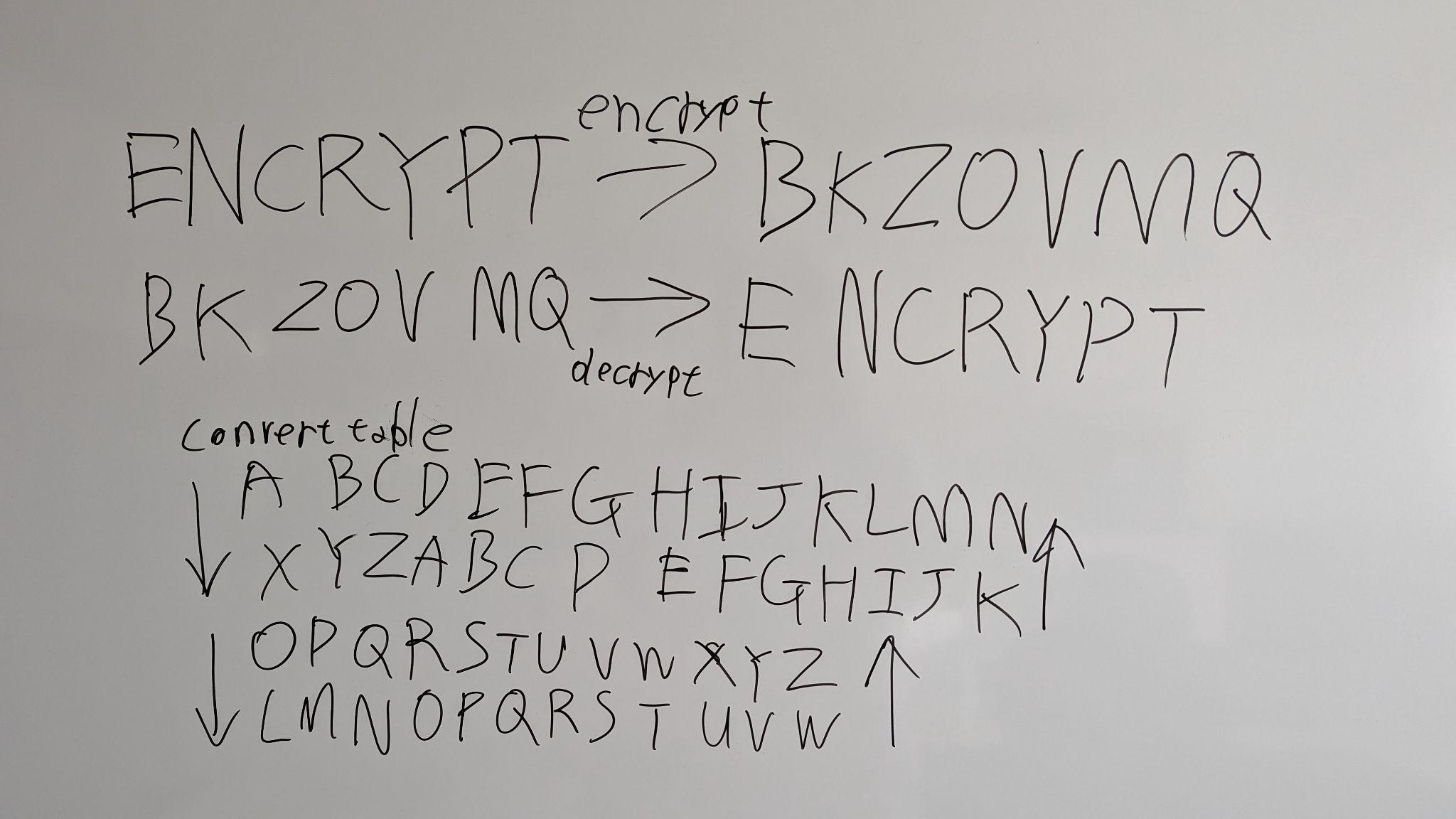


図 2:変換テーブル

次に平文を定義する今回は’ENCRYPT’を平文とする

そして変換テーブルに従い変換を行う(図3)．

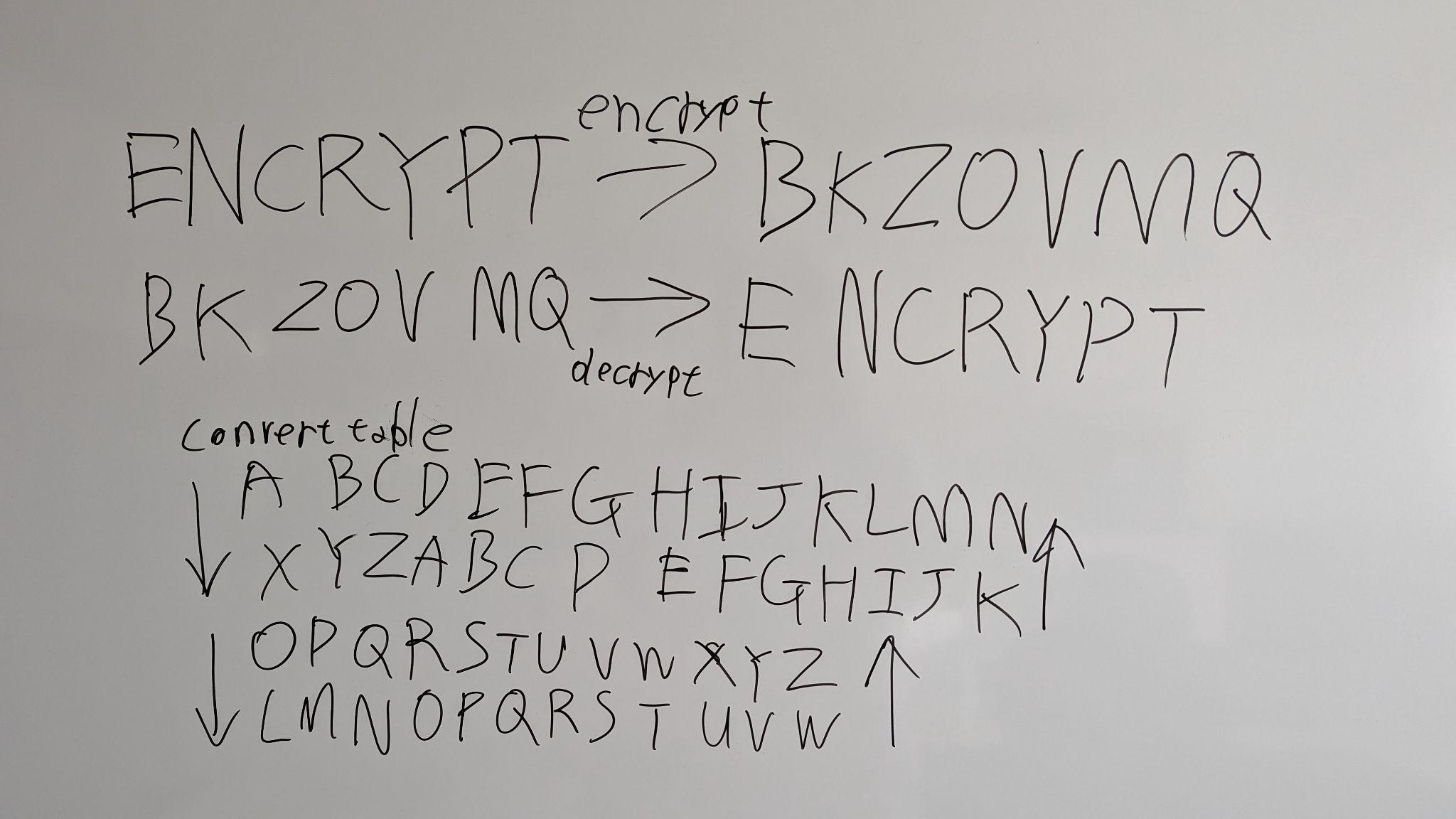


図 3:シーザー暗号を利用して暗号化を行った物

次に生成した暗号文を平文へ復号する(図4)．

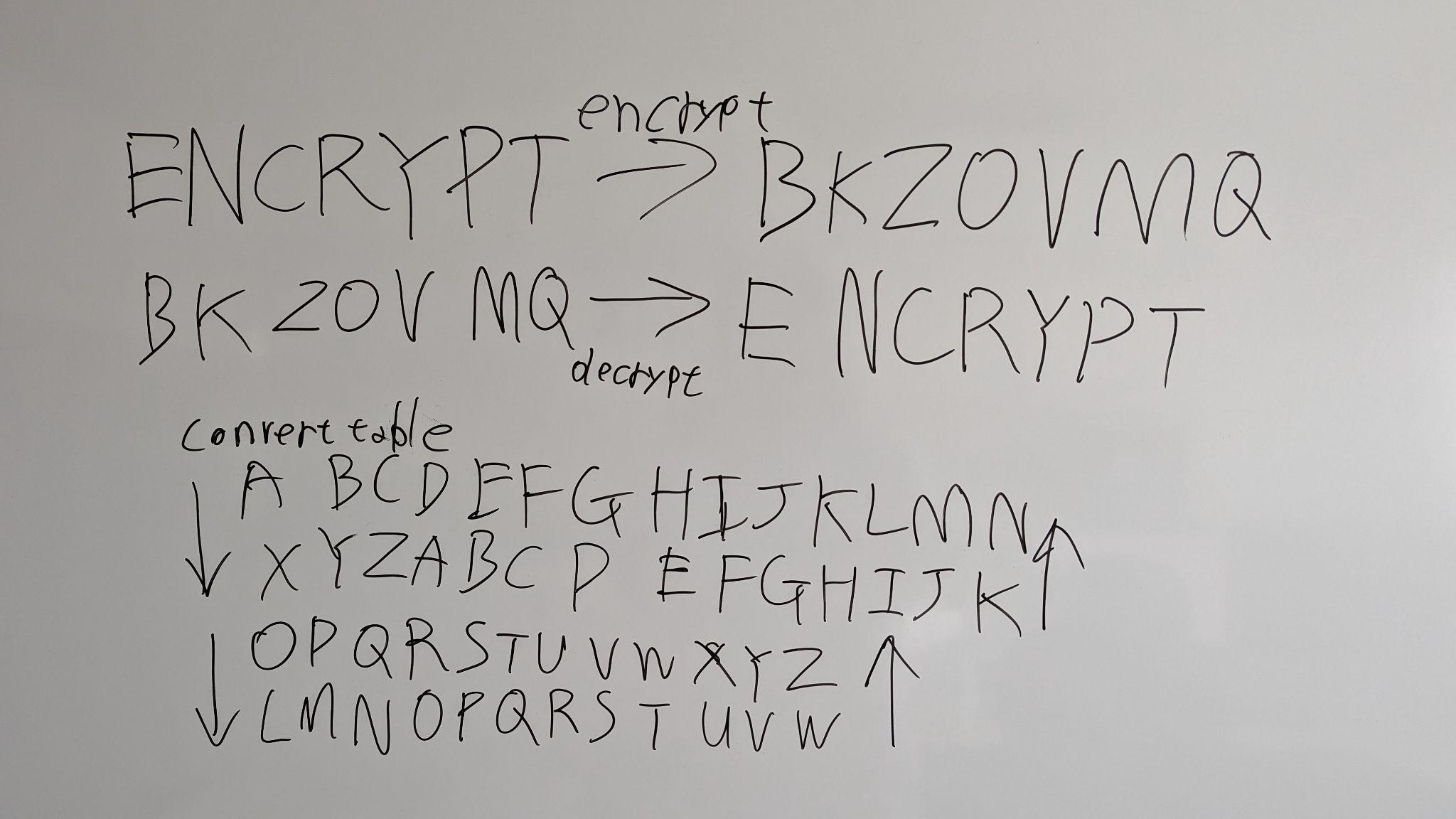


図 4:暗号文を復号化したもの

このようにシーザー暗号は変換テーブルを用意するだけで容易に変換が可能になっている．

つまり暗号文の文字のシフト数がわかってしまえば簡単に復号化されてしまうと言う脆弱性が存在するということだ．

さらに文字を入れ替えて暗号化を行っているだけなので長文になると頻度分析[[4]](#footnote-3)などを利用して暗号文を解読できてしまう．

実際に頻度分析を用いて復元を行っていこうと思う．

頻度分析を行うため暗号文は比較的長めの文が好ましい．

すでに暗号化された文を用意した.

uhvshfw wkh sulydfb ri rwkhuv.

wklqn ehiruh brx wbsh.

zlwk juhdw srzhu frphv juhdw uhvsrqvlelolwb.

暗号化された文は3行ある．

頻度分析を行おうと思う．

まず文字の頻度を書き出してみる(図5)．

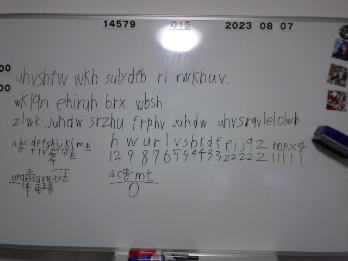


図 5: 頻度分析の結果

英字の文章に含まれる文字の頻度が一番高いのはeである.．

そのため暗号文中最も多く出現しているhはeである可能性が非常に高い．

文字のシフト数を計算するとe、f,g、hでシフト数は3である.

そのため文章を3でシフトすると(図6)．

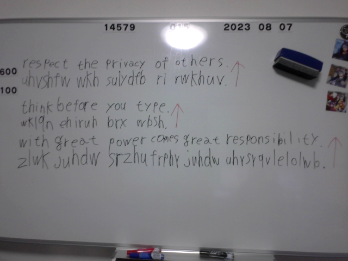


図 6:復号化

respect the privacy of others.

think before you type.

with great power comes great responsibility.

このように平文を解読することができる．

# 3. 転置式暗号

転置式暗号は換字式暗号とは違い、文字自体ではなくその配列を入れ替える暗号化方式である．

転置式暗号には以下の種類が存在する．

* スキュタレー暗号[[5]](#footnote-4)

転置式暗号では主にスキュタレー暗号について調べていく．

スキュタレー暗号は円柱状の物体に紙を巻き付け文字列を記入し棒から解くという暗号化を行う．

そして同じ直径の円柱状の物体に巻き付け、暗号を解読するというものである．

巻きつける棒としてどの家庭にもあるラップの芯を使おうと思う．

そしてA4用紙を縦に4分割したものを一つ用意する．

平文を’SUBSCRIPTION’とする.

まずラップの芯にA4用紙を巻きつける．

巻きつけ終わったら書き込みやすいよう固定する．

固定が完了したら平文を記入していく．

記入が完了したら紙を取り外す．

暗号文はこのようになった．

再び同じサイズの棒に巻きつけ復号化を行う．

少し太い棒に巻きつけて復号化を行ってみる．

違うサイズの棒に巻きつけても復号化を行えないことが見てわかる．

# 4. 共通鍵暗号

共通鍵暗号とは暗号化、復号化ともに共通の鍵を使用して暗号化することが可能な暗号化方式である．

共通鍵暗号ではさらに分けて二種類の方式が存在する．

その方式はブロック方式とストリーム方式である．

前者、ブロック方式の例としては以下が挙げられる[6]．

* DES
* 3DES
* AES[[6]](#footnote-5)

後者ストリーム方式の例としては以下が挙げられる．

* RC4
* ChaCha2
* SNOW 2.0

ブロック方式ではAES暗号の暗号化について調べていく．

ブロック方式とは平文を一定の長さで区切り部分ごとに暗号化を行う方式である．

AES暗号ではブロック長が128ビット、鍵長は128,192,256ビットの中から選び利用できる．

ブロック長とは平文を区切る長さのこと．

鍵長は暗号化を行うための鍵の長さのこと．

初めに鍵の生成を行う．

今回の鍵長では256ビットで生成する．

鍵はOpenSSL[[7]](#footnote-6)を使いランダムに生成する(図8)．

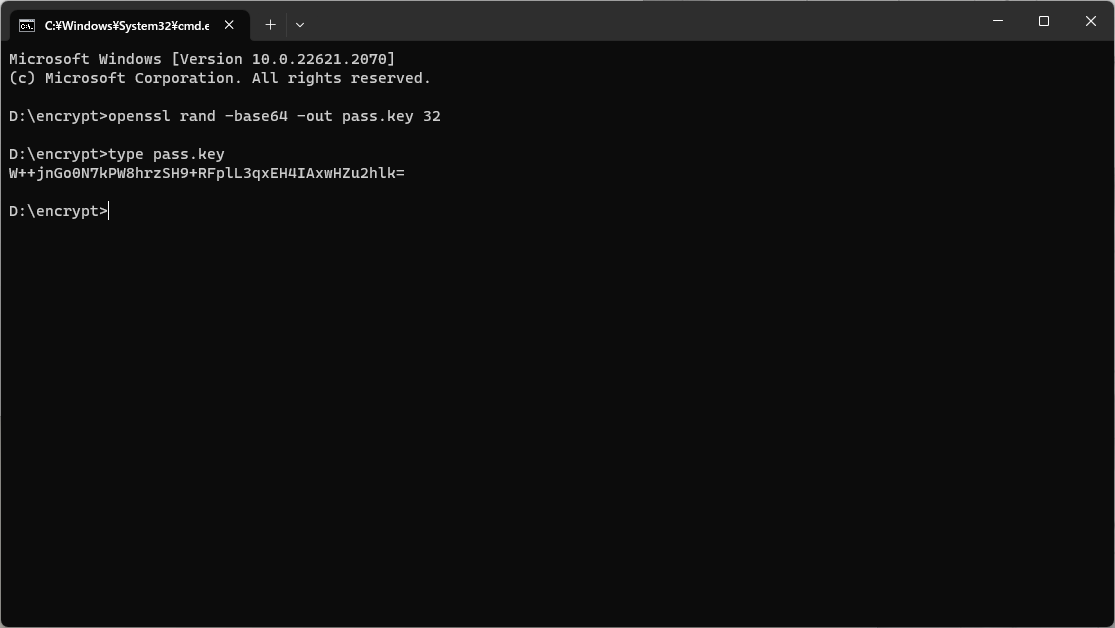


図 8:鍵を生成し、表示している

次に暗号化を行っていく．

ブロック方式は平文を一定の長さに区切り、暗号化を行う．

AES暗号では鍵長によってラウンド数が変化する．

128ビットなら10回、192ビットなら12回、256ビットなら14回．

次に平文を暗号化していこうと思う

平文はファイルstrings.txtに保存する．

先ほど生成したpass.keyを用いて暗号化を行う(図9)．

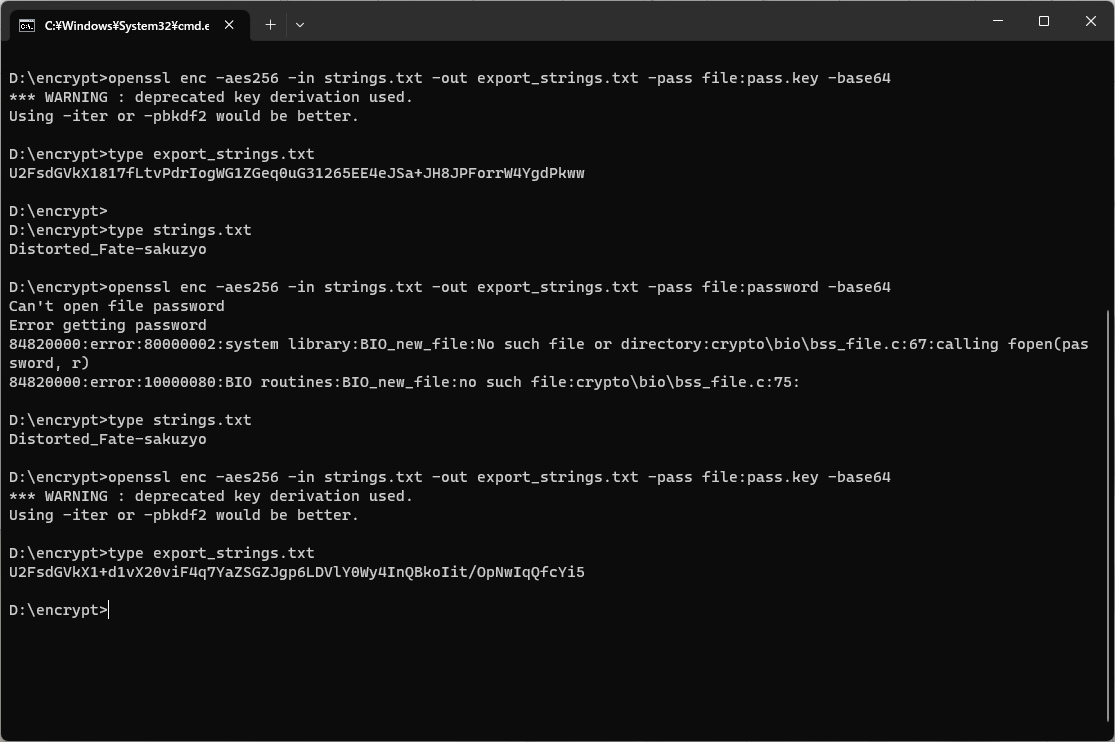
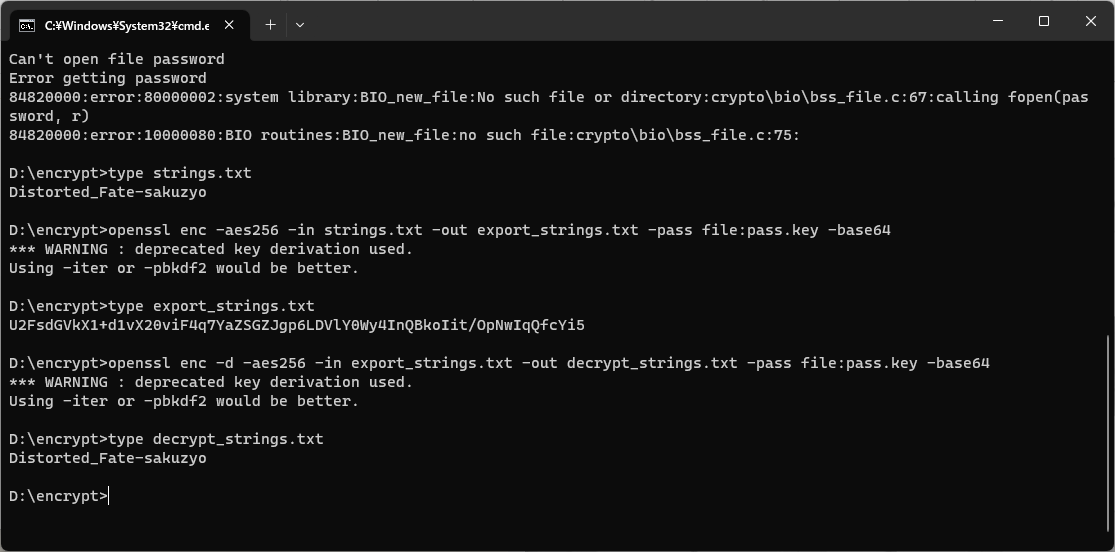


図 9:平文を表示し暗号化を行い、暗号文を表示している

暗号化された文をpass.keyを用いて復元する．



1. 図 10:復元を行い復号結果を表示している

しっかりと復元されていることが確認できる．

次に先ほどと同じ共通鍵、平文を使用して暗号化を手動で行おうと思う．

暗号化は次の手順で実行される．

1. SubBytes変換
2. ShiftRows変換

(上記の手動実行か下記の解析かどちらかを優先させる．)

# 5. 公開鍵暗号

公開鍵暗号とは暗号化に使用する鍵と、復号化に使用する鍵が異なる暗号化方式である．

公開鍵暗号には以下の種類が存在する．．

* RSA暗号
* ElGamal暗号

公開鍵暗号で最も標準的なRSA暗号を調べていこうと思う．

RSA暗号は桁数が多い合成数を素因数分解することが困難であることを利用した暗号化方式である．

暗号化、復号化に使用する数式は下記のものである．(式 1,2)

式 2: 上段 : 暗号化

下段 : 復号化

暗号化、復号化を行うために必要な定数として、公開鍵と秘密鍵がある．

以下はその求め方である．(式 3)

初期条件として、素数 p,q を選ぶ．

素数 e を選ぶ．

なお素数eはf以下となるように選ぶ．

次の式でdを求める．

e,nが公開鍵となりd,nが秘密鍵である.

式 3: 定数の算出方法

暗号化に際して一つ注意点が存在する．

それは平文はn以下でないとならないということだ．

今回もOpenSSLを使用して暗号化を行う．

暗号化方式をRSAとして公開鍵と秘密鍵を作成する．

そして平文を作成する．

公開鍵を用いて暗号化を行う．

秘密鍵を用いて復号化を行う．

プログラムを使用すると簡単に求めることができる．

では上記の点に気をつけて実際に暗号化を行って行く．

まず素数を求めるためにエラトステネスの篩[6]を使っていこうと思う．

手順としては以下の通り(図2)

1. 求めたい素数の最大範囲を決める
2. 1から最大範囲までのリストを作成する
3. リストの最初の数をFalseにする
4. リストを先頭から確認しTrueである数にあたる時その数を素数として素数用のリストに追加する
5. 上の手順で当たった数を変数として代入する
6. より以上の数での倍数にあたる数をFalseにする
7. 変数が最大範囲の平方根の数より大きくなるまで4~6を繰り返す
8. リスト内のTrueの数字すべてを素数用のリストに追加する

図 2: エラトステネスの篩の手順

最初は軽く100で実行してみよう

まず100の平方根を求める(式5)

式 5: 計算結果

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,

98, 99, 100]

次に最初の数を除外する

[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,22,

23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]

最初の数2を変数として2の累乗4以上の2の倍数を除外する

[3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]

[2]

最初の数3を変数として3の累乗9以上の3の倍数を除外する

[5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49, 53, 55, 59, 61, 65, 67, 71, 73, 77, 79, 83, 85, 89, 91, 95, 97]

[2, 3]

最初の数5を変数として5の累乗25以上の5の倍数を除外する

[7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 49, 53, 59, 61, 67, 71, 73, 77, 79, 83, 89, 91, 97]

[2, 3, 5]

最初の数7を変数として7の累乗49以上の7の倍数を除外する

[11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

[2, 3, 5, 7]

最初の数が11で100の平方根11を超えたため、リスト内の残りの数を素数用リストへ追加する

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

100までの素数用リストが完成した．

次に上記の手順で生成した素数を使用してRSA暗号を使用していこうと思う．

素数pを17、素数qを89とする(定数1)

定数 1

次にnを求める(式5)

次にfを求める(式6)

式 6

素数eをf以下の53とする(定数2)

次にdを求める(式7)

式 7

この公式はユークリッドの互除法を利用することで解くことができる．

解き方についてはあるサイトの内容を参考にした[9]．

#拡張ユークリッドの互除法より

式 8

上記の手順で算出した値を用いて暗号化を行っていこうと思う．

暗号化を行う式は以下の通りである(式8)

式 8

上記の式に代入して計算して行こうと思う.

平文を”1008”としてそれをUTF-8にしたがって変換する

暗号化を行う

復号化を行う

解読文と平文が一致しているため成功していると言える．

次にRSAの安全性について考えていく．

# 6.暗号化方式のまとめ

収集した暗号化アルゴリズムの情報を整理していく．

まず各暗号化方式を表にしてまとめたものが以下のものになる(図 10)

|  | 変換速度 | 安全性 | ファイルサイズ | 変換難易度 |
| --- | --- | --- | --- | --- |
| 換字式暗号 | 極めて早い | 低い | 不変 | 簡単 |
| 転置式暗号 | 早い | 普通 | 不変 | 簡単 |
| 共通鍵暗号 | 普通 | 高い | 不変 | 難しい |
| 公開鍵暗号 | 遅い | 極めて高い | 不変 | 極めて難しい |

図:10 各暗号化方式の概要

# 7.暗号化アルゴリズムの活用

収集した暗号化アルゴリズムの情報を元に暗号化アルゴリズムを組み合わせたものを作ろうと思う．

暗号化アルゴリズムの組み合わせと言っても、ただ繰り返し暗号化するのではなく効率的かつ利用しやすいものが望ましい．

利便性を考慮すると使用すべき暗号化方式は2種類にしたほうが良いだろう．

6の情報を参考に，組み込むべき暗号化方式を選ぶと公開鍵暗号と共通鍵暗号が好ましい.

参考文献

[1]Wikipedia [暗号] <https://ja.m.wikipedia.org/wiki/%E6%9A%97%E5%8F%B7>

[2]Wikipedia [換字式暗号] [換字式暗号 - Wikipedia](https://ja.wikipedia.org/wiki/%E6%8F%9B%E5%AD%97%E5%BC%8F%E6%9A%97%E5%8F%B7)

[3]Wikipedia [シーザー暗号] <https://ja.wikipedia.org/wiki/%E3%82%B7%E3%83%BC%E3%82%B6%E3%83%BC%E6%9A%97%E5%8F%B7>

[4]Wikipedia [ヴィジュネル暗号] <https://ja.wikipedia.org/wiki/%E3%83%B4%E3%82%A3%E3%82%B8%E3%83%A5%E3%83%8D%E3%83%AB%E6%9A%97%E5%8F%B7>

[5]Wikipedia [スキュタレー] <https://ja.wikipedia.org/wiki/%E3%82%B9%E3%82%AD%E3%83%A5%E3%82%BF%E3%83%AC%E3%83%BC>

[6]Wikipedia [ブロック暗号]

<https://ja.wikipedia.org/wiki/%E3%83%96%E3%83%AD%E3%83%83%E3%82%AF%E6%9A%97%E5%8F%B7>

[7]Wikipedia [エラトステネスの篩]

<https://ja.wikipedia.org/wiki/%E3%82%A8%E3%83%A9%E3%83%88%E3%82%B9%E3%83%86%E3%83%8D%E3%82%B9%E3%81%AE%E7%AF%A9>

[8]Chat GPT [ユークリッドの互助法を使用したRSAの秘密鍵の計算方法]

[9]高校数学の美しい物語[ユークリッドの互助法の証明と不定方程式]<https://manabitimes.jp/math/672>

1. 大館桜桂高等学校一年B組 [↑](#footnote-ref-0)
2. カエサル式暗号、シフト暗号とも言われる[3] [↑](#footnote-ref-1)
3. ビジュネル暗号とも書く[4] [↑](#footnote-ref-2)
4. 文章中の出現する頻度が高い文字(例えばE)をあてはめて解析を行う手法． [↑](#footnote-ref-3)
5. ギリシャ語で*σκυτάλη*、バトンの意味[3] [↑](#footnote-ref-4)
6. Advanced Encryption Standardの略称 [↑](#footnote-ref-5)
7. 暗号化に使用したソフトウェア<https://www.openssl.org/> [↑](#footnote-ref-6)