Pythonで学ぶ
暗号理論

博士(理学) 神永 正博
博士(工学) 吉川 英機 共著

コ ロ ナ 社

ま　え　が　き

本書は，専門的に暗号理論を学ぶ大学生，大学院生，技術者向けの教科書である。もともと，暗号の研究は，軍事的な目的で秘密裏になされていた。1970 年代の終わりに，標準暗号アルゴリズム DES や，公開鍵暗号が発明され，後にインターネットが普及してから暗号の研究が急速に拡大した。Wi-Fi，インターネットショッピング，IC 乗車券，携帯電話の SIM カード，SNS での暗号通信，ブロックチェーン技術など，いまや暗号なしの生活は考えられない。情報セキュリティ技術はソフトウェアを含む広大な領域であり，暗号理論はその一部にすぎないが，きわめて重要な領域である。暗号は，自己と他者を区別する免疫の役割を担っているからである。

暗号理論の核心部分は，共通鍵暗号，ハッシュ関数，公開鍵暗号の 3 つである。本書では，共通鍵暗号，ハッシュ関数，RSA 暗号，ラビン暗号，楕円曲線暗号の解説とともに，ブロック暗号に対する差分解読法，線形解読法，そしてハッシュ関数の解析，RSA 暗号に対する攻撃などを体験できる Python プログラムを提供する。本書を活用することにより，理論の学習だけでは実感が湧きにくい暗号の仕組みについて，プログラムを動かしながら楽しく学べると考えている。

Python には，暗号処理用のモジュールが用意されており，これらを使えば，ほとんどのことができる。そのためには，そこで何が行われているかを理解しておく必要がある。また，IoT 端末などでは，ハードウェア規模や電力リソースの制限などで，便利なモジュールを直接利用できないことも多く，そうした機器への暗号技術の実装のためには，原理面からの理解が欠かせない。本書では，暗号の解読技術を通じて「何が危険なのか」の説明に比較的多くのページを割いている。加えて，利用されている暗号のアルゴリズムについても，どのような難しさがあり，それがどう克服されているか，またはどのように現実と妥協しているかについて説明する。

暗号理論は，符号理論と並んで，実用的な技術の基礎理論としては際立って純粋数学的であり，特に代数学に関する知識が多用される。伝統的な工学教育では特に代数学の教育がきわめて限定的で，そのことが暗号理論を学ぶ際の大きな障害となっているように思われる。数学科で学ぶような代数学（群論，環論，体論，初等整数論）の基礎をすべて学んでから暗号理論を学ぶのは理想的だが，工学系の学生には負担が大きすぎるであろう。本書ではこうした知識を補い，Python で実際にアルゴリズムを実装して理解を深めていく。暗号で必要になる代数学の知識は多岐にわたるが，必要になった段階で導入する形を取った。

適宜，Python の操作やプログラムの実行などを含む基本的な問題が出題されているので，解いてみていただきたい。ごく一部を除いては簡単に解くことができるはずである。やや時間がかかりそうな問題には，∗ 印がついている。時間がないときは ∗ 印のついている問題を飛ばして

読んでも問題なく読み進めることができるように配慮した。プログラムの実行には，Spyder，Visual Studio 等の統合環境を想定しているが，実習の際には，Jupyter Notebook や Google colaboratory を使うほうが適しているかもしれない。読者にとって最も適当な環境で使っていただければ幸いである。

暗号学習者が陥りがちな罠は，暗号理論の一部のみの学習を深めてしまうことである。例えば，数学が達者な人は公開鍵暗号ばかり勉強してしまいがちであるが，それだけでは実用的なシステムをつくることは難しい。少なくとも，共通鍵暗号，ハッシュ関数，公開鍵暗号とその周辺技術，モジュールまで含めた理解とソフトウェア実装経験が必要であろう。

なお，本書を通じて，すでにモジュールが用意されていても，暗号の仕組みを理解するためにあえてプログラムを提供する場合がある。プログラムは練習問題の解答にあたるものも含めて，コロナ社のサイト（https://www.coronasha.co.jp/np/isbn/9784339029468/）より取得することができる。本書のプログラムはアルゴリズムを説明するためのものなので，説明の邪魔になる例外処理を含め情報セキュリティ上必要なパラメータのチェックなどは行っていない。そのまま製品やサービスに組み込まないようお願いするとともに，本書のプログラムは，あくまで暗号の基本原理を理解するためにのみお使いいただければ幸いである。多数のプログラムが提供されるが，何度も利用される処理をクラスにまとめるようなことはしていない。あちこちで重複が生じているが，アルゴリズム全体を理解するには，このほうが好都合だと思われる。

本書は，著者（神永）が，工学部の「情報セキュリティ工学」と大学院電気工学専攻における「暗号・情報セキュリティ工学特論」で毎年アップデートしながら講義してきた内容をベースにして執筆した。神永が全体設計を行い，共通鍵暗号・ハッシュ関数の解析を得意とする吉川が，第 3 章（の前半），第 4 章，第 6 章を執筆し，両者で全体の見直し，表現の統一などを行ってまとめたものである。東北学院大学の森島佑先生には暗号利用モードの処理についてご教授いただいた。また，神永研究室の大学院生の佐藤健人君にもプログラムのチェックをしていただいた。記して感謝したい。

2024 年 8 月

神永 正博，吉川 英機

本書の URL は，すべて 2024 年 8 月現在である。また，本書では，製品名に $^{\mathrm{TM}}$，® マークは明記していない。

目　　次

1. 共 通 鍵 暗 号

2. ブロック暗号の基礎

3. 現代のブロック暗号と暗号利用モード

4. ブロック暗号に対する差分解読法・線形解読法

5. ハッシュ関数とメッセージ認証子

6. ハッシュ関数の衝突シミュレーション

7. RSA 暗号と RSA 電子署名

8. RSA暗号の実装アルゴリズム

9. 素数生成

10. RSA暗号に対する攻撃

11. 平方剰余とラビン暗号

12. 楕円曲線と楕円曲線上の離散対数問題

13. 楕円曲線の暗号への応用

1 共通鍵暗号

本章では，共通鍵暗号の仕組みの概略，特に通信路暗号化とチャレンジレスポンス認証を説明し，素朴な暗号として，単換字式暗号とその解読方法である頻度解析を説明する。

1.1 共通鍵暗号の基本

1.1.1 暗号化，復号，秘密鍵

暗号理論で用いられる基本的な概念を説明しておこう。最もシンプルな共通鍵暗号のイメージは，**図 1.1** のようなものである。図 1.1 は，A 氏から B 氏にメッセージ P を暗号化して送る様子を表している。メッセージ（暗号化されていないもの）は，**平文**（plaintext）と呼ばれる。読みは，「ひらぶん」であって「へいぶん」ではない。平文が秘密鍵 K によって暗号化され，暗号文となる。共通鍵暗号では，秘密鍵 K は何らかの手段で通信前に共有されている必要がある。図 1.1 における E_K は暗号化関数であり，秘密鍵 K を用いて平文 P を暗号文 C に変換する。つまり，$C = E_K(P)$ と書くことができる。一方，D_K は復号関数である。K を固定したとき，D_K は逆関数 $D_K = E_K^{-1}$ になっている。つまり，$D_K(C) = E_K^{-1}(E_K(P)) = P$ となる。K を固定したときは，平文と暗号文は一対一に対応している。理想的な暗号では，K を知らない限り，C は乱数と区別できず，暗号文だけ見ても平文に関する情報が一切得られない。ただ，実際にはそうではないところが暗号理論の面白いところである。暗号解読を試みる者を攻撃者（アタッカー）という。

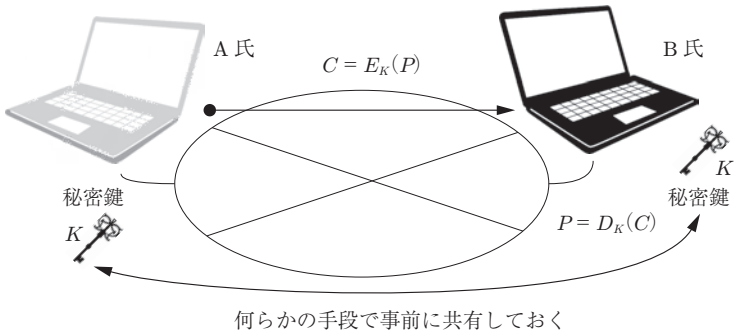

図 1.1 インターネットにおける共通鍵暗号化と復号のイメージ

1.1.2 チャレンジレスポンス認証

図 1.1 は，通信路暗号化という，最もわかりやすい暗号の使い方であるが，この他に重要な使い方として**チャレンジレスポンス認証**（challenge-responce authentication）がある。**図 1.2** は，IC 乗車券を例に取ったチャレンジレスポンス認証の概念図である。IC 乗車券は，カード内に IC チップがあり，そこで暗号の処理を行うことができる。改札からの電波による電磁誘導で電力を生み出し，IC チップに供給する非接触 IC カードである。まず，IC チップが改札に近づくと電磁誘導により IC チップが起動し，ゲートを開けるように要求する。これに対し，改札は，**チャレンジ**（challenge）と呼ばれる乱数 r を IC 乗車券に向けて送信する。IC 乗車券は，乗車券発行時に付与された秘密鍵 K を用いてチャレンジ r を暗号化し，$r' = E_K(r)$（**レスポンス**（response））を改札に送り返す。改札側（正確には改札につながっているサーバ）は，自らも保持している共通の鍵 K を用いてチャレンジを暗号化したものと r' が一致しているかどうかを確認する。これが一致していれば，カード側は発行者のリストにある鍵 K を保持していることがわかる。残高などが規定条件を満たせば，ゲートを開ける，という流れになる。チャレンジは毎回異なるので，K を持っていない限り正しい $E_K(r)$ を計算することはできない。これがチャレンジレスポンス認証である [†1]。チャレンジレスポンス認証の仕組みは幅広く用いられており，車の鍵として遠隔で開錠できるキーレスエントリでも同様である。

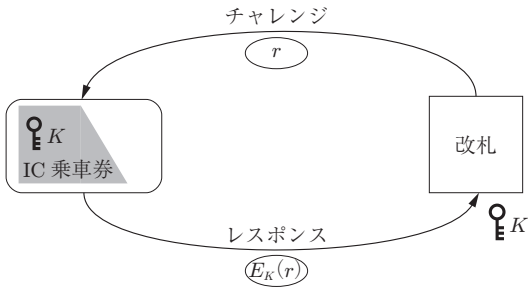

図 1.2　チャレンジレスポンス認証

1.1.3 暗号用乱数と安全なパスワード生成

なお，一般にチャレンジは**疑似乱数**（pseudorandom numbers）である。本物の乱数を疑似乱数に対比させて**真正乱数**（true random numbers）と呼ぶ。真正乱数に近いものがよい疑似乱数であるが，その「近さ」に関しては，少なくとも 2 つの視点がある。1 つは統計的な視点である。例えば，0 と 1 の頻度や，独立性などを統計的な検定で評価し，多くの検定を通過するものがよい疑似乱数である。この方向では，**メルセンヌツイスタ**（Mersenne Twister，MT）がきわめて優れた疑似乱数生成器である [†2]。MT は，松本眞と西村拓士によって発明された。

[†1] r はメッセージにあたるが，乱数であって，何か意味のある情報ではないことに注意されたい。

[†2] より高速な SIMD-oriented Fast Mersenne Twister（SFMT），Double precision SIMD-oriented Fast Mersenne Twister（dSFMT）もあり，現在広く利用されている。

Matsumoto-Nishimura[1)†1]は記念碑的論文である（論文は 1998 年掲載だが，結果は 1996 年の国際会議で発表されている）。MT は，$2^{19937}-1$（メルセンヌ素数）という長大な周期を持っており，モンテカルロシミュレーション（乱数を用いたシミュレーション）などではほとんどの場合満足のいく乱数を生成する。Python をはじめとして，多くのプログラム言語が，MT を採用して乱数を生成している。Python の `random` モジュールは MT を提供している。もう 1 つの視点は，予測困難性である。疑似乱数は所定の規則に従ってシード（初期値）を更新するアルゴリズムであるから，原理的には予測可能だ。しかし，暗号用途では，予測困難であることが必要である。MT は，予測困難ではないので，暗号用途で使うことは避けなければならない。MT のコード[†2]によれば，MT は内部ステートとして 624 個の 32 ビットの（unsigned long 型）整数を持っており，624 回出力すると内部ステートがすべて更新される。したがって，624 個の出力から MT の内部ステートを復元してつぎの出力が予測できる。

Python で暗号用の乱数を生成したい場合は，`secrets` モジュール[†3]を利用するのが一般的である。例えば，`secrets.randbits(k)` は k ビットの予測困難な疑似乱数を生成する。なお，C に標準装備されている `rand` は周期の短い線形合同法で生成された疑似乱数を提供するが，統計的視点で見てもまったく不十分なものであり，予測困難性の観点では論外である。

真正乱数は真円と同様に数学における理想的な概念であるが，暗号理論の文脈では，物理的な手段で生成される乱数を真正乱数と呼ぶことが多い。例えば，抵抗間の電圧のゆらぎをオペアンプで増幅し，そのアナログ信号をビット列に変換したり，高速なクロック信号を低速なクロックでサンプリングするなどの方法が用いられる。他にも環境電磁波を利用するなどの方法もある。これらは，温度や電波状況を故意に操作した場合に乱数の質が悪化する可能性があるが，通常は，物理乱数とよい疑似乱数生成器の組み合わせで十分安全な乱数が生成できると考えられている。

問題 1-1 `secrets` モジュールの `secrets.randbits(k)` を用いて，8 ビットの乱数を生成し，2 進数表示せよ。2 進数表示するには，`bin` 関数を用いればよい。

`secrets` モジュールを使えば安全なパスワードを生成するプログラムも簡単にできる。**リスト 1.1** のプログラムは，`length` で指定した文字数（ここでは 8）のランダムなパスワードを生成するものである。

リスト 1.1（StudySecurePassword.py）

```
1 import secrets
2 import string
3
4 length = 8
5 lst = string.ascii_letters + string.digits
6 password = ''.join(secrets.choice(lst) for i in range(length))
```

†1 肩付きの数字は，巻末の引用・参考文献の番号を示す。
†2 http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/MT2002/CODES/mt19937ar.c
†3 このモジュールは，オペレーティングシステム（OS）が提供する最も安全な乱数源に基づく乱数を生成するために用いられる。`secrets` は，`os.urandom` と `random.SystemRandom` に基づく。

```
7 print("password:", password)
```

string モジュールは，アスキー文字列，数字列などを含むもので，5 行目の lst は，アルファベットと数字を列挙したもの，つまり

```
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
```

である。6 行目で secrets.choice を使って，上記の文字列から，length で指定した回数だけランダム選択してつないでパスワードを生成している。

問題 1-2　リスト 1.1 のプログラムを修正して，10 文字のパスワードを生成してみよ。

1.1.4　リレーアタック

一見すると暗号を破ることなしにチャレンジレスポンス認証を破ることは不可能に思えるが，実際には抜け道がある。実際に犯罪に使われたものとして自動車のスマートキーに対する**リレーアタック**（vehicle relay attack）がある。リレーアタックは，スマートキーが車に近づくと（1 m 程度）鍵が開くという仕組みを利用した攻撃手法である。この物理的に鍵を差し込まずに解錠できるシステムは，**スマートエントリシステム**（smart-entry system）と呼ばれるもので，1993 年にシボレー・コルベットに採用された**パッシブ・キーレスエントリー・システム**（Passive Keyless Entry System）や 1998 年にメルセデス・ベンツ・S クラスに採用された「キーレス・ゴー」など広く用いられている。スマートエントリシステムでは，ドライバーが車から離れると自動的に施錠される。

車の盗難防止の仕組みとしては，**イモビライザ**（immobilizer）がある。イモビライザとは，電子的なキーの照合システムによって，専用のキー以外ではエンジンの始動ができないという自動車盗難防止システムのことである。物理的な方法でドアを開けてもエンジンがかからないという意味で効果的な盗難対策である。スマートキーと呼ばれるものは通常イモビライザキーを兼ねている。

リレーアタックでは，スマートエントリにおいて，スマートキーが近くにあることを車のコントロールユニットが検知して（弱い電波が発信されている）ドアロックが解除され，エンジンがかかる（イモビライザを兼ねているから）ことを利用して車を盗難する。スマートキーが近くにあるのと等価な状況をつくり出すことができれば施錠を解くことができるからである。一方の機器は車の隣に配置する必要があり，もう一方の機器は車の所有者のスマートキーの近くに配置する必要がある。

リレーアタックでは，自動車とスマートキーを持っているドライバーとの間に入り，アンプで信号を増幅して中継することでロックの解除が可能になる。信号は復号されるのではなくコピーされ，結果的には秘密鍵を傍受するのと同じ効果がある。リレーアタックによる車の盗難は，欧米で先行して確認され，2018，2019 年頃には日本でも被害が確認された。リレーアタックの手順をまとめるとつぎのようになる。犯行は通常複数名で行われることが多い（**図 1.3**）。

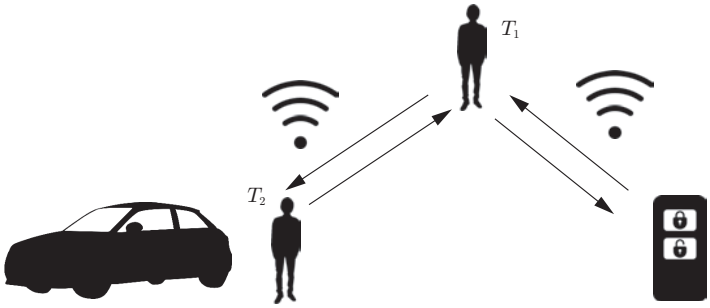

図 1.3 リレーアタック

1. グループの 1 人 T_1 がスマートキーに近づく（キーは運転手自身の手元にある場合もあるし，そうでない場合もある。また，運転手の手元ではなく，例えば，玄関先にキーが置かれている場合であれば，玄関に近づくという意味）
2. スマートキーの微弱電波を受信し増幅させて犯行グループの別の一人 T_2 に送信
3. 増幅させた電波を受けた T_2 が受信機を持って車に近づき，ロック解除
4. エンジン始動・盗難

リレーアタックにおいては，犯人は暗号を解読したわけではない。彼らは信号を中継しただけである。これは，**中間者攻撃**（man-in-the-middle attack）の一種である †。プロトコルを変えずに対策するには，スマートキーを金属缶などで遮蔽（しゃへい）することが比較的容易で有効な対策である。ポータブルな遮蔽グッズも販売されている。IC 乗車券などでリレーアタックが問題にならないのは，電波の中継が困難だからである。

1.1.5 暗号への攻撃法の種類

暗号に対する攻撃はアタッカーがどのような情報を得ることができるかによって分類されることが多い。暗号においては秘密鍵を復元する攻撃が最も強力である。秘密鍵が手に入れば，どんな暗号文でも復号できるからである。暗号理論においては，平文 P，暗号文 C，秘密鍵 K についての方程式

$$E_K(P) = C$$

を何らかの方法で「解く」ことが求められる。この観点で暗号への攻撃を分類すると以下のようになる。

- **選択平文攻撃**（chosen plaintext attack）： アタッカーが，任意の平文 P に対し，暗号文 C を得ることができる場合の攻撃
- **既知平文攻撃**（known-plaintext attack）： アタッカーが，既知の平文 P に対応する暗号文 C を得ることができる場合の攻撃

† 通常の中間者攻撃では，A 氏，B 氏がおのおの C 氏を通信相手と思っている（A 氏は C 氏を B 氏と思っており，B 氏も C 氏を A 氏と思っている）状況（なりすまし）をつくり出さなければならない。

索　　引

—— 著 者 略 歴 ——

神永　正博（かみなが　まさひろ）
1991年　東京理科大学理学部数学科卒業
1993年　京都大学大学院理学研究科修士課程修了（数学専攻）
1994年　京都大学大学院理学研究科博士課程中退（数学専攻）
1994年　東京電機大学助手
1998年　株式会社日立製作所勤務
2003年　博士（理学）（大阪大学）
2004年　東北学院大学講師
2005年　東北学院大学助教授
2007年　東北学院大学准教授
2011年　東北学院大学教授
　　現在に至る

吉川　英機（よしかわ　ひでき）
1989年　大分工業高等専門学校電気工学科卒業
1991年　電気通信大学電気通信学部通信工学科卒業
1993年　電気通信大学大学院博士前期課程修了
　　（電子情報学専攻）
1993年　鈴鹿工業高等専門学校助手
2000年　博士（工学）（大阪市立大学）
2003年　鈴鹿工業高等専門学校講師
2007年　東北学院大学准教授
2019年　東北学院大学教授
　　現在に至る

Python で学ぶ暗号理論

Cryptography with Python

2024 年 10 月 15 日　初版第 1 刷発行　★

検印省略

著　者　神永　正博
　　吉川　英機
発行者　株式会社　コロナ社
　　代表者　牛来真也
印刷所　三美印刷株式会社
製本所　株式会社　グリーン

112–0011　東京都文京区千石 4–46–10
発行所　株式会社　**コロナ社**
CORONA PUBLISHING CO., LTD.
Tokyo Japan
振替 00140–8–14844・電話(03)3941–3131(代)
ホームページ　https://www.coronasha.co.jp

ISBN 978–4–339–02946–8　C3055　Printed in Japan　（西村）