# 暗号技術入門

## 第3章「対象暗号」

# 暗号技術入門

出版：SB Creative

著者：結城 浩

第三版は2015年発行。暗号化技術の基礎から丁寧に解説されており、各所で初心者にお勧めされている。改訂版ではビットコインなど最新の情報も記載。

# 対象暗号

# 対象暗号

・イメージはスクランブルエッグ。ぐちゃぐちゃにしたら戻せない。

・平文をできるだけぐちゃぐちゃの暗号文にする。

・スクランブルエッグと違うのは鍵が分かれば復元できるところ。

# 文字の暗号からビット列の暗号へ

・今まで出てきた暗号たちはみんな文字をぐちゃぐちゃにする暗号。

・しかし、コンピュータは全てをビット列で表現する。

・現実世界のものをビット列で表現することを**符号化**と呼ぶ。

# 排他的論理和

0 XOR 0 = 0

0 XOR 1 = 1

1 XOR 0 = 1

1 XOR 1 = 0

# 排他的論理和

| | |
|---|---|
| 0 XOR 0 = 0 | 偶数(0) ＋ 偶数(0) = 偶数(0) |
| 0 XOR 1 = 1 | 偶数(0) ＋ 奇数(1) = 奇数(1) |
| 1 XOR 0 = 1 | 奇数(1) ＋ 偶数(0) = 奇数(1) |
| 1 XOR 1 = 0 | 奇数(1) ＋ 奇数(1) = 偶数(0) |

# ワンタイムパッド

# ワンタイムパッド

・本書では「**使い捨てパッド**」。山崎先生は「**ワンタイムパッド**」って呼んでた。
　発案者の名前から「**バーナム暗号**」とも。

・鍵空間を総当たりすればどんな暗号もいつかは解読される。

・しかしこれは違う。「**絶対に解読できない暗号**」

# ワンタイムパッドの暗号化

・ワンタイムパッドは「**平文のビット列とランダムなビット列のXORをとる**」ことで暗号化する。

| | | m | i | d | n | i | g | h | t |
|---|---|---|---|---|---|---|---|---|---|
| **平文のビット列** | | 01101101 | 01101001 | 01100100 | 01101110 | 01101001 | 01100111 | 01101000 | 01110100 |
| **ランダムなビット列 = 鍵** | XOR ⊕ | 01100110 | 01010001 | 00101001 | 01001001 | 01010100 | 01001010 | 00100100 | 01000100 |
| **暗号文のビット列** | | 00001011 | 00111000 | 01001101 | 00100111 | 00111101 | 00101101 | 01001100 | 00110000 |

# ワンタイムパッドの復号化

- 復号化は暗号化の逆計算となる。暗号文と鍵となったビット列をXOR計算すると平文を得ることができる。

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **暗号文のビット列** | | 00001011 | 00111000 | 01001101 | 00100111 | 00111101 | 00101101 | 01001100 | 00110000 |
| **鍵** | XOR ⊕ | 01100110 | 01010001 | 00101001 | 01001001 | 01010100 | 01001010 | 00100100 | 01000100 |
| **平文のビット列** | | 01101101 | 01101001 | 01100100 | 01101110 | 01101001 | 01100111 | 01101000 | 01110100 |
| | | m | i | d | n | i | g | h | t |

# なぜ解読ができない？

・ここでいう「**解読不可能**」とは「現実的な時間で解読するのが困難」という意味ではない。

・総当たり攻撃するといつかは“midnight”が出てくるが、総当たりなので“aaaaaaa”や“onenight”なども出てくる。

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **暗号文のビット列** | | 00001011 | 00111000 | 01001101 | 00100111 | 00111101 | 00101101 | 01001100 | 00110000 |
| **鍵** | XOR ⊕ | 01100110 | 01010001 | 00101001 | 01001001 | 01010100 | 01001010 | 00100100 | 01000100 |
| **平文のビット列** | | 01101101 | 01101001 | 01100100 | 01101110 | 01101001 | 01100111 | 01101000 | 01110100 |
| | | m | i | d | n | i | g | h | t |

# なぜ解読ができない？

- つまり平文として解釈可能な文章が無数に出てくるため、どれが本物なのか分からない。“mignight”が求めていた文章であることが分からない。
- ので「**解読不可能**」であると言える。

- 1949年、シャノンにより解読不可能なことが数学的に証明された。
- **無条件に安全**であり、**理論的に解読不可能**である。

# ワンタイムパッドは最強？

・でもない。非常に使いにくいのでほぼ実用されていない。理由は以下。

・**鍵の配送:** 最大の問題。平文と同じ長さの鍵。
鍵を安全に送る方法あるならそれで平文送れや。

・**鍵の保存**: 鍵が長い。このサイズの鍵を安全に保存できるなら以下略。

・**鍵の再利用**: 過去に使ったランダムビット列はもう使えない。漏洩した瞬間終わる。

・**鍵の同期**: とにかく鍵がでかい。うえに1ビットでもズレると詰む。

・**鍵の生成**: 乱数を大量に生成することになる。しかも疑似乱数でなく真の乱数。

# ワンタイムパッドは最強？

・全てクリアするには非常に莫大な手間と資金が必要になる。

・つまり、機密性が最優先事項である事例において採用される。

・例えば大国間のホットラインでは一部ワンタイムパッド方式が用いられる。

・その場合、きっと任を受けたエージェントが鍵をせっせと運んでいるのでしょう…。

・またストリーム暗号にも生かされる。これについては後の章で。

# DES

# DES

・Data Encryption Standard

・1977年にアメリカ合衆国の連邦情報処理標準規格に採用された対象暗号。

・世界中の政府や銀行で広く使われた。

・しかし、コンピューター技術の発展により総当たりで簡単に解読できるようになってしまった。

・RSA社が99年に行なったDES解読コンテストでは22時間ちょいで鍵が発見された。

# DES

ので使うべきではない

# DESの暗号化・復号化

・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象暗号アルゴリズム。

・鍵も64ビットだが7ビットおきにエラー検出情報が1ビット入るため実質56ビット。

・DESは64ビットをまとめて暗号化する。このまとまりを「**ブロック**」と呼び、
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**」と呼ぶ。

・64ビット以上の暗号化はDESを繰り返し行うことで実現。

# ファイステルネットワーク

DESの基本構造。

難しいので詳しいことはまたの機会に。

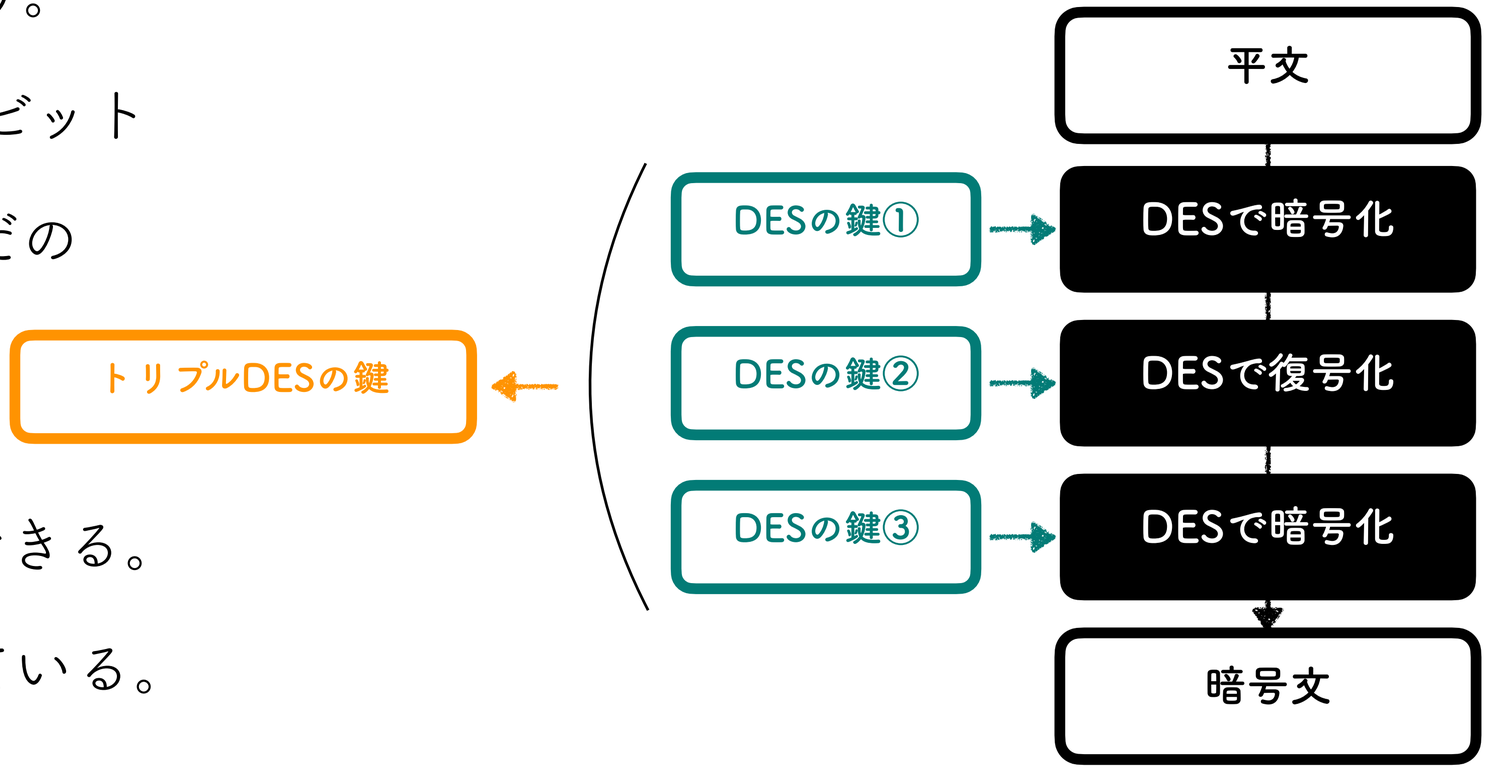
# トリプルDES

# トリプルDES

- その名の通りDESを３段重ねにした暗号アルゴリズム。
- 現実的な時間内に解読されるようになってしまったDESに代わるブロック暗号として開発。
- 3重DESやTDEAとも呼ばれる。

# トリプルDES

- 暗号化→復号化→暗号化を行う。
- トリプルDESの鍵は56*3=168ビット
- ①～③の鍵を等しくするとただの
  DESになる。
- つまり、DES暗号を
  トリプルDESを使って複合化できる。
- DESに対する上位互換を持っている。



# AES

# AES

・Advanced Encryption Standard

・DESもトリプルDESも古くなったので新しいのを作ることになった。

・米の標準化機関であるNISTにより公募で決定。

・全てはオープンでなければならず、隠すことによるセキュリティは認められない。

・選考プロセスでは世界中の専門家が評価する。

・このようなコンペ方式による標準化は暗号アルゴリズムの選定において正しい姿。

# AES - Rijndael

・最終的に選定されたブロック暗号アルゴリズム。

・AESは世界で広く使われている対象暗号アルゴリズムであり、
どんな暗号ソフトもAESをサポートする。

・鍵の長さは128ビットから256ビットまで32ビット単位で選択可能。

・AESの規格ではその中から128,192,256ビットが採用。

・アルゴリズムの背景には数学的構造があり、計算を数式で表現可能。

・理論上、数学的に解読できる可能性を秘めている…が今のところ見つかっていない。

# SPN構造

AESの基本構造。

難しいので詳しいことはまたの機会に。

# どの対象暗号を使うべきか

- **DES**: 新しい用途には使うべきでない。現実的な時間内に解読が可能である。
- **トリプルDES**: これも新しい用途に使うべきでない。過去との互換性重視で使われる場面もあるが、今後AESにおきかわっていくだろう。
- **AES**: 現在使うならAESが良い。安全で高速、かつ幅広いプラットフォームで利用可能。また、世界中の研究者により検証が続けられているので、万一欠陥が見つかっても即座に世界中に知れ渡ることでしょう。
- **AESの最終候補者たち**: 最終選考まで残ったアルゴリズム達も使えないわけではない。AESのバックアップとして利用するのが良い。