情報ネットワークⅡ（情213）

【第3回】

暗号技術（1）

（教科書：第2章）

担当教員：長田智和

E-Mail: nagayan@ie.u-ryukyu.ac.jp

URL: http://n-lab.info/

（講義日：2017年10月19日）

第2章：暗号技術（1）

2.1 暗号技術の基礎

- 2.1.1 暗号技術の用語
  - 暗号とは「暗号化されたデータを見ても特別な知識なしでは、意味のある情報として読めないように変換する表記法（アルゴリズム）もしくは表記（データ）」のこと。
  - 暗号化＝暗号という処理（アルゴリズム）
  - 暗号文＝暗号化されたデータ
  - 平文＝暗号化される前のデータ
  - 復号＝暗号文から平文へ変換する処理
  - 暗号化鍵＝暗号化する際に使う特殊な情報
  - 復号鍵＝復号する際に使う特殊な情報
- （教科書p.29の図2.1を参照）

2.1 暗号技術の基礎

- 2.1.2 古典的な暗号化アルゴリズム
  - シーザー暗号
    - 平文をアルファベットの文字列とすると、決められた数だけ各文字をアルファベット順にずらす暗号方式。
    - 何文字ずらすかが「暗号化鍵」となる。
    - シーザー暗号の問題点
      - 最大25回総当たりすれば必ず平文を得られる。
- （教科書p.29の図2.2を参照）

2.1 暗号技術の基礎

2.1.2 古典的な暗号化アルゴリズム

バーナム暗号

- 一様分布性及び予測不可能性を満たした乱数列を使って暗号化する暗号方式。
- 平文をビット単位で毎回異なる乱数列を用いて暗号化する。
  →暗号化鍵を見つけることができない。
- バーナム暗号の問題点
  - 暗号化鍵の共有が難しい。（安全に暗号化鍵を共有する方法があれば、そもそも暗号は必要ない）
  - 暗号化鍵を再利用できない。（暗号文を復号するヒントを第三者に与えてしまうリスクがある）
  - 環境によっては真正乱数が利用できない。（物理乱数生成器などから乱数を得ることが望ましい）

2.1 暗号技術の基礎

- 2.1.3 共通鍵暗号化方式と公開鍵暗号化方式
  - 共通鍵暗号化方式
    - 暗号化鍵と復号鍵が同一である暗号化方式
    - 共通鍵暗号方式の問題点
      - 暗号化鍵を共有する組み合わせが多くなるほど、必要な暗号化鍵の数が増える。(n人だとn(n-1)/2個必要)
      - 暗号化鍵を安全に共有することが難しい。
  - 公開鍵暗号化方式
    - RSA(Rivest, Shamir, Ademanら開発者3名の頭文字)
    - 暗号化鍵(公開鍵)と復号鍵(秘密鍵)が別である暗号化方式
    - 秘密鍵から公開鍵は求まるが、その逆は困難である。
- (教科書p.33-34の図2.3-2.4を参照)

2.2 共通鍵暗号化技術

- 2.2.1 ストリーム暗号化とブロック暗号化
  - ブロック暗号化方式
    - データ撹拌部と鍵生成部から構成され、暗号化をブロック単位で行う。
  - ストリーム暗号化方式
    - 疑似乱数生成器が生成する乱数を使い、平文をビット単位で暗号化する。
- （教科書p.35の図2.5を参照）

2.2 共通鍵暗号化技術

- 2.2.2 代表的な共通鍵暗号化技術
  - DES（Data Encryption Standard）
    - IBMが開発。1977年にFIPS Publication 46として公開。
    - FIPS Publication 46-1（DES）~ 46-3（3DES）と改定。
    - 64ビット（56ビット（パリティビット除く））単位で暗号化。
    - DESの問題点
      - 短時間で暗号化鍵が発見される手法が登場した。

2.2 共通鍵暗号化技術

- 2.2.2 代表的な共通鍵暗号化技術
  - 3DES
    - DESを3回行う（暗号化→復号→暗号化）ことで暗号強度を高めた暗号化方式。
    - 鍵を2種類使う方法：1回目と3回目の暗号化で同じ鍵を使用（有効鍵長:56x2=112ビット）*非推奨
    - 鍵を3種類使う方法：1回目の暗号化と復号、2回目の暗号化の鍵が異なる。（有効鍵長:56x3=168ビット）
    - 3DESの問題点
      - 現時点はDES同様に非推奨でAES利用が推奨
- （教科書p.36の図2.6を参照）

2.2 共通鍵暗号化技術

2.2.2 代表的な共通鍵暗号化技術

AES（Advanced Encryption Standard）

- 2001年にFIPS Publication 197として公開。
- 128ビットのブロック暗号化方式。但し、鍵長は128ビット、192ビット、256ビットの3つがある。（192,256ビット長の鍵を使用する場合は、パディング（後述）を行う）
- DESに対して安全で高速とされている。
- 現在の標準的な共通鍵暗号化方式。

RC4

- 正式名称は「Rivest Cipher 4」で仕様は非公開。
- RC4と同等の「Arcfour」が広く一般に公開された。
- 鍵長40~256ビットの可変長。
- 現在はAESなどに置き換えられている。

2.2 共通鍵暗号化技術

- 排他的論理和（XOR）の性質
  - 同じ値を適用すると元の値に戻る性質がある。

    $C = P \oplus K$

    $P = C \oplus K = (P \oplus K) \oplus K$

    （例）

    $P = \underline{1101}, K = 1010$ とする。

    $C = 1101 \oplus 1010 = 0111$

    $P = C \oplus K = 0111 \oplus 1010 = \underline{1101}$
  - ほとんどのCPUがXORを高速に処理できる。
  - 暗号化の処理の一部で採用されている。

2.2 共通鍵暗号化技術

- 2.2.3 DESの仕組み
  - Feistel構造
    - データ撹拌部のラウンド処理段階で、各ラウンドの入力データ（64ビット）が半分に分けられ、XOR及び関数 f を用いて処理される構造。
    - Feistel構造は暗号文の逆変換で平文が得られる。

2.2 共通鍵暗号化技術

- 2.2.3 DESの仕組み
  - DESのデータ撹拌部
    - 入力平文を64ビット単位で分割する。
    - 初期転置を行い、32ビット(L_0,R_0)に分ける。
    - R_0はそのまま次の入力L_1($L_1=R_0$)とする。①
    - $R_1 = L_0 \oplus f(R_0, K_1)$とする。②
    - ①②を一般化した下式をn=16まで繰り返す。

$$\begin{cases} L_n = R_{n-1} \\ R_n = L_{n-1} \oplus f(R_{n-1}, K_n) \end{cases}$$

    - R_{16}とL_{16}を連結して1つのブロックとし、最終転置を行って得られた値が暗号文となる。
- （教科書p.38-42の図2.7-2.17を参照）

2.2 共通鍵暗号化技術

- 2.2.3 DESの仕組み
  - DESの鍵生成部
    - 入力鍵（64ビット）を転置PC-1で56ビットに変換する。
    - 左循環シフトしたC_1,D_1を連結し、転置PC-2で48ビットに変換する。
    - 同様に、n=2,…,16のサブ鍵を求める。
- （教科書p.42-43の図2.18-2.20,表2.4を参照）

2.2 共通鍵暗号化技術

- 2.2.4 パディング
  - 平文がブロック長に満たない場合、不足部分に特定のデータを埋め込む処理。
  - 埋め込んだ部分が正確に分かるために、パディング処理は暗号化側、復号側で一意である必要がある。
- （教科書p.44の表2.5を参照）

2.2 共通鍵暗号化技術

2.2.4 パディング

- PKCS#5パディングの例:

  ブロック長が満たない平文 = EM / EMの長さ = M

  EMに対するパディング文字列 = PS /

  パディング後のブロック = EB

  EB = EM | PS

  EB = EM | 01　(M mod 8 =7)

  EB = EM | 0202　(M mod 8 = 6)

  …

  EB = EM | 0808080808080808　(M mod 8 = 0)

  ※最終文字からパディング文字数が分かりEMが特定できる。

2.2 共通鍵暗号化技術

- 2.2.5 ブロックチェイニング
  - ブロック個別に暗号化処理を行う方法（ECB（Electric Code Book）モード）では、特定のブロックのみを解読対象とされる危険性がある。
  - 各ブロックに相関性を持たせて平文と暗号文を1対1になることを避けることで安全性を高める方法。
- （教科書p.46の図2.21を参照）

2.2 共通鍵暗号化技術

- 2.2.5 ブロックチェイニング
  - CBCモード
    - 1つ前のブロックとXORをとりその結果を暗号化するモード。
    - 最初のブロックは初期ベクトルとXORをとり暗号化する。
  - OFBモード
    - 初期ベクトルを次々と暗号化したものと平文のXORをとりその結果を暗号化するモード。
  - CTRモード
    - 初期ベクトルの代わりにブロック長カウンタを用いるモード。
    - 並列化が可能であるため高速処理が可能。
- （教科書p.47-48の図2.22-2.27を参照）

【次回予告】
第4回
暗号技術(2)
(第2章)

また来週！