情報理論第二（10）

ハミング符号・秘密鍵暗号

人間コミュニケーション学科
梶本　裕之
kajimoto@hc.uec.ac.jp

授業の流れ（予定）

第1週（10/ 8） 情報量と情報エントロピー
第2週（10/15） 相互情報量
第3週（10/22） 体育祭のため休講
第4週（10/29） 情報源符号化とデータ圧縮
第5週（11/ 5） ハフマン符号とデータ圧縮
第6週（11/12） 情報源符号化定理
第7週（11/19） 調布祭のため休講
第8週（11/26） 出張のため休講
第9週（12/ 3） マルコフ情報源モデル
第10週（12/10） 通信路のモデル化
第11週（12/17） 誤り検出・誤り訂正符号
第12週（12/24） 出張のため休講
第13週（ 1/ 7） 線形符号
第14週（ 1/14） センター試験だった…
第15週（ 1/21） ハミング符号・秘密鍵暗号
第16週（ 1/28） 公開鍵暗号
第17週（ 2/ 4） 出張のため休講

復習：線形符号

- 元の語 x ：kビット
- 符号語 y ：nビット

ある適当な生成行列 G（n行k列）を用いて

$$y = Gx$$

と符号化する方法を線形符号という

復習：生成行列の標準形

- 生成行列は，適当な変形により，
  以下のような標準形に変形できる

$$G = \begin{pmatrix} 1 & & 0 \\ & 1 & \\ 0 & \cdots & 1 \\ \hline & P & \end{pmatrix}$$

k 行

n-k 行

復習：標準形の生成行列による符号化

- 標準形をした生成行列:
- 元の語と付加されたパリティ部分が
  きれいに分かれた符号語を生成できる！

$$\begin{pmatrix} x \\ \hline Px \end{pmatrix} = \begin{pmatrix} 1 & & 0 \\ & 1 & \\ 0 & \cdots & 1 \\ \hline & P & \end{pmatrix} \begin{pmatrix} x \end{pmatrix}$$

符号語　　元の語

復習：検査行列

- 生成行列 G に対して，
  検査行列 H をつくることができる

$$G = \begin{pmatrix} 1 & & 0 \\ & 1 & \\ 0 & \cdots & 1 \\ \hline & P & \end{pmatrix}$$

k 列　k 行　n-k 行

$$H = \begin{pmatrix} -P & \begin{matrix} 1 & 0 \\ & \ddots & \\ 0 & & 1 \end{matrix} \end{pmatrix}$$

k 列　n-k 列　n-k 行

ただし2進数の場合 $-P=P$.

復習：検査行列の性質

- 生成行列Gによって作られた符号語をHに適用すると，0になる！！

$$H(Gx) = (HG)x = 0x = 0$$

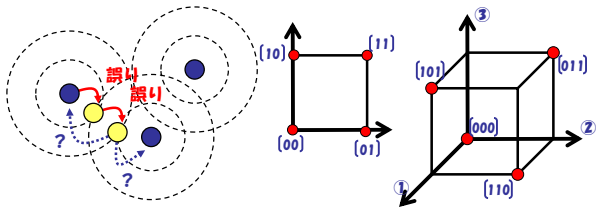
$$\begin{pmatrix} -P & \begin{matrix} 1 & 0 \\ & \ddots \\ 0 & 1 \end{matrix} \end{pmatrix} \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \\ \hline & P & \end{pmatrix} \begin{pmatrix} \\ x \\ \\ \end{pmatrix} = (-P+P)x$$

受信語にHをかけて0になるかどうかで
伝送中に誤りが生じたかどうか検査できる

線形符号の符号語間最小距離

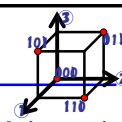
復習：何がしたかったのか

元の語：2bit(2次元)の情報．4通り
符号語：3bit(3次元)．8つの格子点の中の4点
⇒できた「すき間」＝誤り検出，誤り訂正のための冗長性
つまり，この「すき間」の最小の間隔が重要！！



この「最小すき間」はどうしたら計算可能か？

符号語間距離＝すき間



符号語y_1とy_2の間のハミング距離$d_H(y_1, y_2)$を考える

- $\{0,1\}$の有限体上では以下のようになる

$$d_H(y_1, y_2) = d_H(y_1 - y_2, 0)$$

∵各桁について考えると，

$(y_1, y_2)=(0,0), (1,1) \rightarrow d_H(0-0, 0) = d_H(1-1, 0) = d_H(0, 0) = 0$

$(y_1, y_2)=(0,1), (1,0) \rightarrow d_H(1-0, 0) = d_H(0-1, 0) = d_H(1, 0) = 1$

ハミング距離は，各桁同士で比較した時の「違うbitの個数」
だから，各桁で上式が成り立っていれば当然成り立つ．

線形符号における符号語間距離

- $d_H(y_1, y_2) = d_H(y_1 - y_2, 0)$

ここで，$y_1 = Gx_1$，$y_2 = Gx_2$とすると，

$y_1 - y_2 = G(x_1 - x_2)$　但し$x_1 - x_2 \in F_2^k$

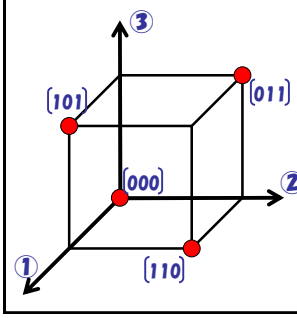
したがって$y_1 - y_2$も，ある元の語$(x_1 - x_2)$
から生成された符号語に必ずなっている．

何が言いたいか？

y_1, y_2が符号語なら，$y_1 - y_2$も符号語である．

線形符号における符号語間距離

y_1, y_2が符号語なら，$y_1 - y_2$も符号語である．



[1 0 1]-[0 1 1]=

[1 0 1]-[1 1 0]=

[1 1 0]-[0 1 1]=

線形符号における符号語間距離

・誤り検出・誤り訂正という観点から，
符号語間の最小距離 D を知りたい！！

$$D = \min d_H(y_1, y_2) = \min d_H(y_1 - y_2, 0)$$

$y_1 - y_2$ が符号語全体がつくる部分空間の中に必ず入っていることを考えれば，
$y = y_1 - y_2$ とおけて，

$$D = \min d_H(y, 0) \quad (y \neq 0)$$

13

線形符号における符号語間距離

$$D = \min d_H(y, 0) \quad (y \neq 0)$$

なにが言いたいか？

線形符号における符号語間最小距離は，

0ベクトルを除いた任意の符号語に含まれる要素1の個数の最小値となる

ことがわかる！

（符号語同士のハミング距離を求めなくても，
符号語と原点のハミング距離を求めれば充分）

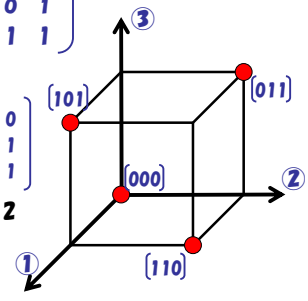
14

例

$$G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

・符号語は下記，

	0	1	1	0
	0	1	0	1
	0	0	1	1
1の数	0	2	2	2



符号語間の最小距離は2である⇒あっている

15

例

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

・符号語は下記，

	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1
	0	1	0	1	0	1	0	1
	0	1	1	1	1	0	0	0
	0	0	0	1	1	0	1	0
	0	0	1	0	1	1	0	0
1の数	0	3	3	4	3	4	4	3

符号語間の最小距離は3である
（だから2誤り検出，1誤り訂正可能）

16

線形符号の符号語間最小距離

実は検査行列Hを観察するだけでわかる！！

定理：

検査行列Hの列ベクトルのうち，任意のm−1個が1次独立ならば，Hによって検査される符号における符号語間最小距離はm以上である．

系：

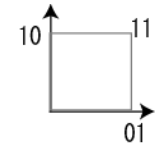
上記に加え，更にあるm個について1次従属となるならば，最小距離はmである．

17

復習：一次独立

集合中の任意の要素が，他の要素の和で書けないこと．

・(0,1)と(1,0)は一次独立
・(0,1)と(1,0)と(1,1)は一次従属
　(0,1) + (1,0) = (1,1)



別の表現：要素の線形和が0にならないこと．

・$a*(0,1) + b*(1,0)$は，$a,b \neq 0$で0とならない．
・$a*(0,1) + b*(1,0) + c*(1,1)$は，$(a,b,c)=(1,1,-1)$に対して0となる→一次従属．

18

例

$G =$ $\begin{pmatrix} 1&0&0&0&1&1 \\ 0&1&0&0&1&0 \\ 0&0&1&0&0&1 \\ 0&0&0&1&1&1 \end{pmatrix}$

- 下記の検査行列Hから，この符号における符号語間最小距離を求めよ．また，実際に符号語をいくつか生成し，求めた最小距離が正しいことを確認せよ．

$$H = \begin{pmatrix} 1&1&0&1&0&0 \\ 1&0&1&0&1&0 \\ 0&1&1&0&0&1 \end{pmatrix}$$

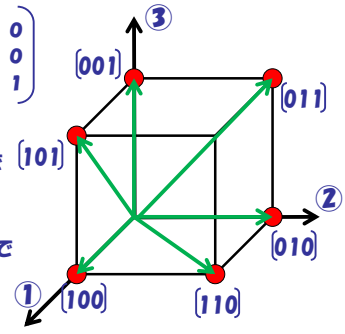
例

$$H = \begin{pmatrix} 1&1&0&1&0&0 \\ 1&0&1&0&1&0 \\ 0&1&1&0&0&1 \end{pmatrix}$$

任意の二つのベクトルは方向が重なっていないので一次独立である．

3つだと例えば
(100)+(010)=(110)なので
一次従属

よってm=3



定理の証明

$H=(h_1\ h_2 \ldots h_n)$（h_i は列ベクトル）とする．
Hは検査行列だから，誤りのない符号語
$y=(y_1 y_2 \ldots y_n)^T$について

$$h_1y_1 + h_2y_2 + \ldots + h_ny_n = 0$$

任意の m−1 個の列ベクトルが1次独立
→ 0以外のyで，常に

$$h_{i1}y_{i1} + h_{i2}y_{i2} + \ldots + h_{i(m-1)}y_{i(m-1)} \neq 0$$

定理の証明

(1) $h_1y_1 + h_2y_2 + \ldots + h_ny_n = 0$
(2) $h_{i1}y_{i1} + h_{i2}y_{i2} + \ldots + h_{i(m-1)}y_{i(m-1)} \neq 0$

(2)の元で(1)が成り立つためには，残りのy_iのうち少なくとも一つは1である必要アリ

任意の m−1 個のh_iに対してそうなるためには，
y_i はm個以上1である必要アリ．
→ 符号語間の最小距離は最低でもmとなる

線形符号における誤り検出・誤り訂正

誤り訂正とシンドローム

- 受信語yに検査行列Hをかけた結果Hyを，受信語yのシンドロームという
- シンドローム＝0なら，受信語に誤りはない（Gによって生成された符号語が届いている）
- シンドローム≠0：どこかに誤りあり！

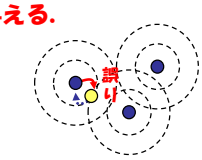
誤り訂正とシンドローム

- シンドローム≠0→どこかに誤りあり！
- 誤りベクトルを e とする．$y=y_{true}+e$
- $Hy = H(y_{true}+e)$
- $=Hy_{true}+He$
- $=He$
- より，シンドロームはエラーeに関する情報を，Heという形で間接的に与える．

25

誤り訂正とシンドローム

- シンドロームはエラーeに関する情報を，"He"という形で間接的に与える．



- ＜誤り訂正の手順＞
- (1) Heを与えるようなeを探す
- (2) そのようなeのうち，含まれる1の数が符号語間の最小距離の半分 よりも小さいものがあれば，そこに誤りが生じたとする
- (3)eが分かればyからy_{true}を作れる(反転)

26

例題

- 受信語として $(1\ 1\ 1\ 1\ 1\ 0)^T$ というベクトルを受け取った．この中に1ビットの誤りが含まれているものとする．下記の検査行列Hを用いて，この受信語の誤りを訂正せよ．
- （ヒント：1ビットの誤りということはeの候補は6通りだけ）

$$H=\begin{pmatrix}1&1&0&1&0&0\\1&0&1&0&1&0\\0&1&1&0&0&1\end{pmatrix}$$

27

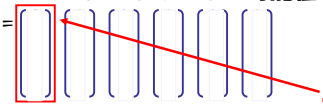
回答

$He=Hy=$

$$\begin{pmatrix}1&1&0&1&0&0\\1&0&1&0&1&0\\0&1&1&0&0&1\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\\1\\0\end{pmatrix}=\begin{pmatrix}1\\1\\0\end{pmatrix}$$

エラーが1bitとすると，可能性は

$e=$



エラーの発生箇所を特定できた！！

の6通り．このときHeは

$He=$

28

回答

確認（エラービットを反転させてみる）

$$Hy=\begin{pmatrix}1&1&0&1&0&0\\1&0&1&0&1&0\\0&1&1&0&0&1\end{pmatrix}\begin{pmatrix}0\\1\\1\\1\\1\\0\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix}$$

確かに検査行列によって0となった！
→正しい符号である！

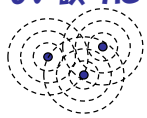
29

ハミング符号

30

復習：符号語間距離

- 一般に，各符号語間のハミング距離が最低でもkと保証されているならば，その符号は最低でも
  - k より少ない誤りについて検出可能
  - k/2 より少ない誤りについて訂正可能



ある符号体系の誤り訂正能力＝符号語間の最小距離で決まる

31

復習：線形符号の符号語間最小距離

実は検査行列Hをみるだけでわかる！！

$$H = \left(\begin{array}{c|c} -P & \begin{matrix} 1 & 0 \\ & \ddots \\ 0 & 1 \end{matrix} \end{array} \right) \begin{matrix} \\ \\ \} \text{n-k 行} \end{matrix}$$

（k 列，n-k 列）

定理：

検査行列Hの列ベクトルのうち，任意のm-1個が1次独立なら，符号語間最小距離はm以上．

系：

さらにあるm個について1次従属となるなら，符号語間最小距離はm．

（証明は前回を参照）

32

ハミング符号

定理：検査行列Hの列ベクトルのうち，任意のm-1個が1次独立なら，符号語間最小距離はm以上．

系：さらにあるm個について1次従属となるなら，最小距離はm．

検査行列Hが持つ上記の性質を利用して，符号語間の最小距離が3（＝2誤り検出・1誤り訂正可能）となる線形符号を生成する手法

33

ハミング符号の作り方

- まず適当な整数 m＞1 を用意する
- mビットの列ベクトルを，0ベクトルを除いて全てつくり，それを横に適当に並べて検査行列Hを作る

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

（2^m-m-1 列，m 列，m 行）

(m=3 の例)

ここが単位行列になるようにする

34

ハミング符号の作り方

- 検査行列Hから，対応する生成行列Gをつくれば出来上がり！

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

（2^m-m-1 列，2^m-m-1 行，m 行）

35

ハミング符号の符号長

- 元の語： 2^m-m-1 ビット
- 符号語： 2^m-1 ビット
  （冗長性として付加される分が m ビット）
  となる
  - 1ビット→3ビット
  - 4ビット→7ビット
  - 11ビット→15ビット，etc…

36

ハミング符号の符号語間最小距離

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

- 検査行列Hの列ベクトルはどの2つも互いに1次独立である → 最小距離≧3
- $(1\ 0\ 0\ \ldots\ 0)^T$, $(0\ 1\ 0\ \ldots\ 0)^T$, $(1\ 1\ 0\ \ldots\ 0)^T$ という3つの列ベクトルを取るとこれらは1次従属になっている → 最小距離＝3

37

ハミング符号を用いた誤り訂正

- ハミング符号の誤り訂正能力は 常に1
- シンドロームが0でなかった場合，

$$e = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \text{ or } \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \text{ or } \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \cdots \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

の 2^m-1 個の誤りベクトルの候補から
$Hy = He$ となるものを探せばよい
（Hの列ベクトルからシンドロームに一致するものを探せばよい）

38

小レポート

- m＝2の場合のハミング符号は，これまでに出てきた

  0→000，1→111

  という3重化による誤り訂正符号になることを確かめよ．

- m＝4の場合についてハミング符号を作成せよ．また，その生成行列を用いて生成した任意の符号語に1ビットの誤りを加え，それを検査行列を用いて誤り訂正してみよ．

39

暗号の基礎

暗号（cryptography）

- やっぱり符号化の一種
  （元の語⇔符号語の変換操作の一種）
- 情報をやりとりしたり保存したりする際に，第3者には復号化（or 符号化）が困難な方法を用いることによって，情報の漏洩を防いだり，メッセージの作成者の同一性（identity）を認証したりする技術

41

情報源・通信路符号化と暗号

- 情報源符号化：
  データサイズを最小化する符号化
- 通信路符号化：
  ノイズの影響を最小化する符号化
- 暗号：
  第3者による盗聴や介入の危険性を最小化する符号化（事実上不可能にする or したい）

42

情報源・通信路符号化と暗号

- 情報源符号化：
  表現効率
- 通信路符号化：
  通信の正確さ
- 暗号：
  安全性・秘匿性

43

身近にある暗号技術

- 昔（2000 B.C. 以前にまでさかのぼる）
  - 軍事目的の通信
  - 国家機密の保持

- 現在：上記に加えて
  - コンピュータのアカウントやファイルの管理
  - ネット上の通信や電子商取引における守秘・認証（SSL，PGP，クレジットカード認証）
  - 電子政府，企業，医療現場等における機密保持・個人情報管理

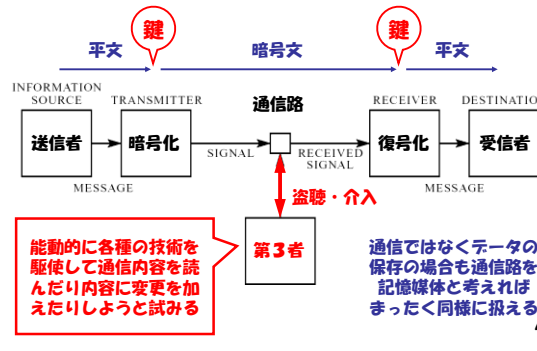
44

注意！

- 暗号理論は考案者と攻撃者との間のイタチゴッコなので**技術の変遷が非常に速い**
- 授業でやる内容は初歩の初歩，
  暗号を更に勉強 or 応用したい人は**最新の技術動向を常に自分でフォローすること**
- 現在実用化されている暗号の安全性は**数学的に証明されていない**ことに注意

45

暗号系とは



46

用語の定義

- 平文：　秘密を守りたい元のメッセージ
- 暗号化：守秘用途の特殊な符号（暗号）で平文を符号化する作業
- 暗号文：暗号化された文（符号語）
- 復号化：正当な受信者が鍵を用いて暗号文を平文にもどす作業
- 鍵：　暗号化・復号化を行う上で重要となる，用いられた暗号に関する部分的知識
- 解読：　正当でない第3者が暗号文から平文を求める行為

47

暗号に用いられる符号化手法

- 情報源符号化や通信路符号化では，元の語と符号語の対応関係はベストのものが**万人共通に1つ**あればよかった
- **暗号系では，異なる鍵を持つ各個人に対して異なる元の語⇔符号語の対応関係を自動的に生成できなければいけないので，それらの対応関係は計算可能な関数を用いて表されることが多い**
  （鍵はその関数を計算する上で必要な情報を与える）

48

暗号系を定義する5成分

- 平文空間　P：平文の集合
- 暗号文空間　C：暗号文の集合
- 鍵空間　K：鍵の集合
- 暗号化関数　$E_k : P \to C$　$(k \in K)$
- 復号化関数　$D_k : C \to P$　$(k \in K)$

49

暗号の種類

- 対称暗号系，非対称暗号系
- 秘密鍵暗号系，公開鍵暗号系

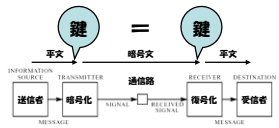
（実用上 対称⇔秘密鍵，非対称⇔公開鍵 である）

50

対称暗号系，非対称暗号系

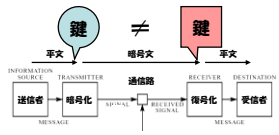
対称暗号系：

暗号化と復号化で用いられる鍵が同じ，または**片方から片方が簡単に計算**できる暗号系



非対称暗号系：

暗号化と復号化で用いられる鍵が異なる，または**片方から片方が容易に計算できない**暗号系

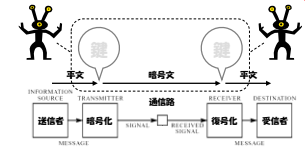


51

秘密鍵暗号系，公開鍵暗号系

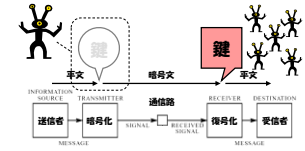
秘密鍵暗号系：

鍵が当事者間だけで秘匿されている暗号系（共通鍵ともいう）



公開鍵暗号系：

鍵のどちらか一方が一般に公開されている暗号系



52

秘密鍵暗号系

秘密鍵暗号系の特徴

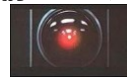
- 暗号化・復号化の計算が一般に簡単かつ高速に実行できる
- 送信者と受信者の間で**鍵を事前に安全な手段を用いて共有し，秘密裏に保管しなければならない**

→ **鍵の共有や保管における情報漏洩のリスクが常にある**

54

古典的な秘密鍵暗号系の例

- 換字（かえじ）方式
  - 文字の種類をあるルールで置き換える方法
    例：アルファベットの順序で1個前にずらす
    IBM → HAL



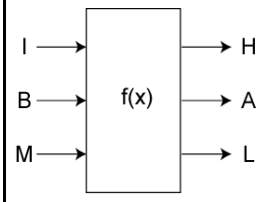
- 転置方式
  - 文字の位置をあるルールで入れ替える方法
    例：偶数位置→奇数位置の順に並べ替える
    INFORMATION → NOMTOIFRAIN

いずれもルールの一部を鍵とする（ずらす個数など）

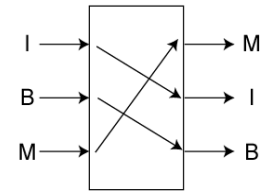
55

換字方式と転置方式

- 換字方式



転置方式



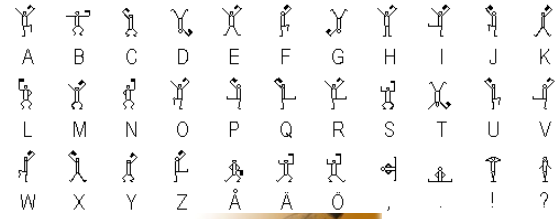
56

転置方式　例　スキュタレー

- 紀元前5世紀スパルタで使用
- 皮ひも上に一見ランダムな文字
- ⇒ある直径の棒に巻きつけると平文が現れる

57

換字方式　例1：踊る人形



58

換字方式　例2：シーザー暗号

- 各文字をアルファベット順にk個ずらした暗号（k = 0～26）
- 元の文字を x=0～26 とすると，

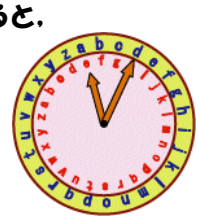
暗号化関数

$x' = E(x) = x + k \bmod 27$

復号化関数

$x = D(x') = x' - k \bmod 27$

（スペース込みで27文字）



- 27通り：力ずくでも解読できる
- 統計的分布を見ればすぐ分かる．

59

例題

- 以下は単純なシーザー暗号によってつくられた暗号文である．使用されているアルファベットは A～Z および空白である．
- 解読を試みよ．

QCVJWILXVVCWRLJBRXW

60

解答

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z _
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

QCVJWILXVVCWRLJBRXW

+9 -9

61

ブロック暗号

- ある固定された長さの文字列（ブロック）を同じ長さの文字列に置換する方法
  - 文字列の長さnをブロック長と呼ぶ
  - 用いられるアルファベットを $\Sigma = \{a, b, \ldots\}$ とすると，ブロック暗号の平文空間と暗号文空間は，ともに Σ^n となる
  - ブロック長自体も鍵の構成要素
    - 換字方式はブロック長＝1のブロック暗号
    - 転置方式はブロック長＝平文長のブロック暗号

62

例

- ブロック長を5とし，各ブロック内では各文字をアルファベットの順序で1個前にずらした後，偶数位置→奇数位置の順に並べ替えるとする

INFORMATION（平文）
→ INFOR:MATIO:N
→ HMENQ:L SHN:M
→ MNHEQ: HLSN:M
→ MNHEQ HLSNM（暗号文）

63

例題

- ブロック長を6とし，各ブロック内では各文字をアルファベットの順序で1個後にずらした後，偶数位置→奇数位置の順に並べ替える．という暗号方式を考えた．以下の暗号文に対応する平文を求めよ．

PASZVBAIAFUFPFCDEAFLSSBF

64

解答

PASZVB | AIAFUF | PFCDEA | FLSSBF

65

例2：アフィン線形ブロック暗号

アフィン暗号：シーザー暗号の一般化

$x' = E(x) = a\,x + b \bmod m$

$x = D(x') = a'\,(x' - b) \bmod m$

アルファベットをaずつ飛ばしたのち，bずらす．

- m ：アルファベットの総数．
- a は m と互いに素な自然数 $(a<m)$
- a $a' = 1 \bmod m$ （を満たす整数a'を見つける）

66

アフィン暗号の例

$x' = E(x) = a\,x + b \bmod m$

$x = D(x') = a'(x' - b) \bmod m$

アルファベット：{ABCDE}の５文字 (m=5)

a=2，b=1とする．

この場合，a'=3 (∵2x3=6=1(mod5))

平文 " A C E"はどう暗号化，復号化されるか？

アフィン暗号の例

{A B C D E}⇒{0 1 2 3 4}

平文：" A C E"を暗号化する

"A C E"
↓
0 2 4
↓ (x2)
0 4 8
↓ (+1)
1 5 9
↓ (mod5)
1 0 4
↓
"B A E"

"B A E"
↓
1 0 4
↓ (-1)
0 -1 3
↓ (×3)
0 -3 9
↓ (mod5)
0 2 4
↓
"A C E"

例２：アフィン線形ブロック暗号

アフィン線形ブロック暗号

(v は１つのブロックの内容を表すベクトル)

$v' = E(v) = A\,v + b \bmod m$

$v = D(v') = A'(v' - b) \bmod m$

(ただし $A\,A' = I \bmod m$)

過去盛んに用いられたが，平文と暗号文の組がある程度手に入ると代数的に簡単に秘密鍵が求まってしまう

暗号系が満たすべき安全性

暗号の仕様は公開されている

- 従来，暗号系はその仕様を秘密にしたまま使用されることが多かった
- 情報通信技術の発達や社会的要請の変化にともない，暗号系の仕様も一般に公開されるようになってきた
  - 工業製品としての普及
  - 第３者による品質検証の必要性
- 詳しい仕組みがわかっていても破れないような強固な暗号系を作る必要がある！

鍵の全数探索に対する安全性

- コンピュータを用いて力ずくで全ての鍵を試してみれば，どんな暗号系も必ず解読されてしまう

⇒ 現実的な時間のうちには全探索が絶対に終わらないような，非常に大きな鍵空間を持つことが必須！！

(現在では最低でも 2^{128} 程度の鍵空間が必要であるというのが業界のある種の共通認識になっている)

ショートカット型解読に対する安全性

- 各暗号化方式の具体的な仕組みに着目し，
  - 平文・暗号文の統計的な分布を計測したり，
  - 微妙に異なる平文がそれぞれどのように暗号化されるかを調べたり（差分解読），
  - 平文と暗号文との間に何らかの線形な関係を仮定したり（線形解読）

  して効率的に鍵を割り出そうとする攻撃に対しても，安全でなければならない
  （そのような全数探索よりも効率的な解読方法が見つかったら，その暗号は「破られた」とみなされる）

73

古典的な暗号系の弱点

- 平文は自然言語なので，ある種の統計的偏りをもっている（文字の出現頻度分布，連続する文字間の遷移確率など）
- 古典的な暗号化方式はこれらの性質がある程度暗号文の分布にも現れてしまう

→大量の暗号文を入手してその統計的性質を解析すれば，暗号文と平文との対応関係を推定できてしまう

74

安全性向上のための手法

安全性向上のための手法いろいろ

- 文脈依存型暗号
- 情報理論的に安全な暗号
- 多重暗号化

76

文脈依存型暗号（これも古典の部類に入りますが…）

- 鍵だけでなく，先行する平文の中身や文字の出現位置に依存して暗号化する手法
  - 例：シーザー暗号を i 番目の文字について以下のように拡張する

    暗号化 $x_i' = E(x_i) = x_i + k + x_{i-1} \bmod 27$

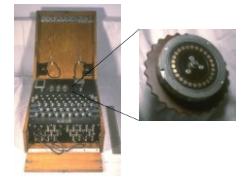
    復号化 $x_i = D(x_i') = x_i' - k - x_{i-1} \bmod 27$

    （スペース込みで27文字：ただし $x_0 = 0$ とする）
- 暗号文を一部分だけ書き換えることが難しくなる

77

具体例：エニグマ暗号

- 第２次大戦中にドイツ軍が使用した暗号
  - 簡単な携帯暗号機で暗号化・復号化できた
  - ３つのローターで文字位置に依存した暗号系を実現
  - Turing の開発した Turing Bombe によって解読されたことが連合軍の勝利に多大な貢献を及ぼした



78

情報理論的に安全な暗号

- Vernam の１回使い捨て暗号 (1917)
  nビット平文を１回送信するたびに一様分布の中からランダムに生成したnビットの鍵との XOR をとって暗号化する方法

平文：P	0 0 0 1 0
鍵：K	1 0 0 1 1
暗号文：C＝K⊕P	1 0 0 0 1
復号化：P＝C⊕K	0 0 0 1 0

79

Vernam の１回使い捨て暗号

- 送信文と同じ長さの秘密鍵を受信者へ別の手段で送らなければいけないので非効率
- しかし完全守秘性をもつ
  （平文と暗号文の間に相互情報量がないため暗号文だけいくら盗聴しても平文に関する情報を得ることができない）
  ことが Shannon によって証明された

80

多重暗号化

- 鍵空間を広げて解読しにくくするため，幾つかの異なる鍵によるブロック暗号化を何重にも適用する方法

例：E－D－E ３重暗号化

$$c = E_3(D_2(E_1(p)))$$

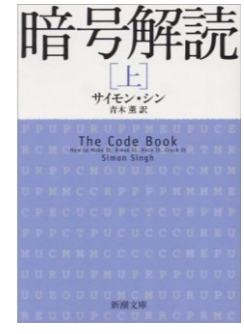
元の語を鍵１を用いて暗号化し，それを鍵２を用いて復号化し（当然平文ではないものになる），それを更に鍵３を用いて暗号化する

81

参考

- 暗号解読（上，下）
- 新潮文庫 シ 37－2
- サイモン・シン

- 下手な教科書よりずっとわかりやすいかも．



82