# 暗号技術の数理としくみ
# — メールでじゃんけん？する方法 —

小川朋宏

電気通信大学 第 1 回

平成 22 年度 ちょうふ市内・近隣大学等公開講座
9 月 17 日

# 電気通信大学の公開講座

## 情報ネットワークと情報保護技術

**社会基盤となった情報ネットワークの仕組みや**
**情報セキュリティーについて解説します.**

- **第 1 回** (9/17)：**小川朋宏　（おがわ ともひろ）**
  **暗号技術の数理としくみ**
  **— メールでじゃんけん？する方法 —**
- **第 2 回** (10/22)：**大坐畠智　（おおざはた さとし）**
  P2P **ネットワークとセキュリティ**
  **—しくみと使い方—**
- **第 3 回** (11/12)：**入江英嗣　（いりえ ひでつぐ）**
  **ネットワーク社会を支えるプロセッサ達**

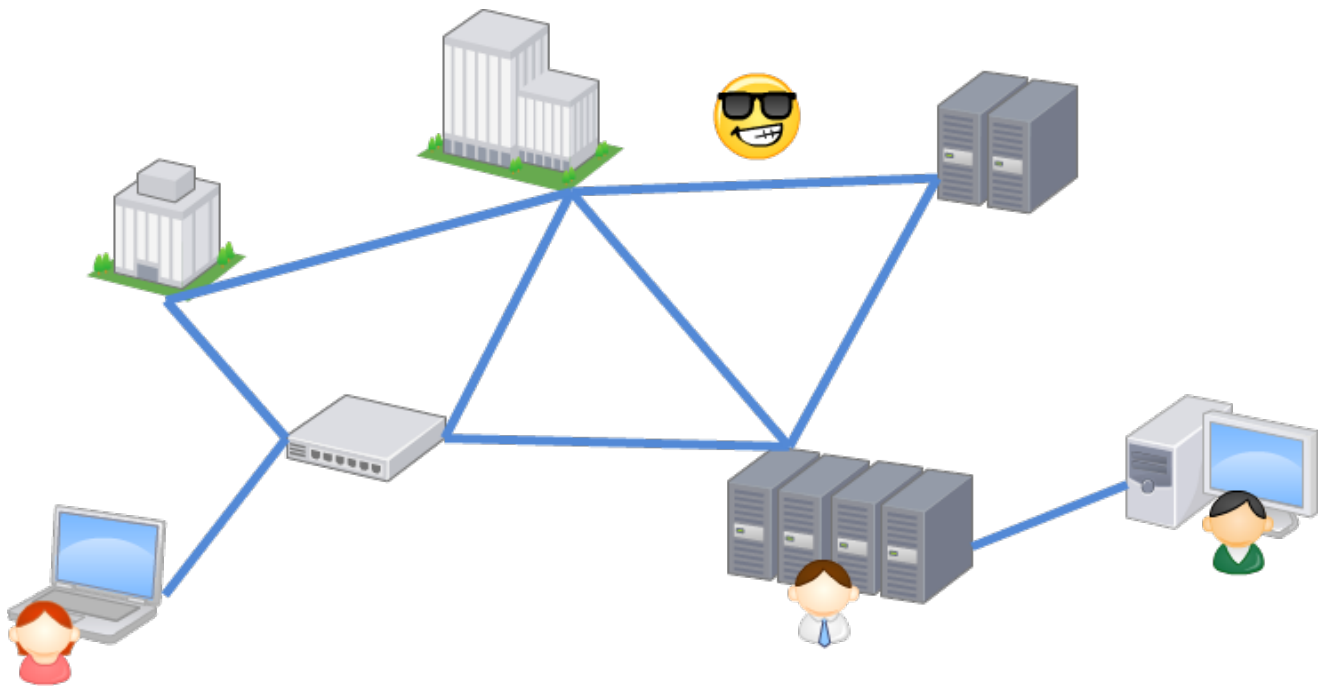
# 自己紹介

- **小川朋宏**
- **電気通信大学 大学院情報システム学研究科**
  **情報ネットワークシステム学専攻 准教授**
- **研究分野**
  - **情報理論**
  - **量子情報理論，量子暗号**
  - **情報理論的安全性に基づく暗号理論**
  - **ネットワーク・コーディング**
- **メールアドレス：** ogawa **あっと** is.uec.ac.jp
- **ホームページ：** http://www.quest.is.uec.ac.jp/ogawa/
- **大学院生募集中（社会人の方も歓迎）**
  http://www.is.uec.ac.jp/

# インターネットのしくみ

## インターネット

- 米国国防総省のネットワーク ARPANet (1969) が始まり
- 政府，企業のネットワークが相互に結合したもの
- バケツリレー方式で，途中の配送経路は様々
  - 配送の遅延が生じる
  - 途中で容易に読み取ることができる
  - 途中で内容が書き換えられる可能性もある



# 課題：メールでじゃんけんできるか？

**後出しの問題を解決しないといけない**

- **電子メールの配達には時間差が生じる**
- **同時に配達される保証はない（分単位のズレは常に起きる）**

**この講義ではメールで “コイン投げ” をする方法を解説する**

- **オリジナル論文：**M. Blum (**ブラム**), Coin Flipping by Telephon (**電話でのコイン投げ**), 1981.
- **論文の冒頭で以下のように書かれている**

**アリスとボブはコイン投げを電話越しで行いたい．**
**（二人は最近離婚をして別々の町に住んでいるが，**
**車をどちらが引き取るかを決めたい．）**

**メールで “じゃんけん” の本質はメールで “コイン投げ” と同じ**

# 講義の概要

## 目次

- 課題１：生活に密着した暗号技術
- 課題２：暗号の歴史
- 課題３：共通鍵暗号
- 課題４：公開鍵暗号
- 課題５：メールでじゃんけん（コイン投げ）

# 課題１
# 生活に密着した暗号技術

# 生活に密着した暗号技術

**暗号技術が日常生活を便利にしていることを学びましょう**
**暗号技術は情報ネットワーク社会のインフラ（社会基盤）**

- **インターネットバンキング，インターネットショッピング**
- **住民基本台帳カード**
- **電子マネー：**
  Suica (JR **東日本**), PASMO (**株パスモ**), Edy (**ビットワレット**)
  **など** [FeliCa (SONY) **系統**]
- IC **キャッシュカード**
- **無線** LAN **通信の暗号化**
- **有料デジタル放送，音楽配信**
- **携帯電話の** SIM **カード** (NTT **ドコモ** FOMA **など**), **認証機能**

# インターネットとSSL

- SSL (Secure Socket Layer) ：インターネット上で情報を暗号化して送受信するプロトコル（通信の手順）
- 盗聴や “なりすまし” を防ぐ
- 口座番号やクレジット番号を安全に送信する手段として使われることが多い
- Web ブラウザでは https://www.··· で確認できる


- 通信路の安全性を保証しているだけなことに注意

# IC カード：住民基本台帳カード，フェリカなど

**住民基本台帳カード（住基カード）**

- **公開鍵暗号による電子署名を生成可能：印鑑に相当**
- **公開鍵の正当性について公的機関による証明書を格納できる：印鑑証明に相当**
- **国税電子申告納税システム** (e-TAX) **で利用可能**
- **その他付加的サービス**

**フェリカ** (Felica, SONY)

- Suica **など国内の電子マネーの多くで採用されている**
- **近年，多くの携帯電話にも搭載**
- **スタンダードな共有鍵暗号：**DES**，トリプル** DES

# 課題２
# 暗号の歴史

# 暗号の歴史

- 古代の暗号
  - 換字式暗号：シーザー暗号など
  - 転置式暗号：レールフェンス暗号，スキュタレー暗号
  - 統計的な分析による解読方法が分かっている
- 戦争と暗号
  - ドイツのエニグマ，日本の紫暗号 $\rightarrow$ 解読されていた
- 商用暗号 (1970 年代〜)
  - 暗号のアルゴリズム（手続き）は公開，標準化する
  - 暗号の鍵を秘密にすることで通信の秘密を守る
  - 米国標準局による標準化
    DES (Data Encryption Standard, 1977)
    AES (Advanced Encryption Standard，2001)
- 公開鍵暗号
  - ディフィー・ヘルマン・マルクル鍵共有方式 (1976)
  - RSA 暗号の発明 (1978)

# シーザー暗号：換字式暗号の例

カエサル（シーザー，紀元前 100 年頃〜紀元前 44 年）が使った暗号

- 文字の順番をずらして置き換える（以下の例は３）

ABCDE FGHIJ KLMNO PQRST UVWXYZ
DEFGH IJKLM NOPQR STUVW XYZABC

- 暗号化の例

平文（ひらぶん）：HELLO $\longrightarrow$ 暗号文：KHOOR

- 復号： 暗号文を元に戻には，上の表を逆にたどればよい

用語について

- 平文（ひらぶん）：暗号化される前のメッセージ
- 換字（かえじ）式暗号：文字を別の文字で置き換える暗号方式
- 転置（てんち）式暗号：文字の並びを置き換える暗号方式

# レールフェンス暗号：転置式暗号の例

- 平文

  ちようふしこうかいこうざ

- 暗号化：

  (1) 1 文字おきに上の段，下の段に書く

  | ち | | う | | し | | う | | い | | う | |
  |---|---|---|---|---|---|---|---|---|---|---|---|
  | | よ | | ふ | | こ | | か | | こ | | ざ |

  (2) 上の段と下の段をつなげる

  ちうしういうよふこかこざ

- 復号：逆の手続きを行えばよい

  ちうしういう｜よふこかこざ → ちうしういう<br>よふこかこざ<br>→ ちようふしこうかいこうざ

- 段数を 3 段，4 段と変化を付けることができる

# 暗号は鍵が命

- **暗号の安全性は鍵を秘密にすることにかかっている**
- 暗号化手続きを秘密にしても，いずれ暴かれる
- アルゴリズム公開により商用通信，大量生産ができる
- **暗号化の鍵：**
  - シーザー暗号の鍵 → シフトする文字数（ずらす文字数）
  - レールフェンス暗号の鍵 → 段数

送信者アリス

受信者ボブ

平文 “HELLO”

平文 “HELLO”

共有鍵　　鍵を事前に共有　　共有鍵

⇓暗号手続き

復号手続き⇑

暗号文

公開通信路

盗聴者イブ

暗号文

# 暗号アルゴリズムの安全性

- シーザー暗号は鍵（= シフトする数）の候補が 25 個しかない
- 25 通りすべて試してみれば，すぐに解読できる
  （**総当り法，ブルートフォースアタック**）
- レールフェンス暗号も同様に安全ではない

**文字の置き換えをもっと複雑にしたら？**

ABCDE FGHIJ KLMNO PQRST UVWXYZ
TIEFH DJKLM GOQRC ASUVW YZBPXN

**置き換えの総数は** $26 \times 25 \times 24 \times \cdots \times 2 \times 1$ **通り！**

- **出現頻度の分析や，文法の知識を使うことで解読可能！**
- 例：英語では出現頻度の順に，E は 12.7%，T は 9.1%，$\cdots$

暗号文で，１番多く現れるのは E，２番に多いのは T，$\cdots$

のように解読されてしまう

# 課題３
# 共通鍵暗号

# コンピュータと暗号

- **現在では暗号化，復号にコンピュータが使われる**
- **コンピュータでは文字を $0$ と $1$ の並び（二進数）で表現する**
- **アスキー** (ASCII) **コード**

| **文字** | A | B | C | $\cdots$ | Z |
|---|---|---|---|---|---|
| **二進数** | 1100001 | 1100010 | 1100011 | $\cdots$ | 1111010 |
| **十進数** | 97 | 98 | 99 | $\cdots$ | 122 |

# ビットの足し算：オセロゲーム

- オセロゲームで “そのまま” $= 0$，“裏返す” $= 1$ で表すと，

$$
\begin{array}{rcrcr}
\text{そのまま}\,(0) & + & \text{そのまま}\,(0) & = & \text{そのまま}\,(0) \\
\text{そのまま}\,(0) & + & \text{裏返す}\,(1) & = & \text{裏返す}\,(1) \\
\text{裏返す}\,(1) & + & \text{そのまま}\,(0) & = & \text{裏返す}\,(1) \\
\text{裏返す}\,(1) & + & \text{裏返す}\,(1) & = & \text{そのまま}\,(0)
\end{array}
$$

- 二つの事柄の区別をビット (bit) という
- コンピュータでは数をビット（$0$ か $1$）の並びで表す

## ビットの足し算

- $0 \oplus 0 = 0$
- $0 \oplus 1 = 1$
- $1 \oplus 0 = 1$
- $1 \oplus 1 = 0$ （２回裏返す＝そのまま）

# ビット列の足し算

- ビット列＝ビットの並び
- ビット列の足し算では，各桁（けた）を足す

$$
\begin{array}{lcccccccc}
 & a & = & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
\oplus) & b & = & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
\hline
c = & a \oplus b & = & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

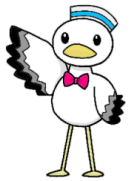
## ビット列 $a$ に，ビット列 $b$ を二回足すと元に戻る

$$
\begin{aligned}
a \oplus b \oplus b &= a \oplus \underbrace{(b \oplus b)} \\
&\qquad\qquad \text{二回そのままか，二回裏返すか} \\
&= a \oplus \underbrace{0000000000} \\
&\qquad\qquad \text{全部そのまま} \\
&= a
\end{aligned}
$$

# ワンタイムパッド (one-time pad)

**送信者アリス**

**受信者ボブ**



| **送信メッセージ** “HELLO” | **受信メッセージ** “HELLO” |
|---|---|
| $a = 11010000 \cdots$ | $a = 11010000 \cdots$ |

| **共有鍵** | **ランダムなビット列を事前に共有** | **共有鍵** |
|---|---|---|
| $b = 01101010 \cdots$ | | $b = 01101010 \cdots$ |

$\Downarrow \oplus$ **暗号化**

**復号** $\oplus \Uparrow$

| **暗号文** | **公開通信路** | **暗号文** |
|---|---|---|
| $c = 10111010 \cdots$ | $\longrightarrow$ | $c = 10111010 \cdots$ |

**盗聴者 イブ** $\searrow$ 

**暗号文を盗聴** $c = 10111010 \cdots$

**（メッセージをランダムに反転したものなので，元が分からない）**

# ワンタイムパッドの手順

- **バーナム暗号** (Vernam, 1926) **とも言う**

## ワンタイムパッドの手順

1. **送信者と送信者が事前に**$b$ **を共有**
2. **送信者（アリス）はメッセージを表すビット列を**$a$ **として**
   $a$ **を**$c = a \oplus b$ **で暗号化**
3. **受信者（ボブ）は**$c \oplus b = (a \oplus b) \oplus b = a$ **で復号**

- **共有するランダムなビット列を共有鍵または秘密鍵という**
- **暗号文は元のメッセージをランダムに反転したもの**
  **→ 盗聴者（イブ）には元のメッセージは分からない**
- **同じ乱数を二回以上使ってはいけない**
  **同じ乱数を何度も使うと安全ではなくなる**

# ワンタイムパッドの安全性と問題点

**利点**

- **無条件に安全なことが，数学的に証明されている**
  **［シャノン (Shannon), 1948］**
- **盗聴者にとって，あらゆる単語がメッセージの候補になる**
  (APPLE, BEACH , $\cdots$, HAPPY, HELLO, $\cdots$, ZEBRA, $\cdots$)

**欠点**

- **鍵共有の問題：事前に直接会って，メッセージと同じ長さの乱数を共有しないといけない**
- **乱数を一回しか使用できないのでコストがかかる**

$\Downarrow$

**暗号の歴史上，鍵共有が最も大変な課題だった**

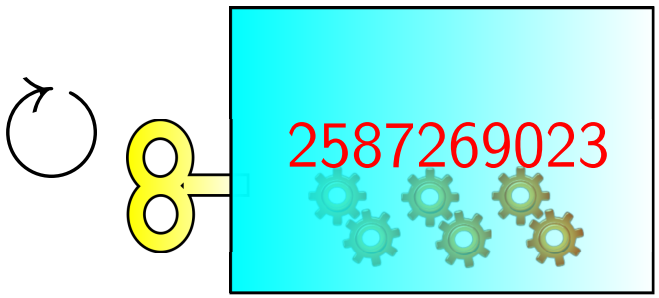
$\Downarrow$

**公開鍵暗号の発明 (1976, ディフィー・ヘルマン・マルクル)**

# 課題４
# 公開鍵暗号

# 暗号の基本的パーツ：一方向性関数

仮想的な機械 “一方向カウンタ” を考えましょう




- ある回数ネジを回すと回数に応じた数字が表示される
- 表示された数字からネジを回した回数を調べることは不可能
  原理的には機械を調べれば可能だが，
  複雑で宇宙の年齢ぐらい時間がかかる
- 公開されていて誰でも入手可能

実際には

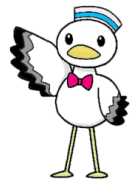
- 数学の整数論，代数学を用いるとコンピュータで実現できる
- 正式な名称は “一方向性関数”

# ディッフィー・ヘルマン・マルクル鍵共有

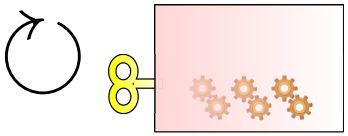
- Diffie-Hellman-Merkle (1976), **鍵共有のジレンマを初めて解決**
- **一方向カウンタを２台使用する**

**アリス**

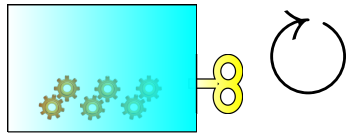
**ボブ**



**ランダムな回数 ($x$ 回) 回す**

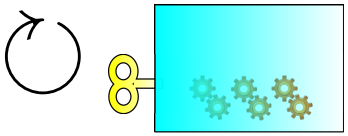


**ランダムな回数 ($y$ 回) 回す**

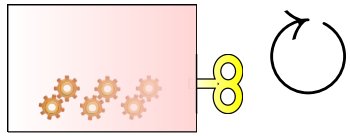


**公開通信路を通して交換**

**同じ回数 ($x$ 回) 回す** $\longleftrightarrow$ **同じ回数 ($y$ 回) 回す**



**盗聴者 イブ**



- **アリスとボブがネジを回した合計は $x + y$ 回で同じなので，二人の手元に “同一の数字 $=$ 鍵” が表示される**
- **イブには公開通信路に流れた途中の数字しか分からない**

# 一方向性関数の数理：合同式（時計の計算）

## 時計の計算

10 時に東京駅発の新幹線 + 大阪駅までの所要時間は 3 時間
= 到着時刻は 13 時 =1 時


- この計算を $10+3=1 \pmod{12}$ と書く
- 12 離れた数（時計の針が一周）は同じとみなす

$$1=13=25=\cdots \pmod{12}$$

- 読み方：mod モッド

# 合同式は余りの演算

- 合同式では数を 12 で割った余りとみなす
  - $1 \div 12 = 0$ (余り 1)
  - $13 \div 12 = 1$ (余り 1)
  - $25 \div 12 = 2$ (余り 1) これらより $1 = 13 = 25 = \cdots \pmod{12}$
- 合同式の等号：引き算して 12 で割り切れること
  - 13-1=12 は 12 で割り切れるので $1 = 13 \pmod{12}$
  - 25-1=24 は 12 で割り切れるので $1 = 25 \pmod{12}$
- 12 に限らず合同式を考えることができる

$$0 = 5,\ 1 = 6,\ 2 = 7,\ 3 = 8,\ 4 = 9 \pmod{5}$$

1 から 5 までの目盛が書いてある円盤の針の位置

- 足し算，引き算，かけ算もできる

$$1 + 2 = 6 + 7 = 3 = 13 \pmod{5}$$

$$4 - 1 = 9 - 6 = 3 \pmod{5}$$

$$2 \times 3 = 7 \times 8 = 6 = 56 = 1 \pmod{5}$$

# 指数関数

- **指数関数** $y = a^x$
- $a$ **を** $x$ **回かけ算する**

$$y = a^x = \underbrace{a \times a \times \cdots \times a}_{x \text{ 回}}$$

- **読み方：** $a$ **の** $x$ **乗**

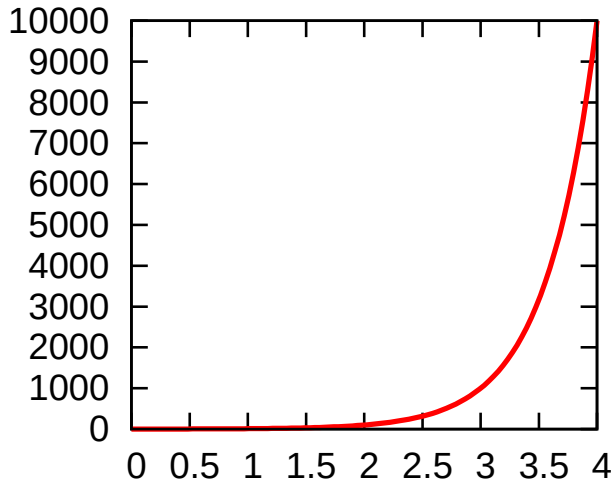
- **例：** $y = 10^x \ (a = 10)$

$(x = 1)\ 10^1 = 10$

$(x = 2)\ 10^2 = 10 \times 10 = 100$

$(x = 3)\ 10^3 = 10 \times 10 \times 10 = 1000$

$y = 10^x$ **のグラフ**



# 対数関数

- **対数関数** $y = \log_a x$
- **指数関数の逆関数** $[y = \log_a x \iff x = a^y]$
- $a$ **を** $y$ **回かけ算して** $x$ **になるときの** $y$ **は？**

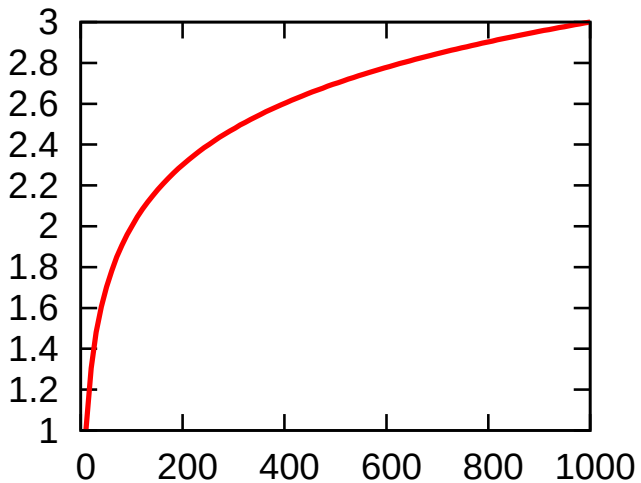
$$x = a^y = \underbrace{a \times a \times \cdots \times a}_{y \text{ 回}}$$

- **読み方：ログ** $a$ $x$

- **例：** $y = \log_{10} x \ (a = 10)$
  $\iff x = 10^y$

$(x = 10)$ $10^1 = 10$ **より** $y = 1$
$(x = 100)$ $10^2 = 100$ **より** $y = 2$
$(x = 1000)$ $10^3 = 1000$ **より** $y = 3$

$y = \log_{10} x$ **のグラフ**



# 離散対数問題

## 離散対数問題

$y$ が分かっているとき，$y = a^x \pmod p$ を満たす $x$ を求める

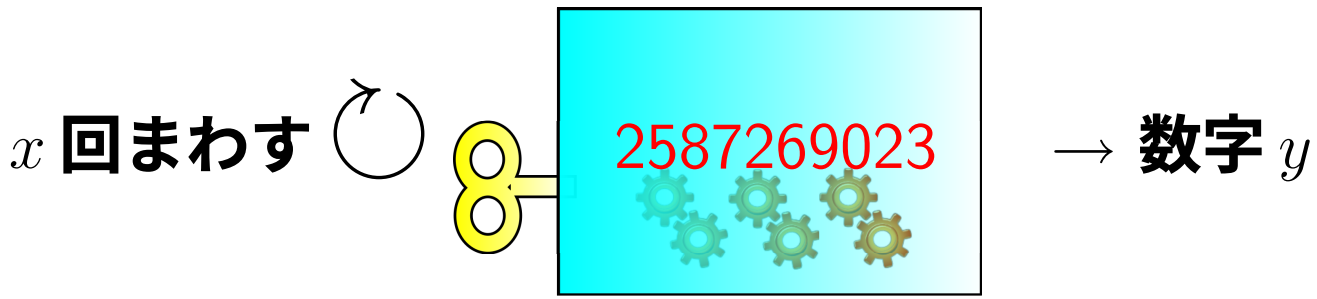
- $y = 10^x \pmod 7$ の表 ($p = 7$ のとき)

| $x$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $y = 10^x$ | 10 | 100 | 1000 | 10000 | 100000 | 1000000 | 10000000 |
| $y = 10^x \pmod 7$ | 3 | 2 | 6 | 4 | 5 | 1 | 3 |

- $y = a^x \pmod p$ の値に規則性がない（ある条件のもと）
  → $y$ から $x$ を決めるのが難しい
- $x = 1, x = 2, \cdots, x = 6$ のすべてを総当たりするしかない
- $p$ がとても大きいと，コンピュータでも計算量が莫大になる
  宇宙の年齢ぐらい計算しないと答えが出ない

# 離散対数問題と一方向性関数

$y = a^x \pmod p$ が “一方向カウンタ” になっている
（ただし $p$ が非常に大きな素数のとき，1024 桁ぐらいの素数）



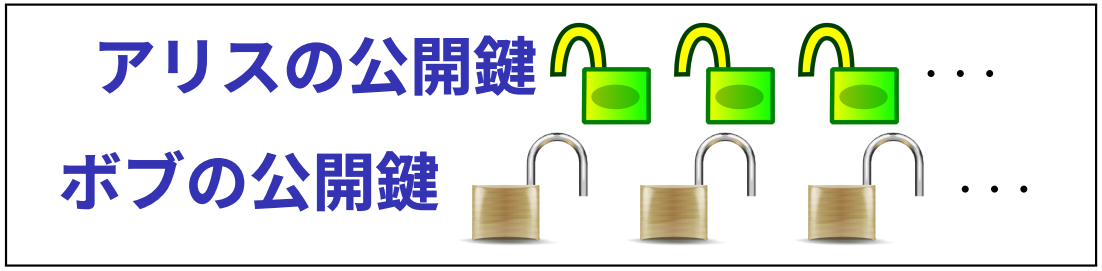

- $p$ が大きな素数とすると一方向性関数になる
  （$a$ がある条件をみたすとき）
- $y = a^x \pmod p$ はすぐに計算できる
- $y = a^x \pmod p$ を満たす $x$ の計算は宇宙の年齢ぐらいかかる

# 公開鍵暗号のアナロジー（比喩）：南京錠

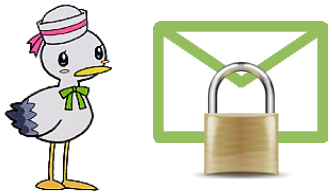
公開鍵 = 錠前  秘密鍵 = 錠前の鍵 

錠前から鍵を複製できないように作る
（原理的に可能だが，宇宙の年齢を要するぐらい複雑にする）

公開鍵簿
（電話帳の役目）



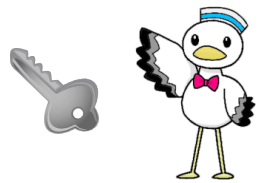

ボブの公開鍵を取得 ↙



送信者アリス

公開通信路 ⟶

盗聴者イブ



受信者ボブ

- 錠前は誰でも閉められる
- ボブの錠前を開けられるのは，鍵を持っているボブだけ

# 公開鍵暗号の手順

**閉める鍵（公開鍵）と開ける鍵（秘密鍵，個人鍵）を別々にする**

## 公開鍵暗号の手順

**準備**

1. 受信者は**暗号化する鍵と復号する鍵のペア**を作成
2. 暗号化する鍵は広く一般に公開をする**（公開鍵）**
3. 復号する鍵は受信者だけの秘密にする**（秘密鍵，個人鍵）**

**暗号通信の方法**

4. 送信者は目的の受信者の公開鍵を使って暗号化する
5. 受信者は自分の秘密鍵を使って復号する

# RSA 公開鍵暗号

- **公開鍵暗号のアイデアは** Diffie-Hellman (1976) **により示された**
- 1978 **年にリベスト・シャミア・アドルマン** (Rivest-Shamir-Adleman) **が具体的な方法を示した**
- **大きな素数のかけ算は短時間で計算できるが，素因数分解はコンピュータでも莫大な時間がかかることを利用する**

**現在では裏の歴史が明らかにされている**

- **公開鍵暗号はイギリス電子通信安全局で独立に作られていた**
- 1960 **年代**, **エリス** (Ellis), **公開鍵暗号のアイデア**
- 1973, **コックス** (Cocks), RSA **暗号の発明**
- 1974, **ウィリアムソン** (Williamson), Diffie-Hellman-Merkle **鍵共有方式の発明**

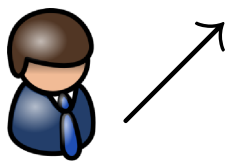
# 課題５
# メールでじゃんけん（コイン投げ）

# ビットコミットメント：将棋の封じ手

## 将棋の封じ手

- 将棋の対局で数日に渡るとき，その日の最後の手番となった対局者は差し手を**封筒に入れて立会人に預ける（委託）**
- 次の日は封筒に書いた通りの手を差す
- お互いに，相手の次の差し手が分からないようにして，考慮時間を公平にするため

立会人



対局者 A

対局者 B

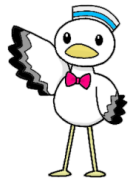
対局者 B の明日の手は？
（封筒の中は変えられない）

対局者 A の明日の手は？
（封筒の中は分からない）

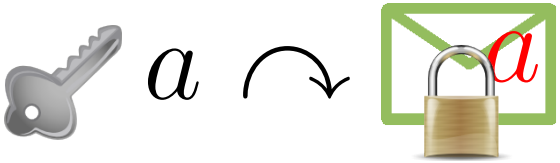
# ビットコミットメント＝立会人なしの封じ手

立会人なしで**ビット**を**委託**（コミットメント）する方法

アリス

ボブ 

1. **委託：**アリスは“$a=0$”**または**“$a=1$”を箱に入れ錠前を閉める




**箱をボブへ送信** $\longrightarrow$

2. ボブは鍵がないので箱の中身が分からない



3. **開示・検証：** アリスは鍵と$a$を送る．ボブは鍵を開けて検証．

$a$**と鍵をボブへ送信** $\longrightarrow$

**検証**

# ビットコミットメントの正当性

**ビットコミットメントに求められる性質**

1. **アリスは後で手を変えられない**
2. **ボブは封筒を勝手に開けることができない**
3. **お互いにルールに従っていれば公平性が保証される**

**ビットコミットメントの正当性**

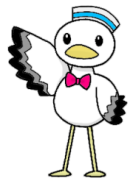
1. **アリスの不正：錠前をかけた箱をボブに送っているので，不正をするには送ったものと異なる値 $a'$ が出てくるような，別の鍵を送らないといけない．これは不可能．**
2. **ボブの不正：アリスから鍵を受け取るまで，開けることは不可能．**

# メールで “コイン投げ” をする方法

**ビットコミットメントを用いることで，コイン投げが実現可能**

**アリス**

**ボブ**



$a = 0$ **または** $a = 1$
**を “ランダムに” 選ぶ**

$a$ **をビットコミットメント**

$\longrightarrow$


$a,\ b$

$b$ **を送信**

$\longleftarrow$

$b = 0$ **または** $b = 1$
**を “ランダムに” 選ぶ**

$a,\ b$

$a$ **と鍵を送信**

$\longrightarrow$

$b,\ a$ ↶

$a$ **を検証**

$\Downarrow$

$c = a \oplus b$

$c$ **をコイン投げの結果とする**

$\Downarrow$

$c = a \oplus b$

# コイン投げに求められる性質

**コイン投げに求められる性質**

- **結果が $c = 0$ または $c = 1$ となる確率は等しく $50\%$**
- **アリスとボブは不正を行えない**

**考えられる不正**

- **不正 $1$：$a$ や $b$ を “ランダム” ではなく恣意的な値にする**
- **不正 $2$：相手の値に応じて自分の値を変える**

# コイン投げの正当性

## 不正 1 （$a$ や $b$ を恣意的な値にする）への対応

- アリスが $a$ を恣意的に決めても，ボブがルールに従っていれば $b$ はランダム．**ビットの足し算**

$$c = a \oplus b$$

**は “ランダムに” $a$ を反転する（オセロ）** ので，$c$ はランダム．
- ボブが $b$ を恣意的に決めても，アリスがルールに従っていれば $c = a \oplus b$ **は同様にランダム**になる．

## 不正 2 （相手の値に応じて自分の値を変える）への対応

- ビットコミットメントの性質**（後で値を変えられない）**より，アリスは $b$ を手にした後に $a$ を変更することは不可能．
- ビットコミットメントの性質**（勝手に開けられない）**より，ボブは $b$ を送る前に $a$ を知ることは不可能．

# 参考文献

## 一般向け

- **サイモン・シン, 暗号解読ーロゼッタストーンから量子暗号までー, 新潮社**, 2001. (Simon Singh, The Code Book, 1999).

**おすすめ**

- **岡本龍明, 暗号と情報セキュリティ, 日経** BP, 1978.
- **結城浩, 新版 暗号技術入門, ソフトバンク・クリエイティブ**, 2008.

## 専門書

- **岡本龍明・山本博資, 現代暗号, 産業図書**,1997.
- **池野信一・小山謙二, 現代暗号理論, 電子通信学会**, 1986.

# おわりに

まとめ

- 暗号技術は情報ネットワーク社会のインフラ（社会基盤）
- 暗号は鍵が命
  - アルゴリズム（手続き）は公開
  - 鍵を秘密にすることで通信の安全をまもる
- 共通鍵暗号と公開鍵暗号
- 公開鍵暗号には整数論，代数学という数学が使われている
  - それまで整数論は役に立たない数学と言われていた
- ビットコミットメント，コイン投げ
  - 電子商取引に応用されている
  - 電子投票，電子マネーなどで偽造防止＋匿名性を確保するために応用されている

未来の暗号技術

- 量子コンピュータ，量子暗号，量子情報理論