

情報技術と数理の利用

コンピュータの発展・情報化社会の進展に伴い、

数理の解明とその利用が

ますます社会と密接になってきた

数理技術

情報技術と数理の利用

物理技術（17世紀以来）:

基礎数理 \implies 物理現象 \implies 実用技術
理論的 仕組み
裏付け の構成

情報技術（20世紀以来）:

数理現象 \implies 数理技術 \implies 実用技術
仕組み 物理的
の構成 実現

数理の解明が直接に技術発展に繋がる

計算機で扱えるもの

計算機では本質的に

有限・離散

のものしか扱えない

- 無限・連続のものに近い
- 有限・離散であることの積極的活用

暗号 (cryptography)

- 秘密通信
- 電子認証・電子署名
- 鍵共有

暗号の利用

- 古典的：戦争・謀略など
- 現代：情報通信一般

→ 個人の独立を守るための重要な数理技術

既に身近な暗号の利用

メディアセンターのコンピュータを使う際の
パスワードによる本人認証にも
暗号（暗号化）が使われている

入力したパスワードを
保管してあるデータと照合しているのだが、

実は、
パスワードそのものを保管しているのではない

定まった方式（暗号化関数）で
パスワードを変換して保管している

パスワードによる本人認証

- 暗号化関数でパスワードを変換して保管
- 入力したパスワードを暗号化関数で変換して、
保管してある文字列と照合

暗号化関数に要請される性質は？

- 間違った文字列では変換結果が一致しない
→ 異なる入力には異なる値を返す（単射）
- 保管してある文字列が露見しても
元のパスワードが判明しない
→ 一方向性関数 (one-way function)

パスワードによる本人認証

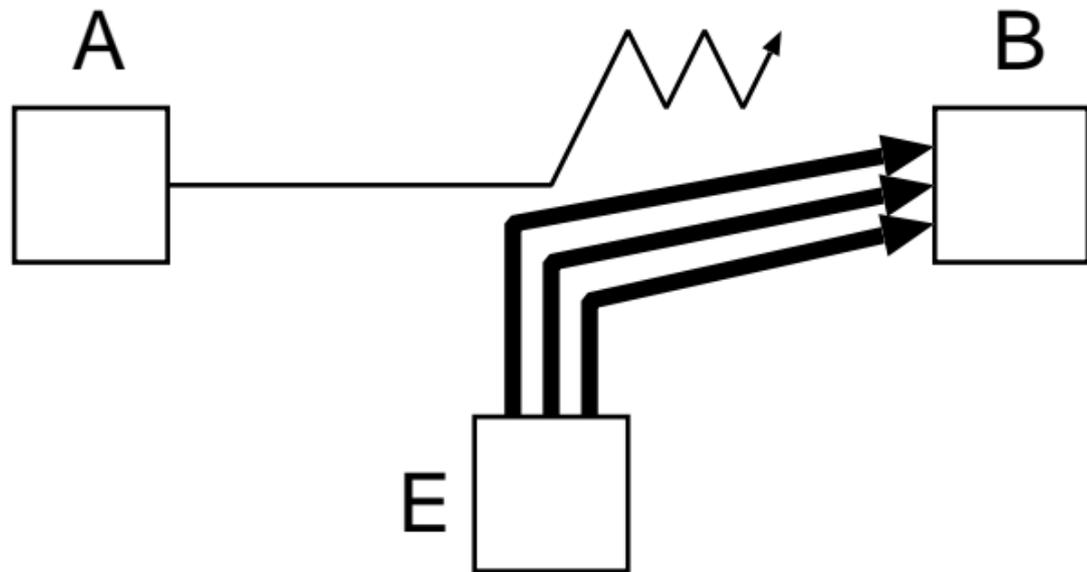
良い暗号化関数で変換して保管していても

通信経路の途中で盗聴されてしまったら

パスワードが露見してしまう

→ **暗号**による秘密通信

DoS (Denial of Service) 攻撃

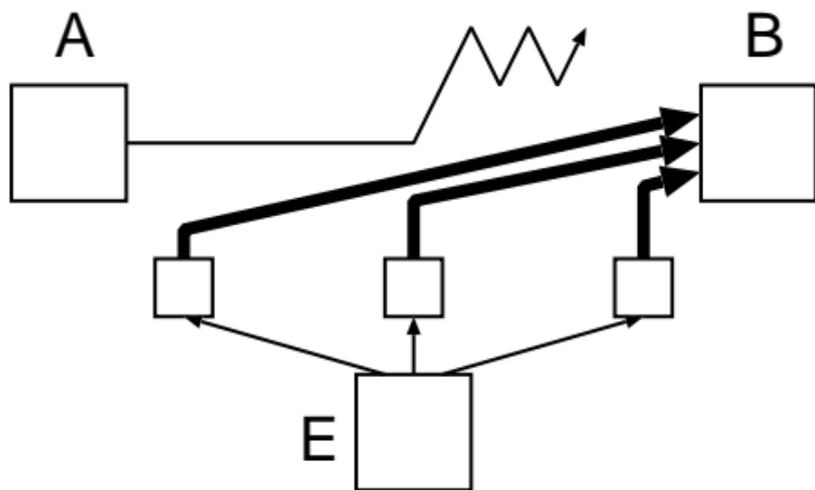


B を機能停止に追い込むには

E に相当のマシンパワーが必要

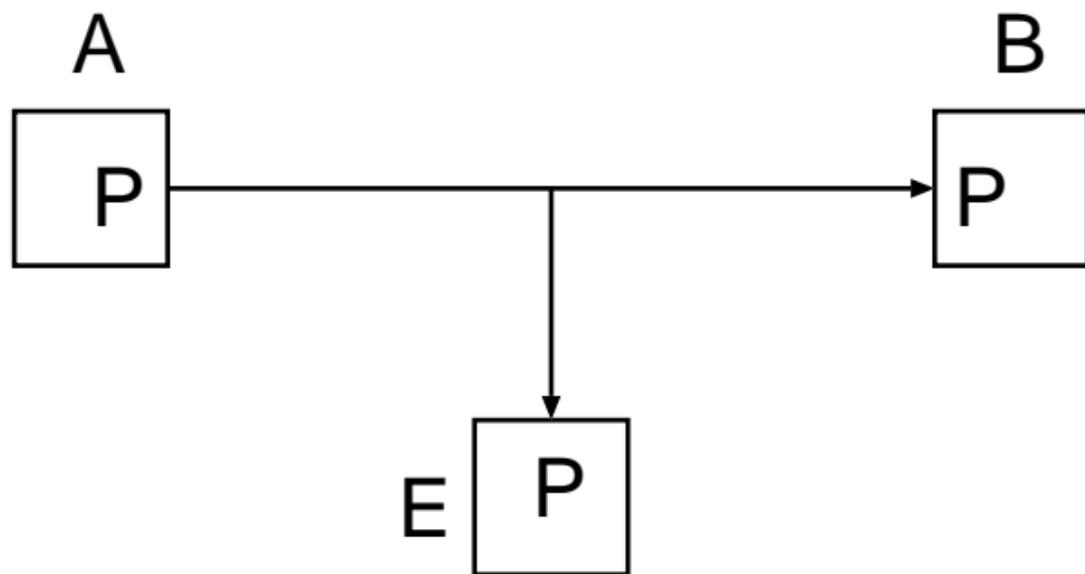
そこで実際には …

DoS (Denial of Service) 攻撃



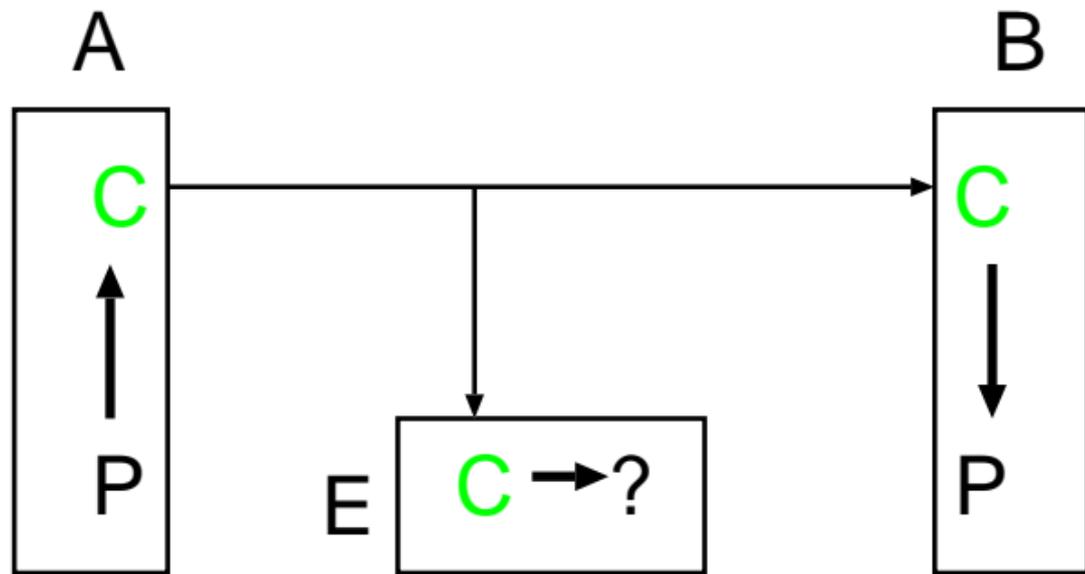
実際には、
コンピュータウイルス・乗っ取りなどで
制御下に置いた多数の機械から一斉に攻撃
(Distributed DoS, DDoS)

盗聴



現在の計算機ネットワークの仕組みでは、
事実上、通信経路は誰にでも見られる

暗号通信で盗聴対策

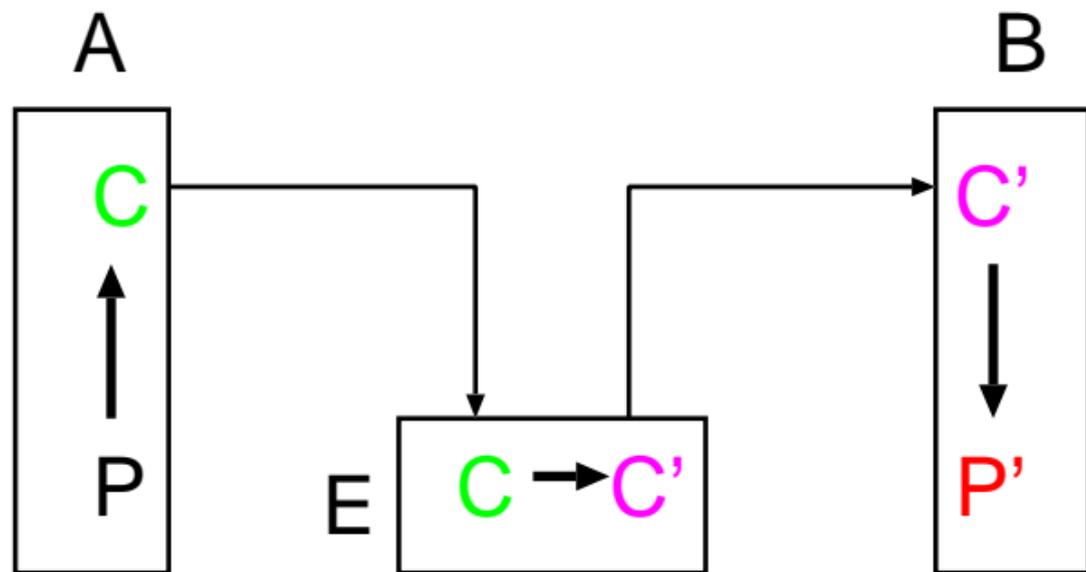


P : 平文 (plain text), C : 暗号文 (ciphertext)

P → C : 暗号化 (encryption)

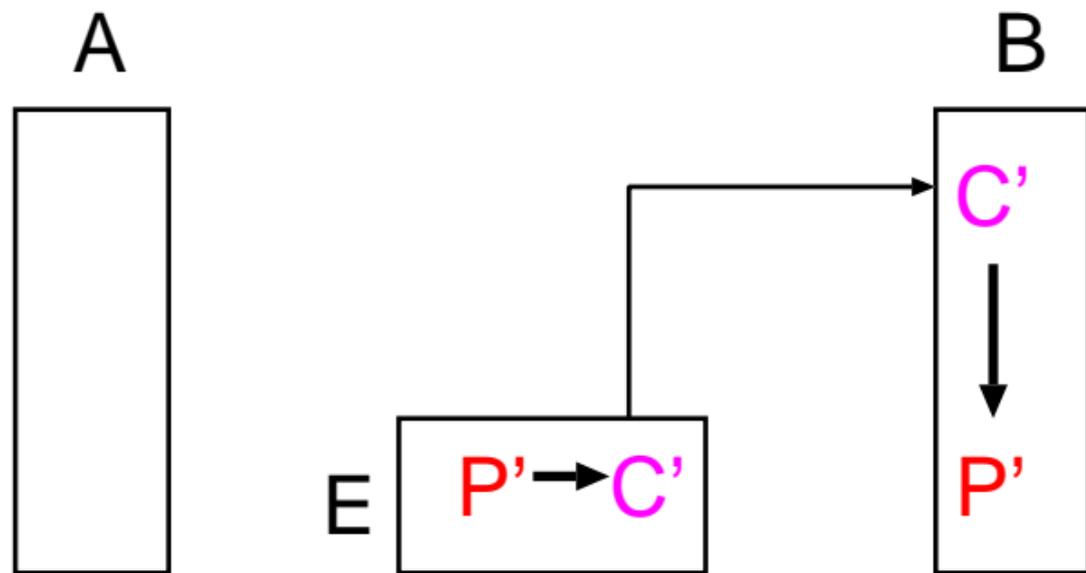
C → P : 復号 (decryption) ・ 解読

改竄



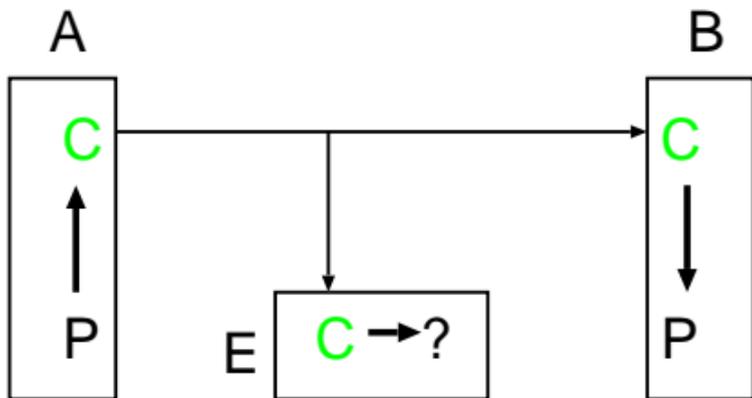
A が送信した情報であることを
確かめられるような仕組みが必要
(電子認証・電子署名)

なり済まし



A が送信した情報であることを
確かめられるような仕組みが必要
(電子認証・電子署名)

暗号 (cryptography)



- 送信者 **A** が平文 **P** を暗号化、暗号文 **C** を送信
- 受信者 **B** が暗号文 **C** を受信、平文 **P** に復号
- 盗聴者 **E** は暗号文 **C** を知っても
平文 **P** を復元できない

→ **B** だけが復号鍵を持っていることが必要

暗号 (cryptography)

仮定：

公開された情報伝達路（盗聴可能と仮定）で、

暗号方式を公開して通信

古典的な暗号の例：Caesar 暗号

- ここでは、
a ~ z のアルファベットから成る文字列を
暗号化
- 鍵： $1 \leq n \leq 25$ なる整数 n
- 暗号化：アルファベットを後ろに n 個ずらす
- 復号：アルファベットを前に n 個戻す
(但し、 $\dots xyzabc \dots$ と繋がるとする)

実習：古典的な暗号の例（Caesar 暗号）

Caesar 暗号で暗号化された

次の文字列を解読してみよう

phq dqg zrphq iru rwkhuv zlwk rwkhuv

- 共通鍵： $1 \leq n \leq 25$ なる整数 n
- 暗号化：アルファベットを後ろに n 個ずらす
- 復号：アルファベットを前に n 個戻す
(但し、 $\cdots xyzabc \cdots$ と繋がるとする)

Caesar 暗号の脆弱性

鍵を知らなくても容易に解読されてしまった

何故か？

- 鍵の可能性が少なく、総当たりで倒せる
- 暗号文に平文の特徴が残っている

このような脆弱性を克服した暗号方式が
現在では用いられている

- **DES (Data Encryption Standard)**
- **AES (Advanced Encryption Standard)**

共通鍵暗号

Caesar 暗号のように、

暗号化と復号とで同じ鍵を用いる暗号を

共通鍵暗号という

- 仕組みが比較的簡明
- 暗号化・復号が一般に高速
- 事前に鍵を秘密裡に共有しておく必要あり

現代における暗号への要請

現在の情報化社会では

様々な場面で暗号が使われている

例：インターネット取引（ネットショッピングなど）

- 不特定多数の人と暗号通信をしたい
- 事前に鍵を共有できない

→ 共通鍵暗号では実現が困難

→ **公開鍵暗号**・鍵共有方式のアイデア

(1976, Diffie, Hellman)

公開鍵暗号

暗号化鍵（公開鍵）・復号鍵（秘密鍵）が別

- 事前の鍵共有の必要無し
→ 見ず知らずの人からも送ってもらえる
- 認証・署名機能がある
 - 改竄・なり済ましの対策
 - 否認防止の機能も持つ

公開鍵暗号

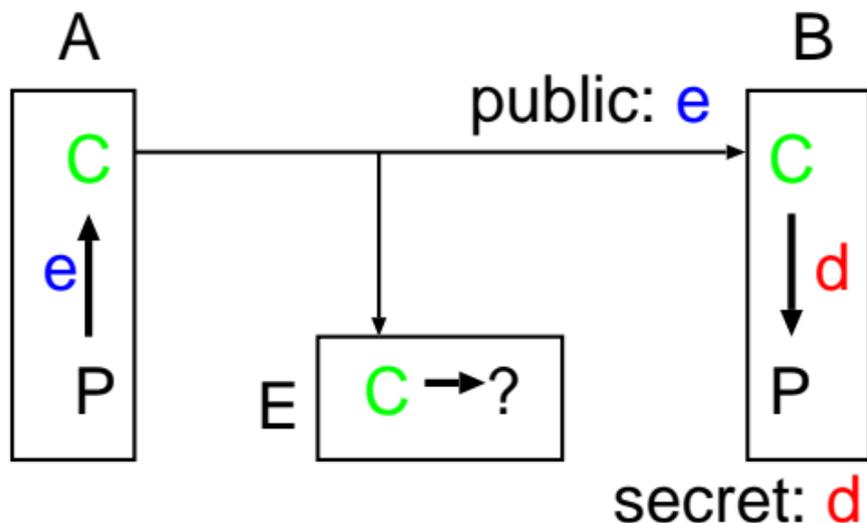
但し、一般には、
暗号化・復号が共通鍵暗号に比べて低速

そこで、

- 始めに公開鍵暗号方式で鍵を送付・共有
- その鍵を用いて秘密鍵暗号方式で通信

というように、組合わせて用いることが多い

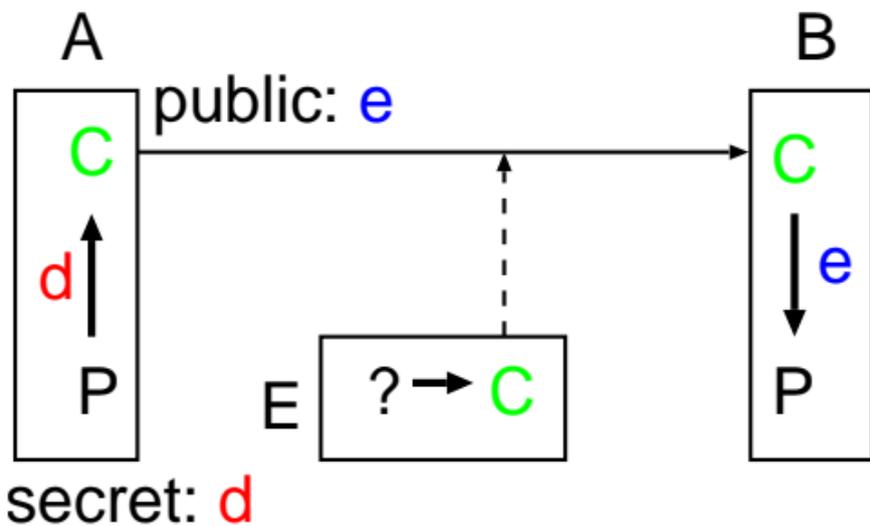
公開鍵暗号による暗号通信



しかし、これだと誰でも暗号化できるので、
A 氏が送った保証がない

→ 署名の必要性

公開鍵暗号を用いた認証・署名



盗聴者 E 氏は

平文 P は判らないが、暗号文 C は盗聴可能

→ いつも同じ署名は使えない

公開鍵暗号を用いた認証・署名

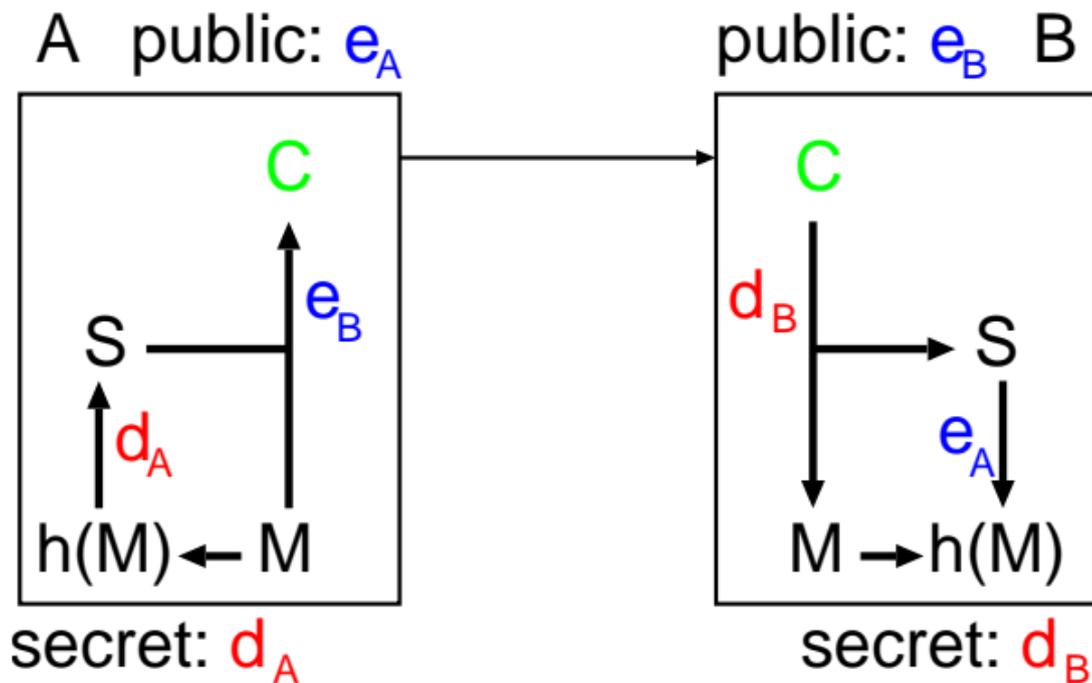
実際には、メッセージ本文 M に対して、

M から決まる短い値（ハッシュ値） $h(M)$ を
送信者 A 氏の秘密鍵で暗号化した文字列 S

を本文 M に添付して、

受信者 B 氏の公開鍵と一緒に暗号化して送る

公開鍵暗号を用いた認証・署名 2



公開鍵暗号の特徴

- 暗号化は誰でも出来る
(暗号化鍵は公開されている)

- 復号は秘密鍵を知らないと出来ない
(もの凄く時間が掛かる)

そんな都合の良い仕組みが本当にあるのか？

公開鍵暗号の例：RSA 暗号

Rivest, Shamir, Adleman (1977)

- 大きな素数 p, q を選び、積 $n = pq$ を作る
- n を用いて、公開鍵 e ・秘密鍵 d の対を作る
- 暗号化の計算は n と公開鍵 e とから可能
- 復号は秘密鍵 d を用いる
- n と公開鍵 e とから秘密鍵 d を求めるには、 n の素因数分解 $n = pq$ が必要
- しかしそれは困難（膨大な計算時間が掛かる）

RSA 暗号

- 大きな素数 p, q を選び、積 $n = pq$ を作る
 - ★ $p - 1$ と $q - 1$ との最小公倍数
 $l := \text{lcm}(p - 1, q - 1)$ を求めておく
- 公開鍵 e ・ 秘密鍵 d の対を作る
 - ★ l と互いに素な整数 e を取る
 - ★ $ed \equiv 1 \pmod{l}$ なる整数 d を求める
- 暗号化 : $C \equiv P^e \pmod{n}$
- 復号 : $P \equiv C^d \pmod{n}$

何故これでうまく機能するのか？

鍵対の構成

l と互いに素な整数 e を与えたとき、

$ed \equiv 1 \pmod{l}$ なる整数 d
(l を法とした e の“逆数”) は、

Euclid の拡張互除法を用いることにより、
効率良く求めることが出来る !!

$(e, l) = 1$ より $\exists c, d \in \mathbb{Z} : ed + lc = 1$

中国式剰余定理

m_1, m_2 が互いに素のとき、

$$\mathbb{Z}/m_1m_2\mathbb{Z} \simeq \mathbb{Z}/m_1\mathbb{Z} \times \mathbb{Z}/m_2\mathbb{Z}$$

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \end{cases}$$

は解を持ち、しかも $\text{mod } m_1m_2$ で一意的

$$\begin{cases} x \equiv a \pmod{m_1} \\ x \equiv a \pmod{m_2} \end{cases} \iff x \equiv a \pmod{m_1m_2}$$

RSA 暗号の検証

- 暗号化 : $C \equiv P^e \pmod{n}$
- 復号 : $P \equiv C^d \pmod{n}$

$P \equiv (P^e)^d \pmod{n}$ であるか

p, q は相異なる素数 \longrightarrow 互いに素
 $\longrightarrow n = pq$ で中国剰余定理が使える

$\longrightarrow \pmod{p}$ と \pmod{q} とで見れば良い

Fermat の小定理

p を素数とするとき、

p と互いに素な整数 a に対し、

$$a^{p-1} \equiv 1 \pmod{p}$$

$n = pq, l = \text{lcm}(p-1, q-1), ed \equiv 1 \pmod{l}$

のとき、

$ed \equiv 1 \pmod{p-1}$ より $P^{ed} \equiv P^1 \pmod{p}$

$ed \equiv 1 \pmod{q-1}$ より $P^{ed} \equiv P^1 \pmod{q}$

併せて、 $P^{ed} \equiv P \pmod{n}$

RSA 暗号の安全性

これで RSA 方式が実際に動かせることが判ったが、

では、RSA 暗号は安全な暗号なのか？

RSA 暗号

- 大きな素数 p, q を選び、積 $n = pq$ を作る
 - ★ $p - 1$ と $q - 1$ との最小公倍数
 $l := \text{lcm}(p - 1, q - 1)$ を求めておく
- 公開鍵 e ・ 秘密鍵 d の対を作る
 - ★ l と互いに素な整数 e を取る
 - ★ $ed \equiv 1 \pmod{l}$ なる整数 d を求める
- 暗号化 : $C \equiv P^e \pmod{n}$
- 復号 : $P \equiv C^d \pmod{n}$

RSA 暗号

- 大きな素数 p, q を選び、積 $n = pq$ を作る
- n を用いて、公開鍵 e ・ 秘密鍵 d の対を作る
(Euclid の互除法を用いる)
- 暗号化の計算は n と公開鍵 e とから可能
$$C \equiv P^e \pmod{n}$$
- 復号は秘密鍵 d を用いる
$$P \equiv C^d \pmod{n}$$
- n と公開鍵 e とから秘密鍵 d を求めるには、
 n の素因数分解 $n = pq$ が必要
(p, q が不明だと d が求まらない)

RSA 暗号の安全性

結局、RSA 暗号の安全性は、

素因数分解問題の（計算量的）困難さ

に掛かっている

現在の所、

充分速い計算法（多項式時間アルゴリズム）は
知られていない

→ 多くの数学者・計算機科学者が鋭意研究中

RSA 暗号の安全性

もしも画期的に高速なアルゴリズムを発見したら、
いち早く学会（世界）に公表して
人類共有の財産とするであろう

我々科学者の独立によって
情報化社会の安全性が保証されている

というのは綺麗事で …

素因数分解問題の他にも、

離散対数問題 (Discrete Logarithm Problem)

も暗号に応用されている

法 p と整数 g とを固定して、整数 A に対し、
$$g^x \equiv A \pmod{p}$$

となる整数 x を求めることができるか

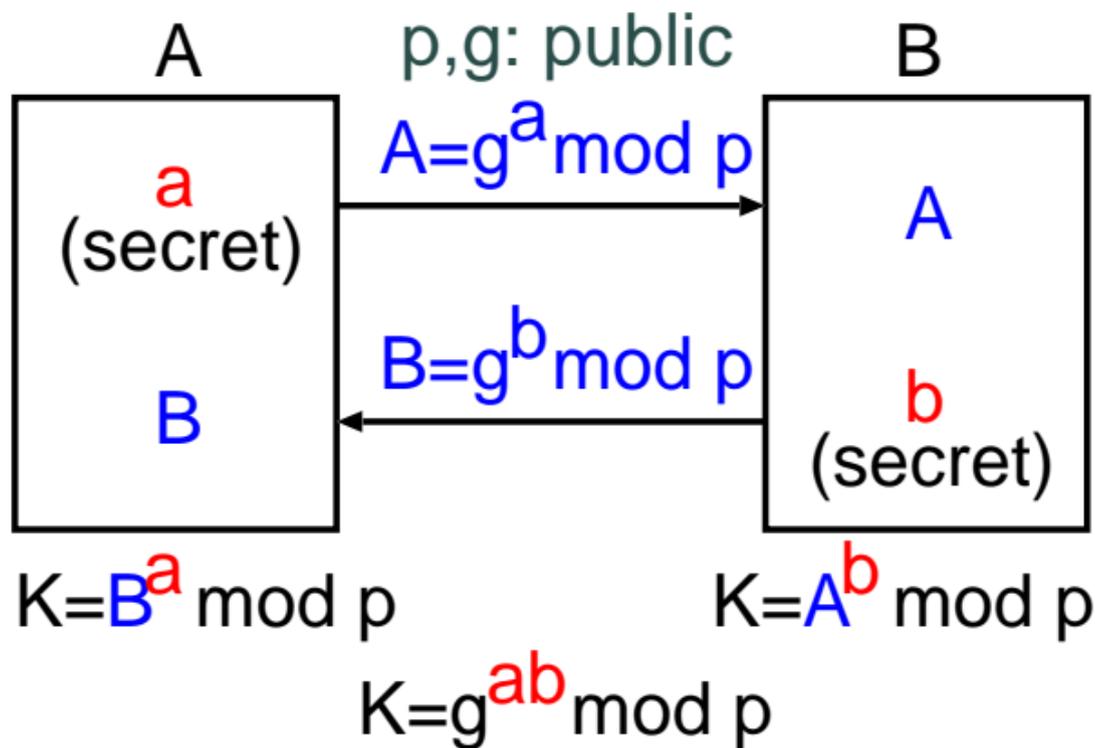
$$“x = \log_g A”$$

離散対数問題の応用 (Diffie-Hellman の鍵共有方式)

盗聴されている情報通信路を用いて、

秘密鍵を共有することが出来るか？

Diffie-Hellman の鍵共有方式



実際には、

共通鍵暗号方式の方が公開鍵暗号方式より高速

→ 両者を組み合わせて用いることが多い

- 始めに公開鍵暗号方式や鍵共有方式で
秘密鍵を共有
- その秘密鍵を用いて共通鍵暗号方式で通信

EIGamal 暗号

離散対数問題と疑似乱数と組み合わせて
暗号方式を構成したもの

RSA 暗号と同様に有限体の乗法群を用いる他にも、

- 有限体上の楕円曲線の有理点の成す群
- 有限次代数体の **ideal** 類群

などの有限アーベル群も用いられる
(様々な数理現象が利用されている)

秘密分散・公開鍵暗号・鍵共有などの

基本的な数理技術を組み合わせて用いると、

電子投票方式などを構成することも出来る

まとめ

- 現代の情報化社会を支える基盤技術として
種々の数理技術が利用されている
- 数理現象の解明が
直接に技術の進歩に繋がっている
- 人間は弱いもの
→ 不正をしようとしても出来ない
システムが望まれる
→ “最も確かなもの” としての数理の利用