期末試験のお知らせ

7月30日(木) 13:30〜14:30

(60分試験)

8-208教室 (ここじゃない)

- 最終回(7/23)の講義内容まで
- 学生証必携
- 学部・大学院合併で行なう

レポート提出について

期日: 8月7日(金)20時頃まで

内容:

配布プリントのレポート問題の例のような内容及び授業に関連する内容で、授業内容の理解または発展的な取組みをアピールできるようなもの(詳細はプリント参照のこと)

提出方法:

- **授業時に手渡し**
- **4-574 室扉のレポートポスト**
- **電子メイル**

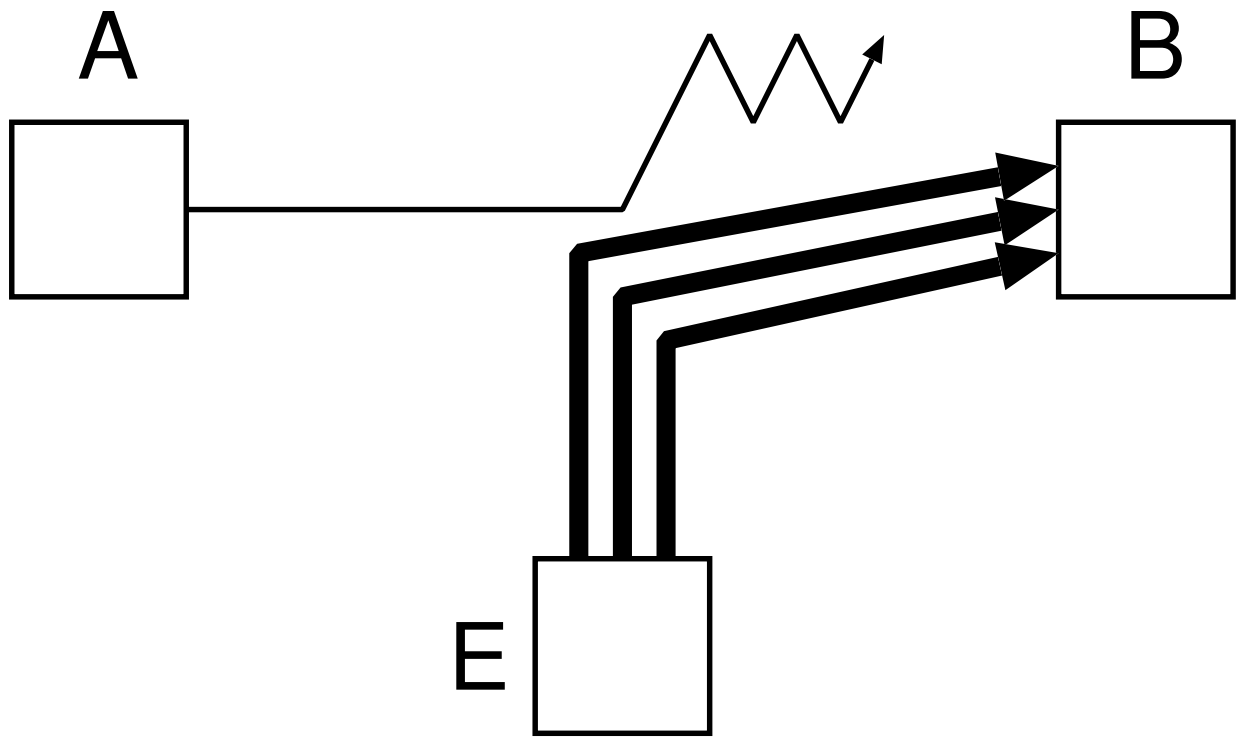
情報通信を行なう際の要請

- **効率的に** $\longrightarrow$ **情報理論**
- **確実に** $\longrightarrow$ **符号理論**
- **安全に** $\longrightarrow$ **暗号理論**

安全な情報伝達を阻害するもの

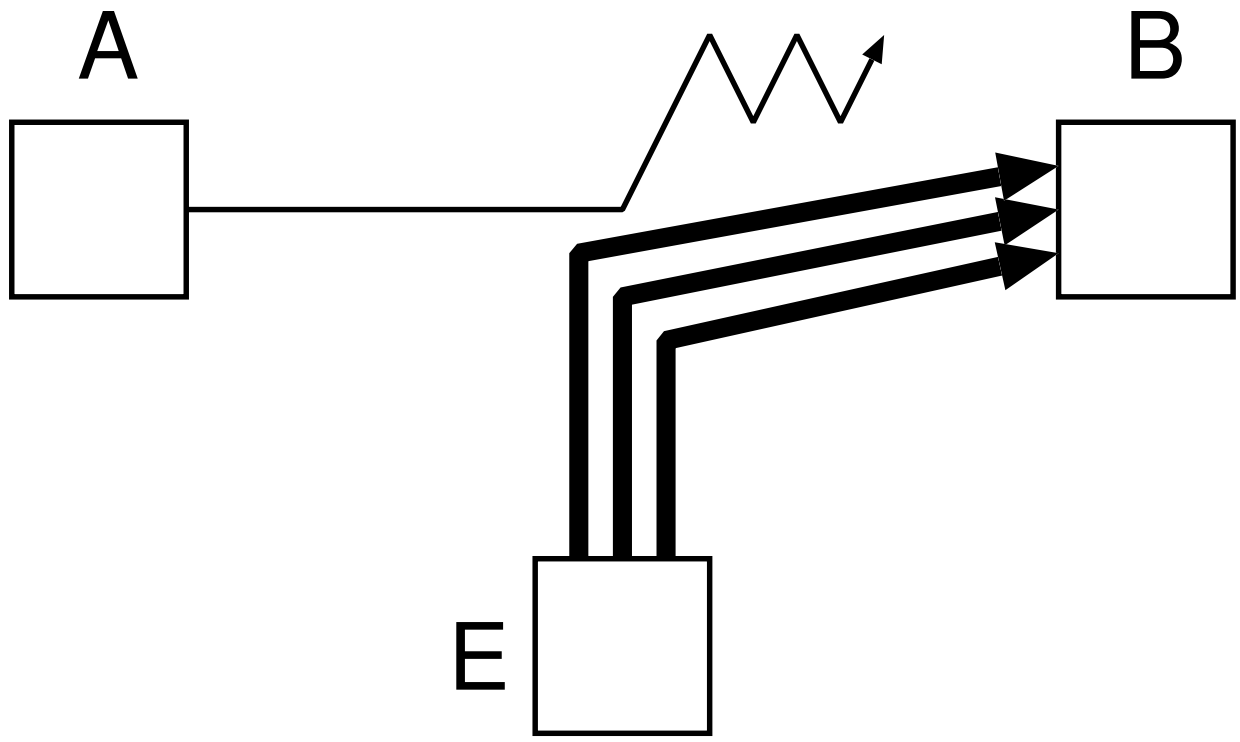
- 妨害 (DoS 攻撃など)
- 盗聴
- 改竄
- なり済まし　　など

DoS (Denial of Service) 攻撃



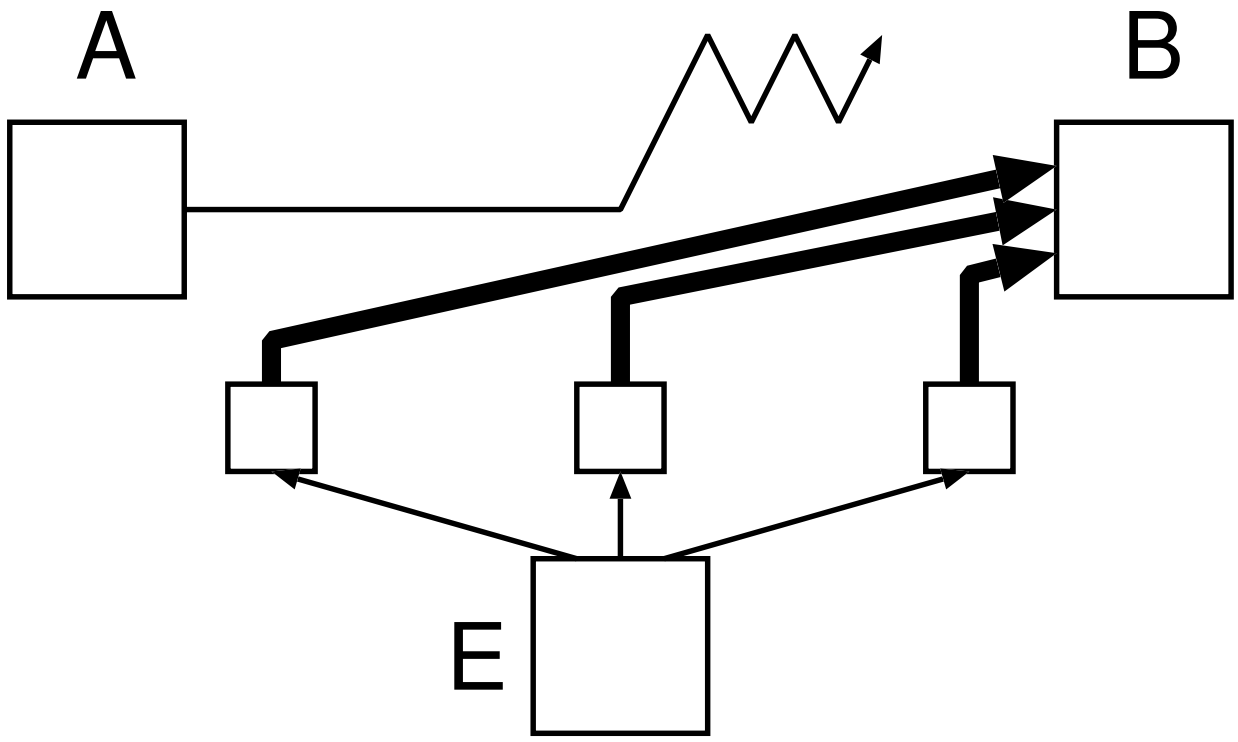
B を機能停止に追い込むには
E に相当のマシンパワーが必要
そこで実際には ...

DoS (Denial of Service) 攻撃



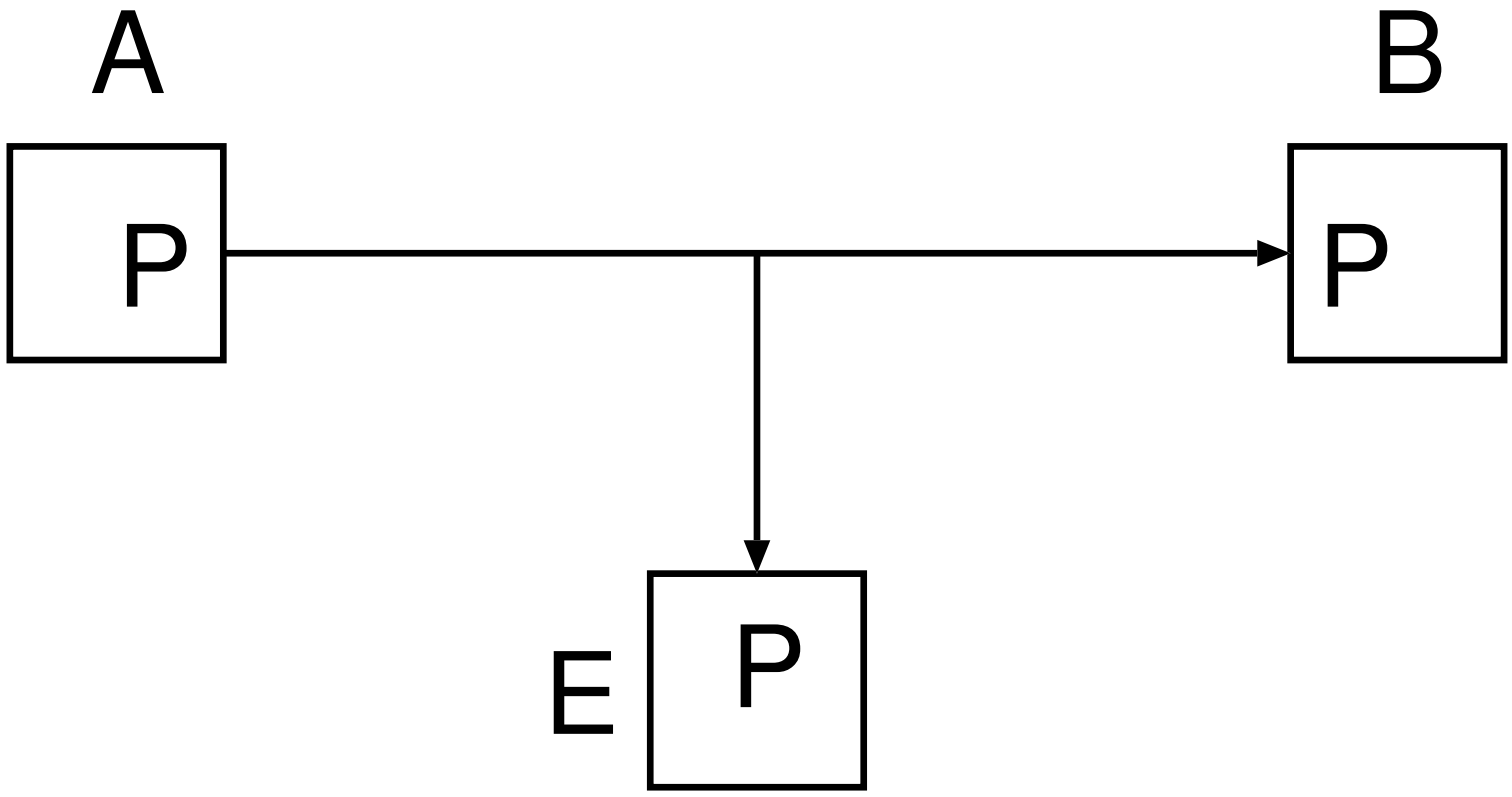
B を機能停止に追い込むには
E に相当のマシンパワーが必要
そこで実際には ...

DoS (Denial of Service) 攻撃



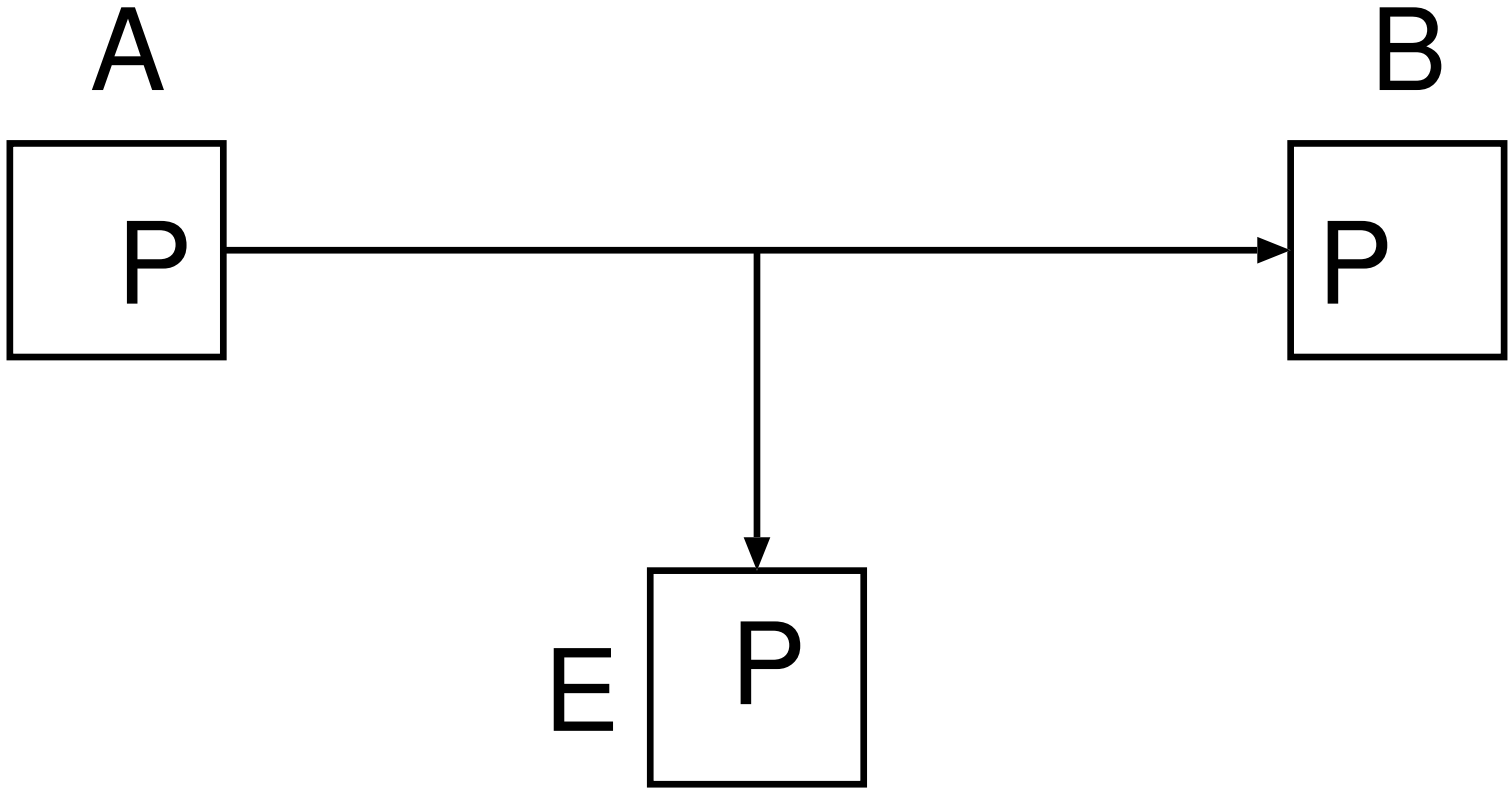
実際には、
コンピュータウイルス・乗っ取りなどで
制御下に置いた多数の機械から一斉に攻撃

盗聴



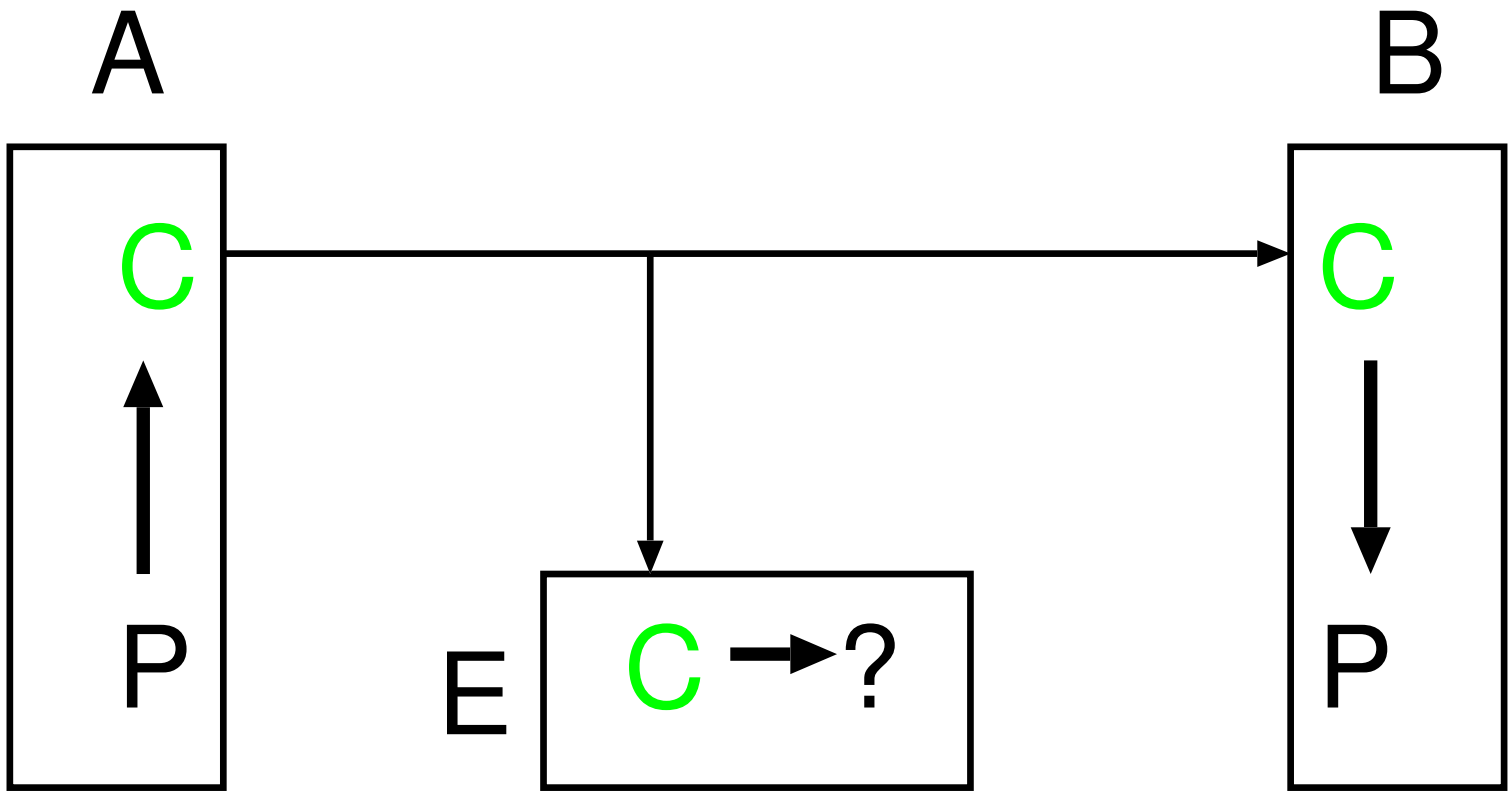
現在の計算機ネットワークの仕組みでは、
事実上、通信経路は誰にでも見られる

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

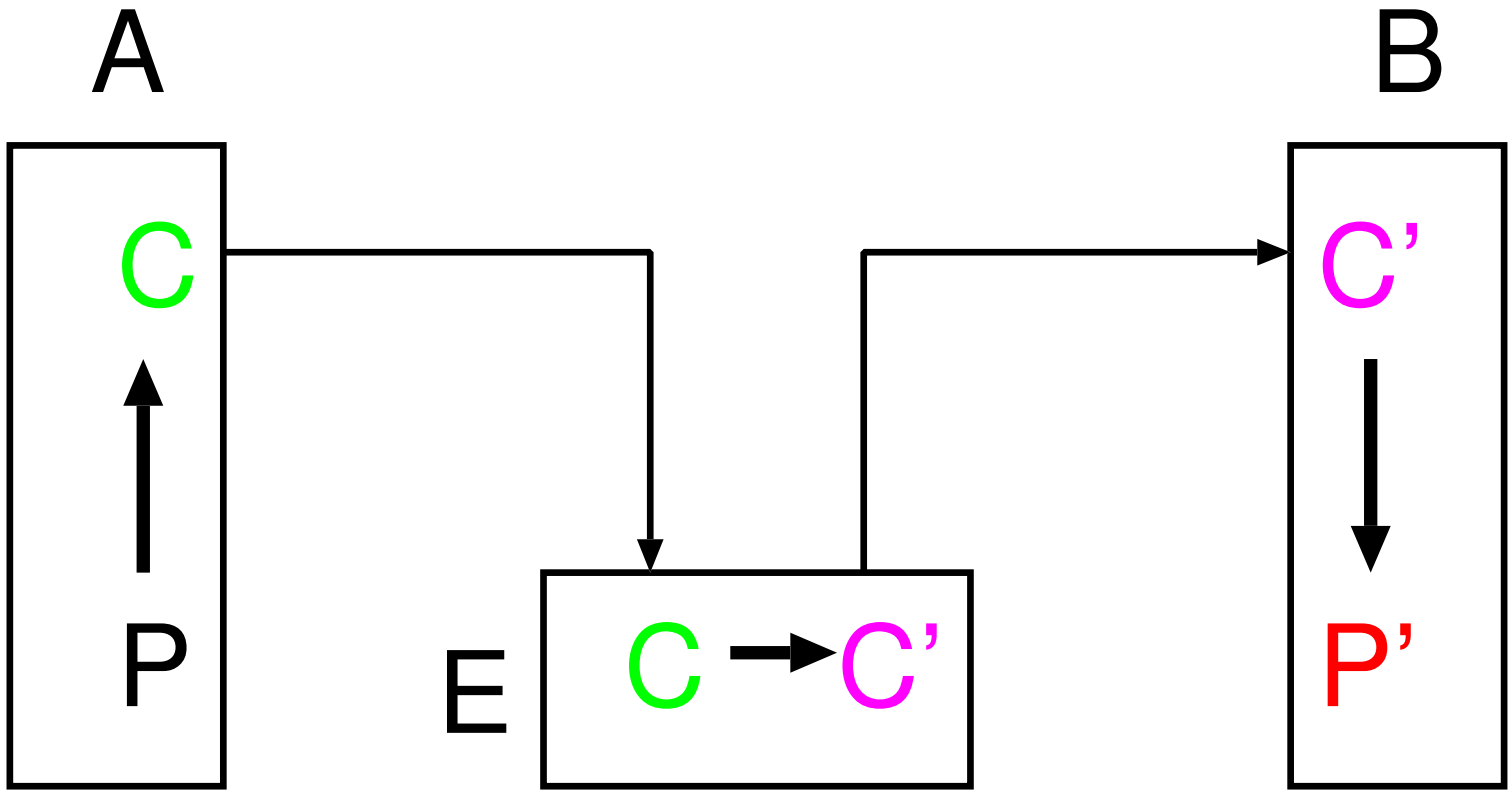


P: 平文 (plain text), C: 暗号文 (ciphertext)

P → C : 暗号化 (encryption)

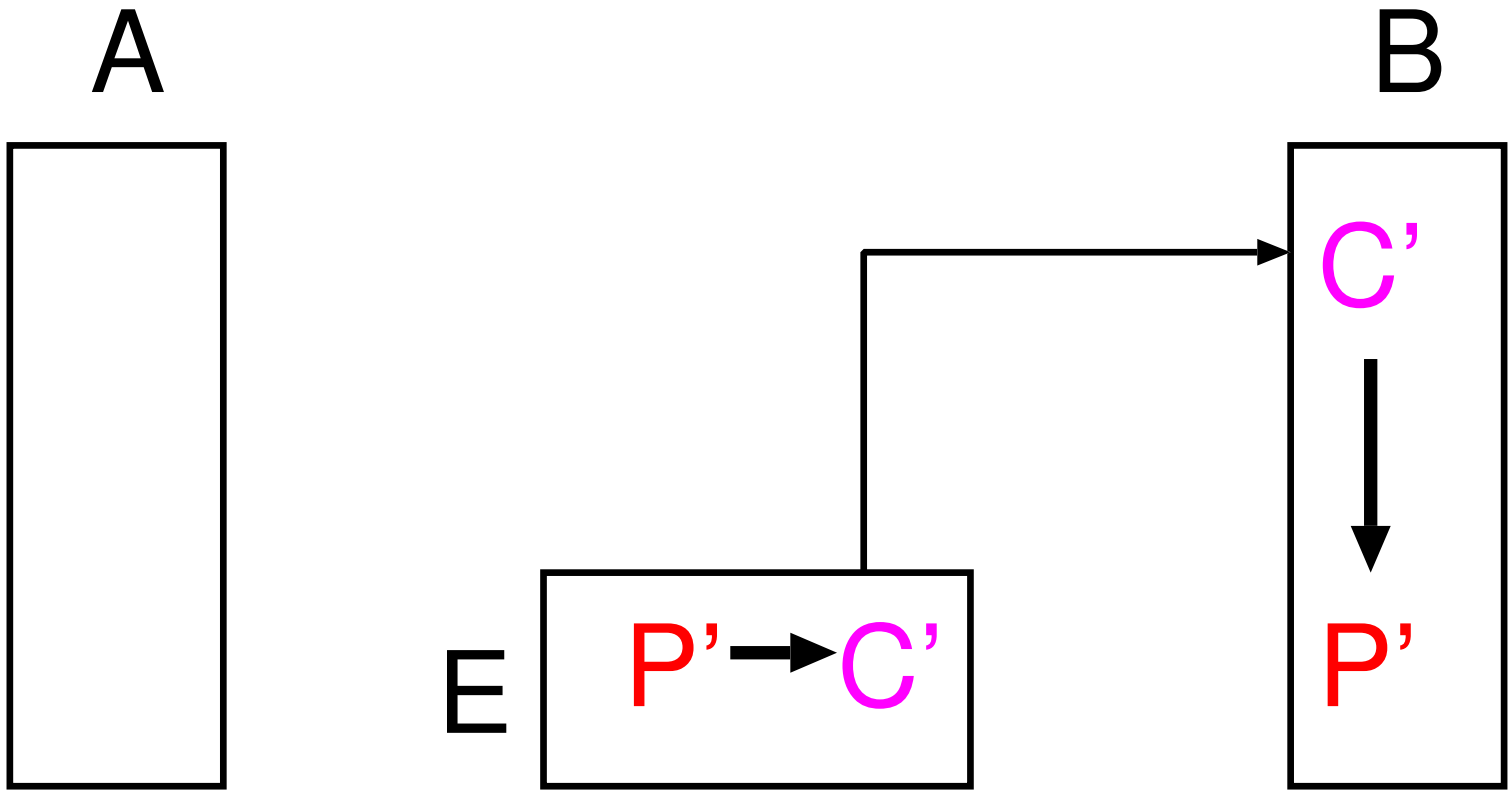
C → P : 復号 (decryption)・解読

改竄



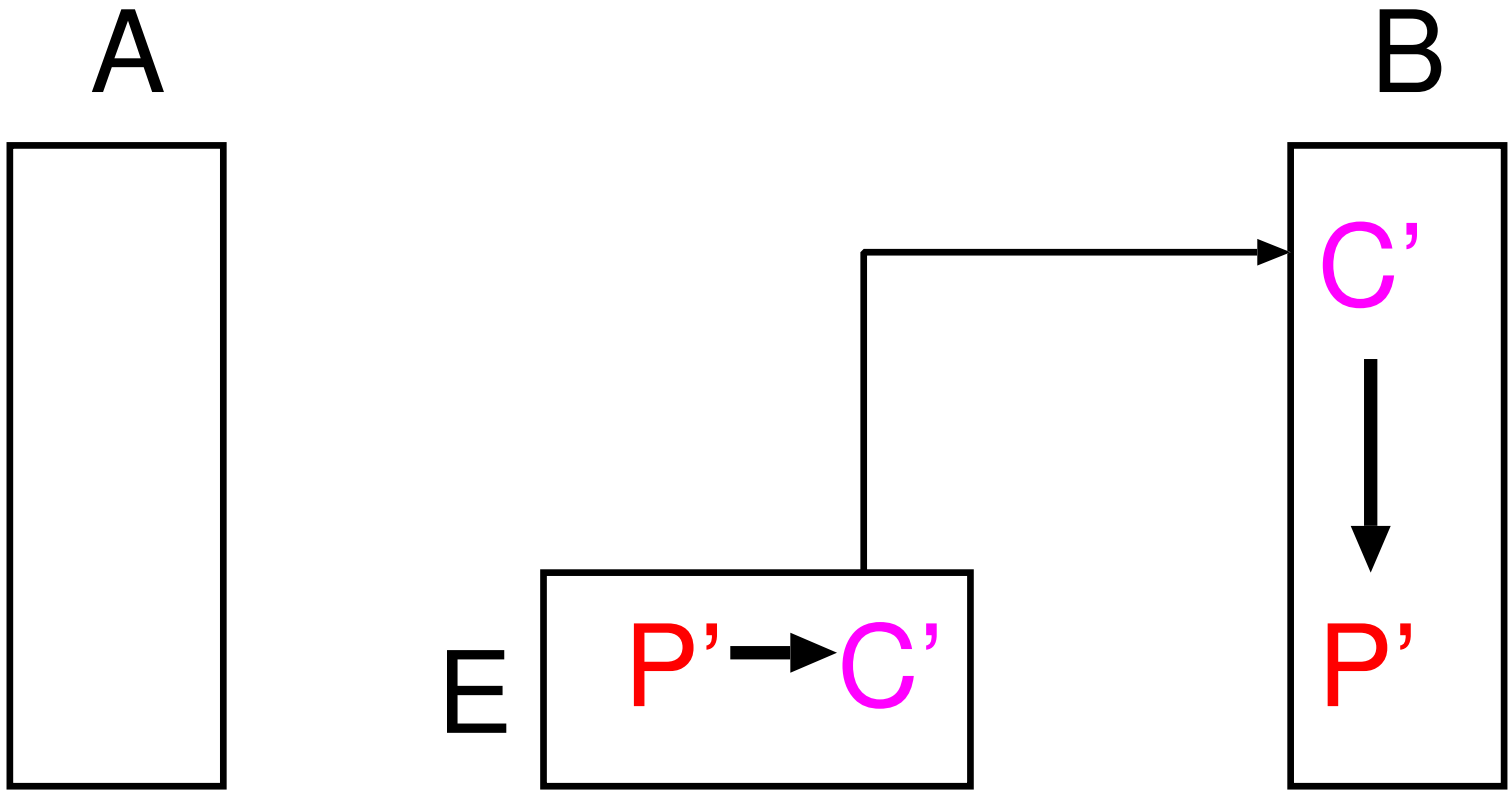
A が送信した情報であることを
確かめられるような仕組みが必要
(電子認証・電子署名)

なり済まし



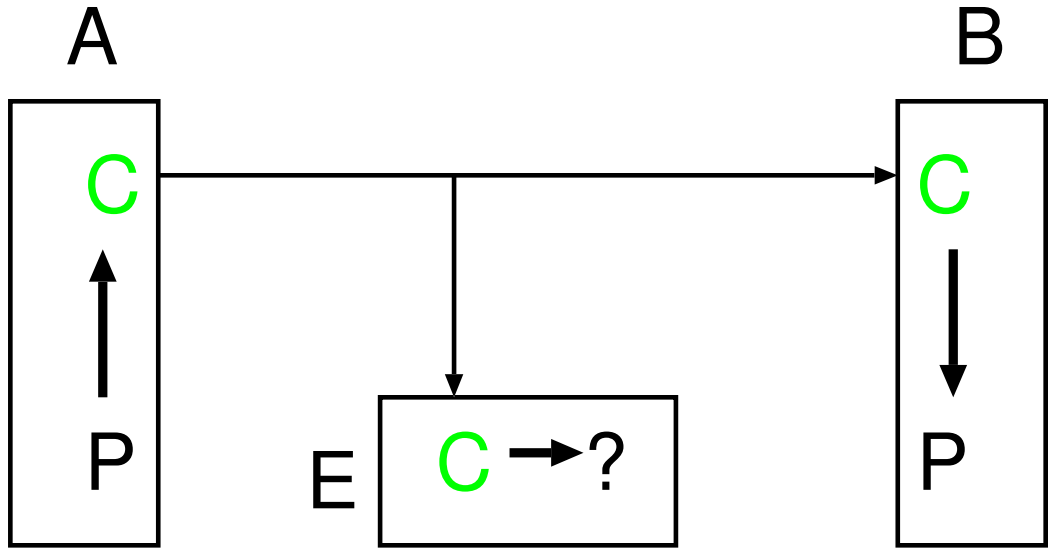
A が送信した情報であることを
確かめられるような仕組みが必要
(電子認証・電子署名)

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)

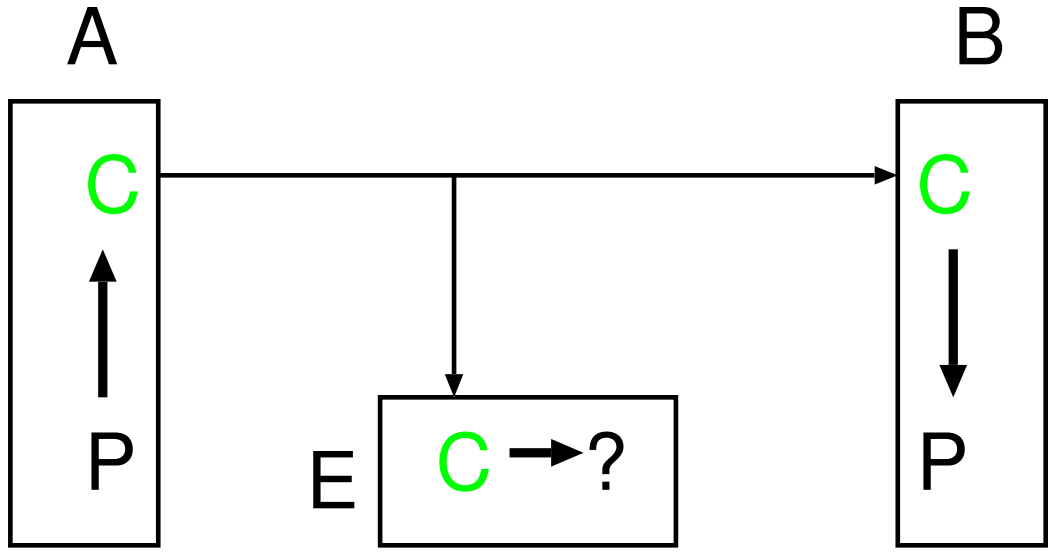
暗号 (cryptography)



- 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)



- 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

- **秘密鍵暗号 (共通鍵暗号)**
- **公開鍵暗号**

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

- **秘密鍵暗号 (共通鍵暗号)**
- **公開鍵暗号**

暗号 (cryptography)

- **共通鍵暗号** (秘密鍵暗号)
  - ⋆ 送信者・受信者で同じ鍵を秘密裡に共有
  - ⋆ 共通の鍵で暗号化・復号を行なう

- **公開鍵暗号**
  - ⋆ 暗号化鍵 (公開鍵)・復号鍵 (秘密鍵) が別
  - ⋆ 公開された暗号化鍵を用いて暗号化
  - ⋆ 復号鍵は受信者だけの秘密

秘密鍵 (共通鍵) 暗号

暗号化鍵・復号鍵が同じ

- 換字暗号・Caesar 暗号
- 線型ブロック暗号
- Vernam 暗号
- DES (Data Encryption Standard)
- AES (Advances Encryption Standard)

例: Caesar 暗号

鍵： $n \in Z/26Z$

暗号化：alphabet を後ろに n だけずらす

復号：alphabet を前に n だけ戻す

$\cdots$ **XYZABCDEFGHIJKLMN**
OPQRSTUVWXYZABC $\cdots$

例: $n =$ **?** **:** **?????** $\longrightarrow$ **KHOOR**

例: Caesar 暗号

鍵： $n \in Z/26Z$

暗号化： alphabet を後ろに n だけずらす

復号： alphabet を前に n だけ戻す

$\cdots$ XYZABCDEFGHIJKLMN
OPQRSTUVWXYZABC $\cdots$

例: $n =$ 3 **：** HELLO $\longrightarrow$ KHOOR

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

- **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 (Deta Encryption Standard)**
- **AES (Advanced Encryption Standard)**

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

- **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 (Deta Encryption Standard)**
- **AES (Advanced Encryption Standard)**

秘密鍵 (共通鍵) 暗号の特徴

暗号化鍵・復号鍵が同じ

- **一般に原理は簡単で高速**
- **事前の鍵共有の必要**
- **通信相手毎に別の鍵が必要**

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
(1976〜77)

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
(1976〜77)

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**

$\longrightarrow$ 共通鍵暗号では実現が困難

$\longrightarrow$ 公開鍵暗号・鍵共有方式のアイデア
(1976〜77)

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
  $\longrightarrow$ 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
  $\longrightarrow$ 改竄・なり済ましの対策
  $\longrightarrow$ 否認防止の機能も持つ

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**
  $\longrightarrow$ 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**
  $\longrightarrow$ 改竄・なり済ましの対策
  $\longrightarrow$ 否認防止の機能も持つ

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**
  $\longrightarrow$ 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**
  $\longrightarrow$ 改竄・なり済ましの対策
  $\longrightarrow$ 否認防止の機能も持つ

公開鍵暗号

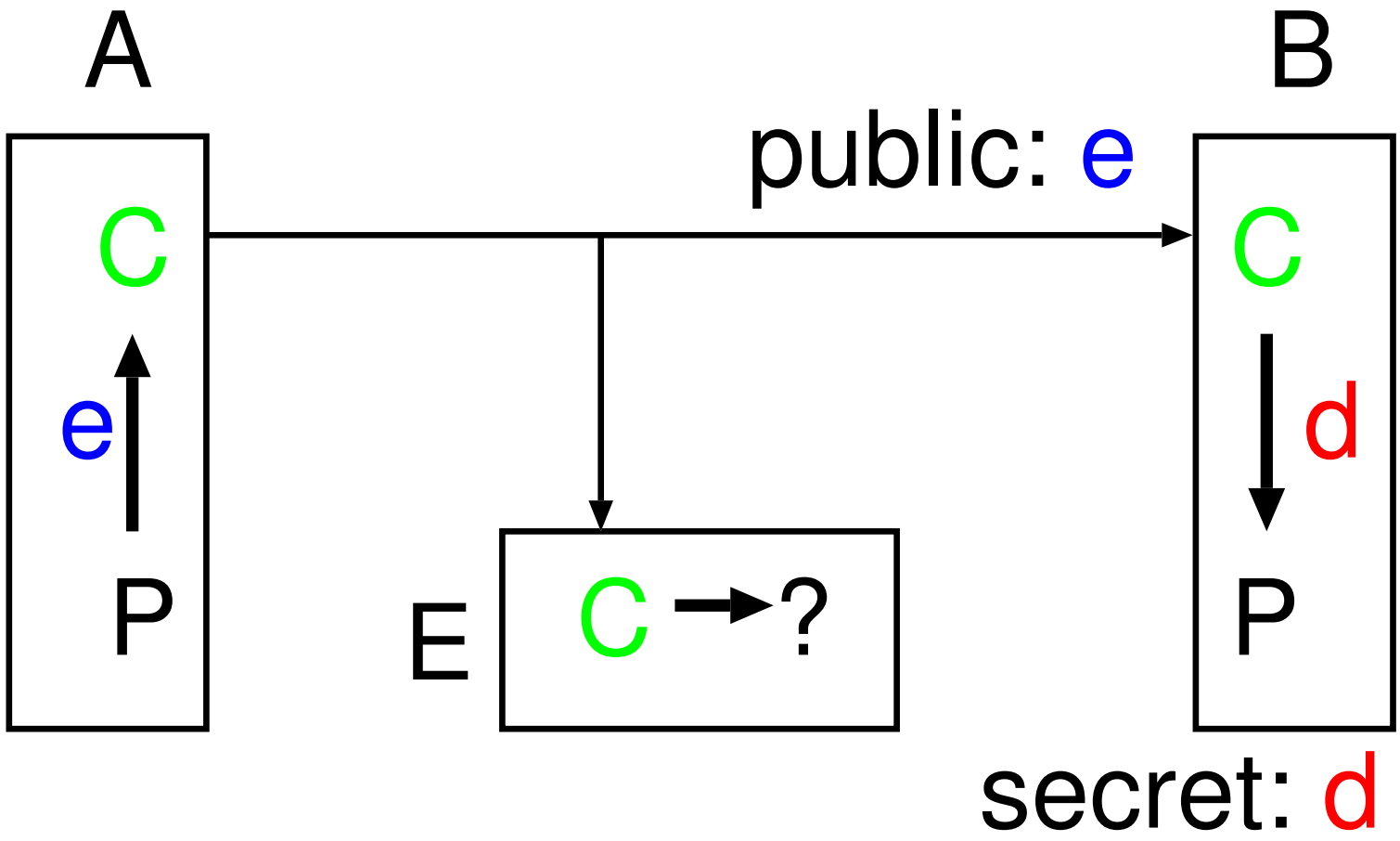
但し、一般には、
暗号化・復号が共通鍵暗号に比べて低速

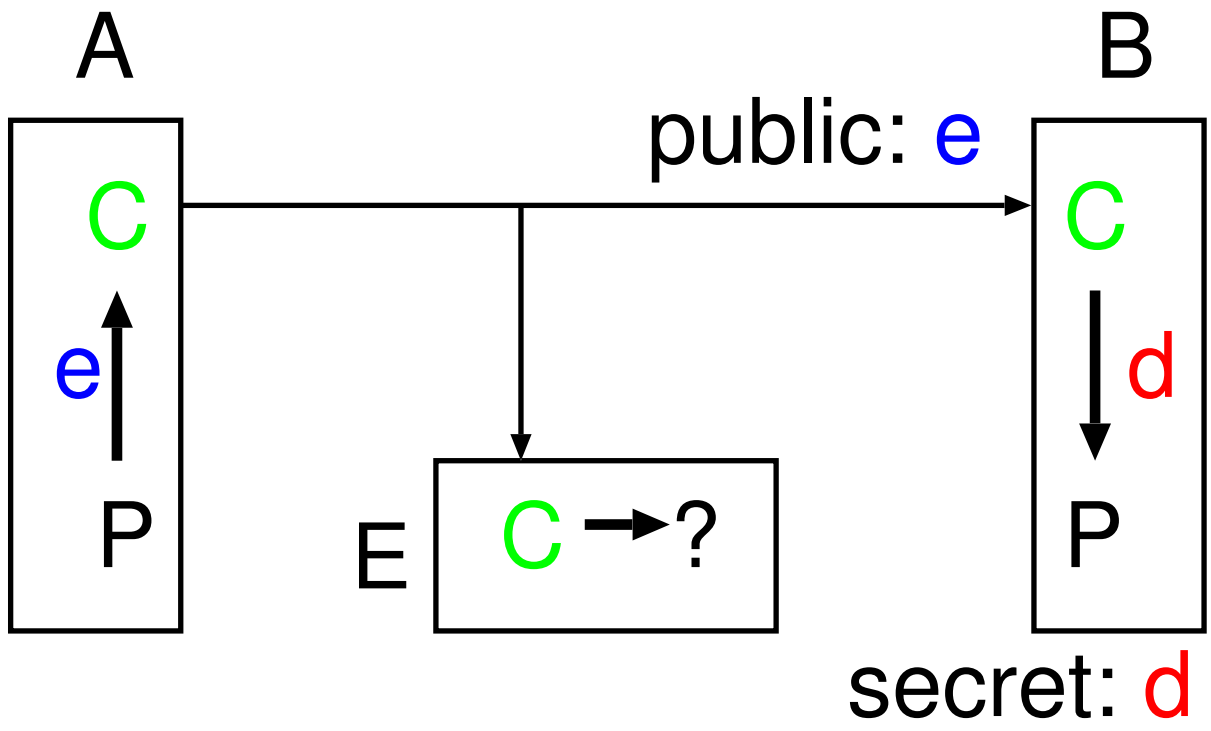
そこで、

- **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



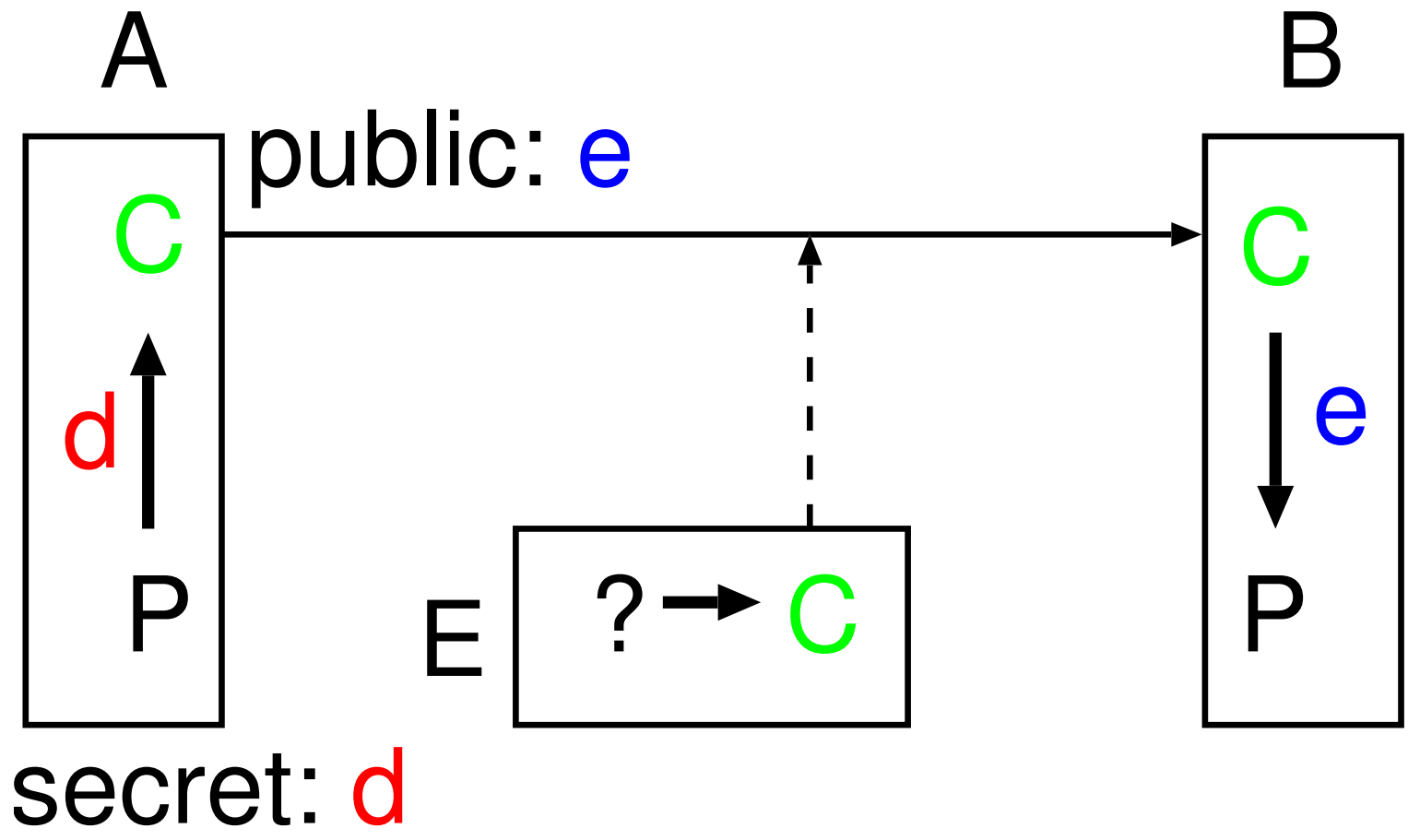


しかし、これだと誰でも暗号化できるので、

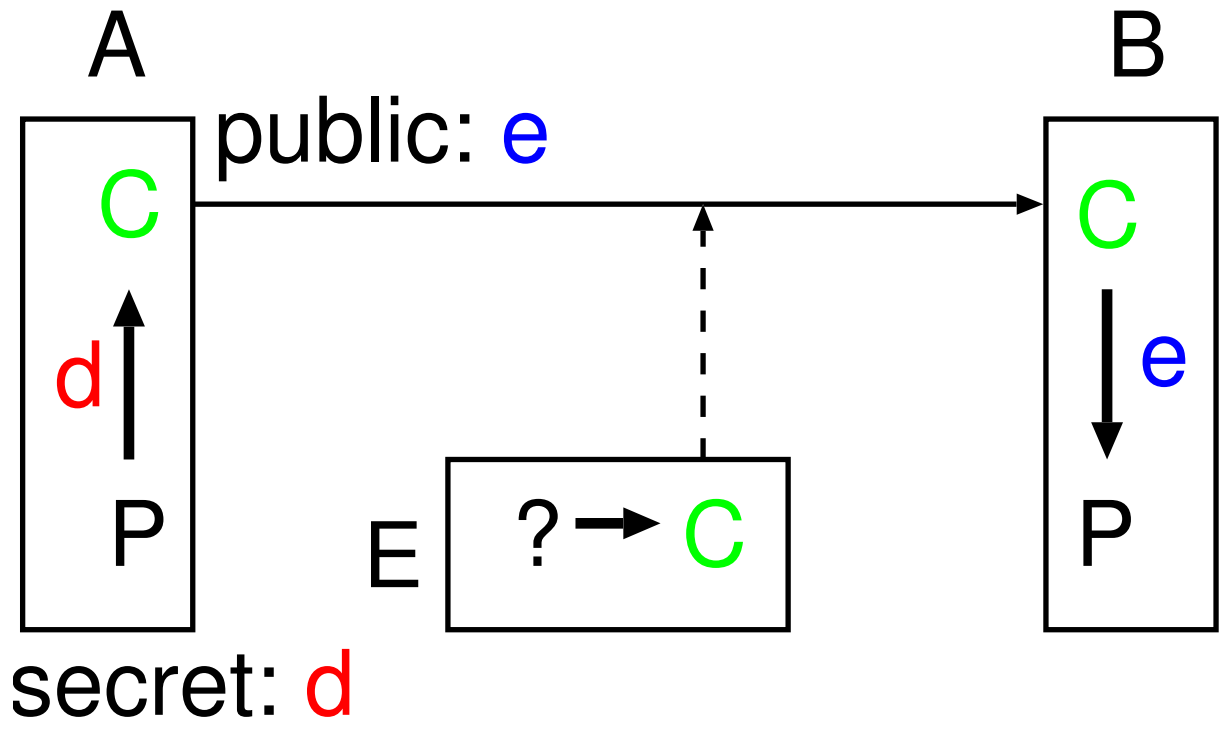
A 氏が送った保証がない

⟶ 署名の必要性

公開鍵暗号を用いた署名



公開鍵暗号を用いた署名



盗聴者 E 氏は

平文 P は判らないが、暗号文 C は盗聴可能

$\longrightarrow$ いつも同じ署名は使えない

公開鍵暗号を用いた署名

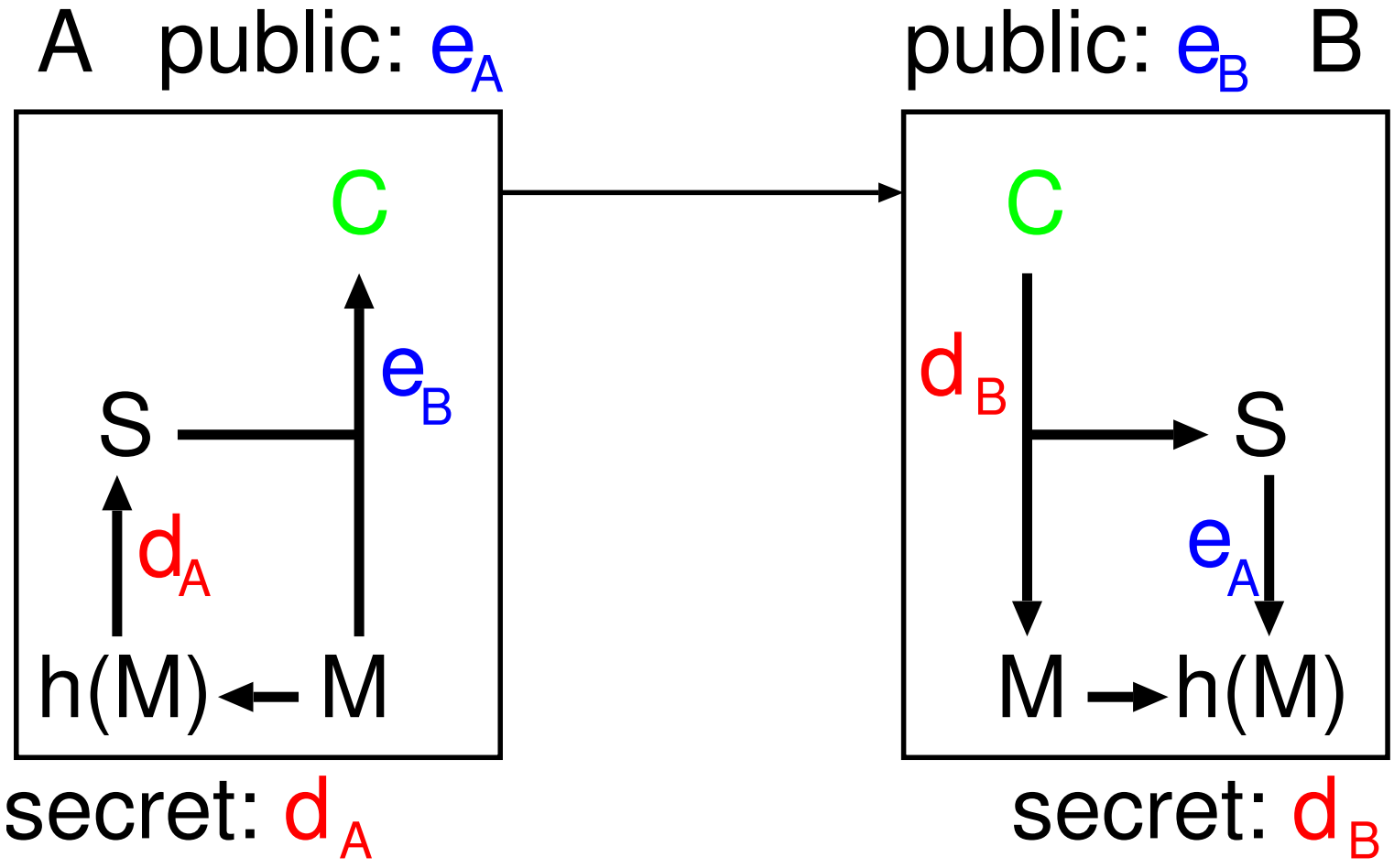
実際には、メッセージ本文 M に対して、

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 氏の公開鍵で一緒に暗号化して送る

公開鍵暗号を用いた署名



公開鍵暗号の特徴

- 暗号化は誰でも出来る
- 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 ？

$\longrightarrow$ ある!! (多分大丈夫)

計算困難な問題 を利用
(素因数分解・離散対数問題)

公開鍵暗号の特徴

- 暗号化は誰でも出来る
- 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 ？

$\longrightarrow$ ある!! (多分大丈夫)

計算困難な問題 を利用
(素因数分解・離散対数問題)

公開鍵暗号の特徴

- 暗号化は誰でも出来る
- 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 ？

$\longrightarrow$ ある!! (多分大丈夫)

計算困難な問題 を利用
(素因数分解・離散対数問題)

代表的な公開鍵暗号方式

- **RSA 暗号 (Rivest-Shamir-Adleman)**
- **Diffie-Hellman 鍵共有**
- **ElGamal 暗号**

代表的な公開鍵暗号方式

- **RSA 暗号 (Rivest-Shamir-Adleman)**
- **Diffie-Hellman 鍵共有**
- **ElGamal 暗号**

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 の素因子分解 $n = pq$ 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 (Rivest-Shamir-Adleman)

素因数分解の困難さを利用

p, q : **相異なる大きい素数**
(実際には現在は 512 **bit 程度)**

$N := pq$: **RSA 方式の法 (modulus)**

$m := \mathrm{lcm}(p-1, q-1)$

$$(\boldsymbol{Z}/N\boldsymbol{Z})^\times \simeq (\boldsymbol{Z}/p\boldsymbol{Z})^\times \times (\boldsymbol{Z}/q\boldsymbol{Z})^\times$$
$$\simeq \boldsymbol{Z}/(p-1)\boldsymbol{Z} \times \boldsymbol{Z}/(q-1)\boldsymbol{Z}$$

: 指数 m **の有限アーベル群**

RSA 暗号 (Rivest-Shamir-Adleman)

p, q：**相異なる大きい素数**

$N = pq,\ \ m = \mathrm{lcm}(p-1, q-1)$

e **を** $\gcd(e, m) = 1$ **となるように選ぶ**

d **を** $ed \equiv 1 \pmod{m}$ **となるように求める**

- (N, e)：**公開鍵 (暗号化鍵)**
- d：**秘密鍵 (復号鍵)**

RSA 暗号 (Rivest-Shamir-Adleman)

p, q：**相異なる大きい素数**

$$N = pq, m = \mathrm{lcm}(p-1, q-1)$$

$$ed \equiv 1 \pmod{m}$$

- (N, e)：**公開鍵 (暗号化鍵)**
- d：**秘密鍵 (復号鍵)**

平文 $M \bmod N$ **の暗号化：** $\boxed{C = M^e \bmod N}$

暗号文 $C \bmod N$ **の復号：** $\boxed{M = C^d \bmod N}$

RSA 暗号 (Rivest-Shamir-Adleman)

公開鍵 (N, e) から秘密鍵 d が計算できるか ？

- **N の素因数分解 $N = pq$ を知っていれば容易**
- **事実上 N の素因数分解と同程度の困難さ**

「困難さ」 $\cdots$ 計算時間が掛かる

RSA 暗号の安全性 $\Longleftrightarrow$ 素因数分解の困難さ

“計算量的安全性”

RSA 暗号 (Rivest-Shamir-Adleman)

公開鍵 (N, e) から秘密鍵 d が計算できるか ？

- **N の素因数分解 $N = pq$ を知っていれば容易**
- **事実上 N の素因数分解と同程度の困難さ**

「困難さ」 $\cdots$ 計算時間が掛かる

RSA 暗号の安全性 $\Longleftrightarrow$ 素因数分解の困難さ

“計算量的安全性”

RSA 暗号 (Rivest-Shamir-Adleman)

公開鍵 (N, e) から秘密鍵 d が計算できるか ？

- **N の素因数分解 $N = pq$ を知っていれば容易**
- **事実上 N の素因数分解と同程度の困難さ**

「困難さ」 $\cdots$ 計算時間が掛かる

RSA 暗号の安全性 $\Longleftrightarrow$ 素因数分解の困難さ

“計算量的安全性”