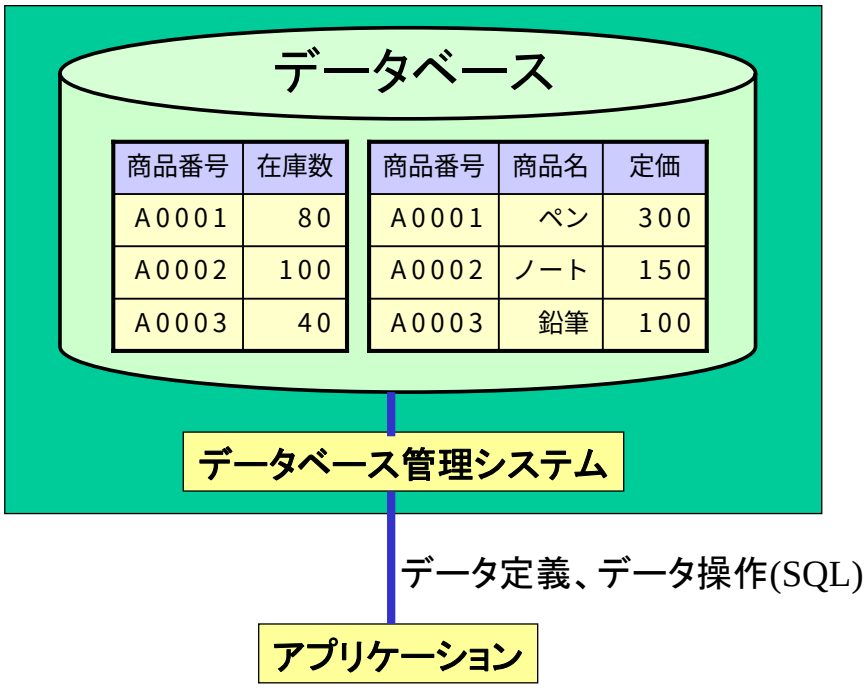
1.データベース

1-1 データベース

- 大量のデータを安全に扱うソフトウェア
- 利点
  - 大量、高速、安全、確実
- 特許
  - 歴史は古い→基本技術は枯れている
  - Webアプリケーションなどで必須
  - 新しい応用: 全文、ゲノム、XMLなど

1-2 関係データベース

- データを表で管理
  - Oracle、PostgreSQL、MySQL、Access…



1-3 データベース設計

- スキーマ: 表を設計
- 実世界のデータを表に対応付ける
  - 表
    - ➡名前
  - 項目
    - ➡名前
    - ➡データ型
  - キー
    - ➡主キー
    - ➡外部キー

```
CREATE TABLE 商品 (
        商品番号  CHAR(10)NOT NULL,
        商品名    CHAR(30),
        定価      DECIMAL,
        PRIMARY KEY (商品番号)
);
CREATE TABLE 製造 (
        製造番号  CHAR(10)NOT NULL,
        商品番号  CHAR(30),
        数量      DECIMAL,
        PRIMARY KEY (製造番号),
        FOREIGN KEY (商品番号) REFERENCES 商品;
);
```

1-4 データベース操作(1)

- 選択
  - データを選ぶ

SELECT * FROM 商品 WHERE 定価 < 200;

商品

商品番号	商品名	定価
A0001	ペン	300
A0002	ノート	150
A0003	鉛筆	100

商品番号	商品名	定価
A0002	ノート	150
A0003	鉛筆	100

- 射影
  - 項目を選ぶ

SELECT 商品名,定価 FROM 商品;

商品名	定価
ペン	300
ノート	150
鉛筆	100

1-5 データベース操作(2)

- 結合
  - 複数の表を操作

SELECT * FROM 商品,製造
WHERE 商品.商品番号 = 製造.商品番号;

商品

商品番号	商品名	定価
A0001	ペン	300
A0002	ノート	150
A0004	鉛筆	100

製造

商品番号	メーカー
A0001	コクヨ
A0003	三菱
A0004	ゼブラ

商品番号	商品名	定価	メーカー
A0001	ペン	300	コクヨ
A0004	鉛筆	100	ゼブラ

- その他
  - 挿入、更新、削除など

INSERT INTO 商品 VALUES ('A0010', '定規', '400');

1-6 Excelではダメなのか？

- 年賀状印刷の住所管理なら十分
- 難しいのは…
  - 複雑な構造、大量のデータ、プログラムから操作、高速な検索、他人と共有など

	ファイル	データベース
構造	1テーブル	多テーブル
適したデータ量	少量	大量
操作方法	自前	SQL
速度	大量だと遅い	大量でも速い
同時アクセス	困難	容易

1-7 索引と計算量

- データベース: 大量のデータ
  - データ量が増えると質が変わる
  - データ数と処理時間を表にした
- 検索

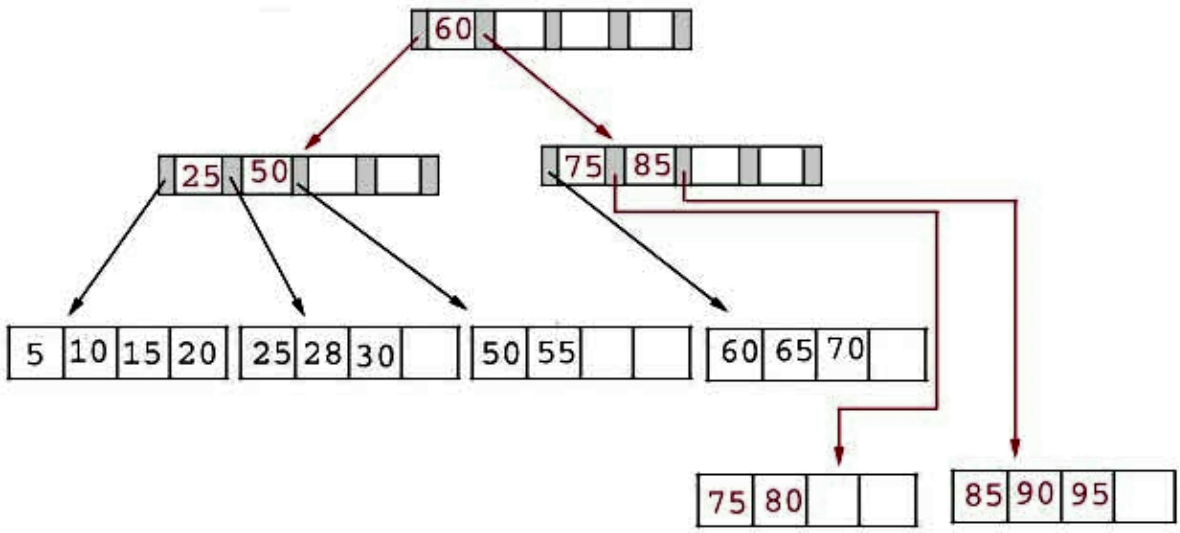
データ数 (N)	リニア検索 (N)	索引検索 (logN)
10	10	3
1000	1,000	10
100万	1,000,000	20

- ソート

データ数 (N)	並べ替え (NxN)	並べ替え (NlogN)
10	100	30
1000	1,000,000	10,000
100万	1,000,000,000,000	20,000,000

1-8 索引(1)

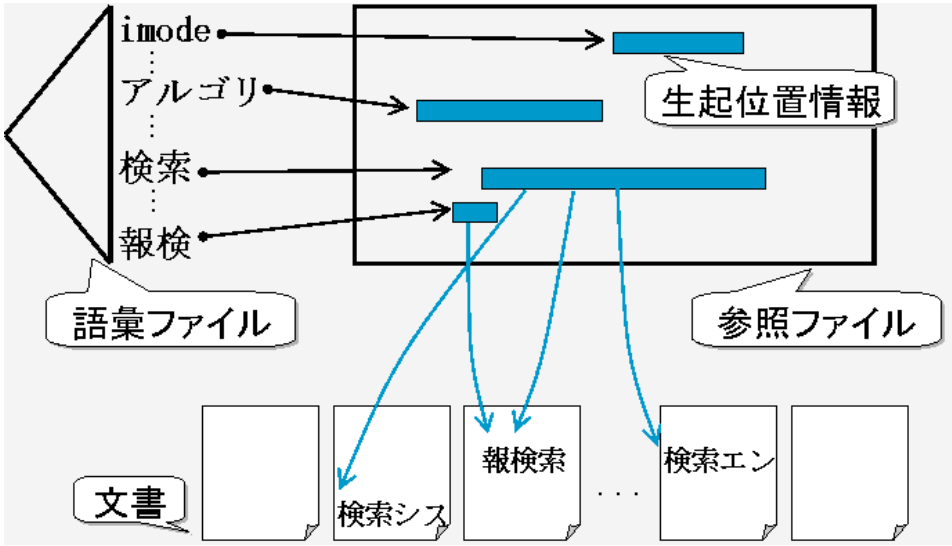
- B+木
  - 数値、文字列(前方/後方一致)に使う



http://babbage.clarku.edu/~achou/cs160/B+Trees/B+Trees.htm

1-9 索引(2)

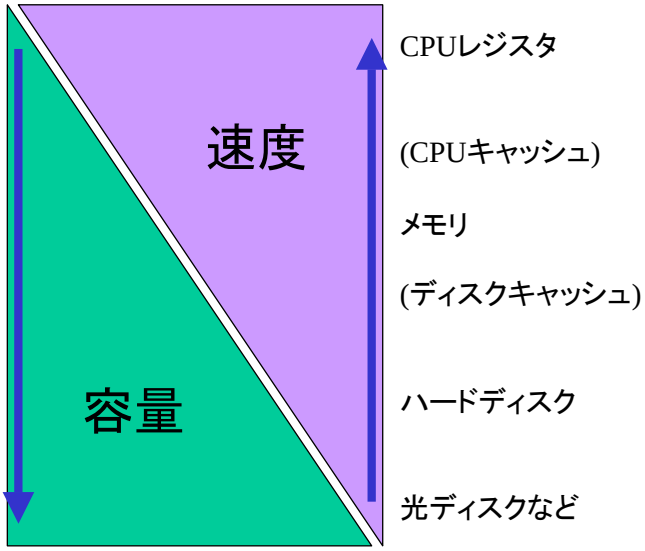
- 転置索引
  - 全文検索に使う



http://www.ingrid.org/~harada/publications/SIGNL136/

1-10 記憶装置

- 種類
  - レジスタ(CPU)
  - 主記憶(メモリ)
  - 補助記憶装置(ハードディスク)
  - 大容量記憶装置(光ディスクなど)
- キャッシュ
  - 速度差を補間(100万倍以上!)



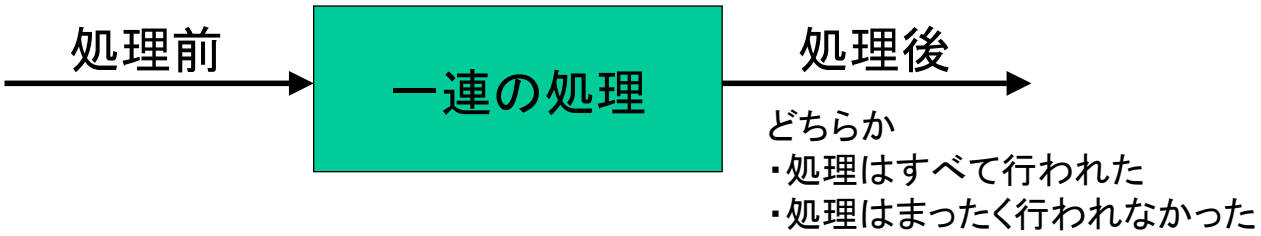
1-11 排他制御

- 資源の競合
  - 1冊の台帳にみんなで読み書きすると…?
  - 読むだけなら同時でも構わないが、読んだものを使って書くと…?
- 銀行振込の例
  - 2人が同じ口座に振込
  - 2万円の行方は?
- 排他制御
  - ロック
  - 単位: テーブル、レコード、フィールド、…

Aさんの処理	Bさんの処理	口座
読み出し(10万円)		10万円
加算(10+2=12万円)	読み出し(10万円)	10万円
書き込み(12万円)	加算(10+3=13万円)	12万円
	書き込み(13万円)	13万円

1-12 トランザクション

- ひとまとまりの処理を
  - 「行った」か「行わなかった」状態にする
    (中間の状態はない)
  - (例)送金処理
    口座1から減算、口座2に加算



- デッドロック(すくみ)
  - A→B、B→A

1-13 障害回復

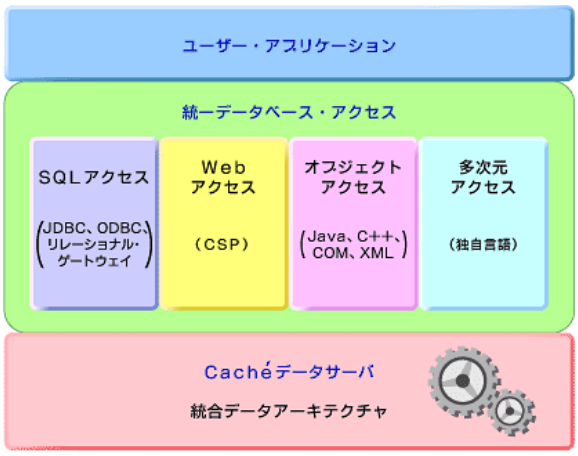
- 処理の確定(commit)中に電源が落ちたら？
  - データベースに矛盾
    - ➡口座1から減額(書き込み成功)
    - ➡口座2に加算(書き込み中に停止)
- 障害回復処理
  - 処理を別ファイルに記録(ジャーナル、ログ)
  - ジャーナルから修復する
- 装置でも補完
  - ディスクの二重化(RAID)
  - バックアップ装置
  - 稼動中のバックアップ

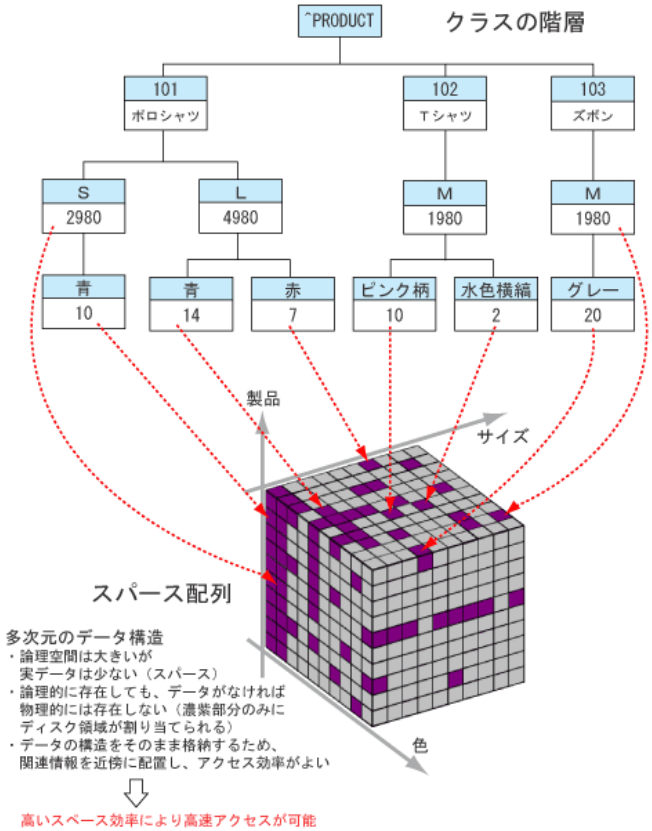
1.14 オブジェクト指向データベース

■ オブジェクトをそのまま格納可能

http://www.atmarkit.co.jp/fdb/rensai/oodbrevive01/oodbrevive01_1.html

1.15 多次元データベース





http://www.atmarkit.co.jp/fdb/rensai/oodbrevive01/oodbrevive01_2.html

http://www.atmarkit.co.jp/fxml/tanpatsu/35oodb/oodb01.html

2.XML

2-1 インターネット以前のデータ交換

- データファイル
  - プログラムが書いてプログラムが読む
  - 専用形式(一太郎、ワード、エクセル...)
  - 読めるのは同じ(または同じ会社の)ソフト
- 企業戦略
  - 昔は「独占」！
  - データ交換: 範囲が狭い(友人に手渡し、社内)
  - 独自ファイル: 過去の資産が乗換えを抑止

2-2 出版

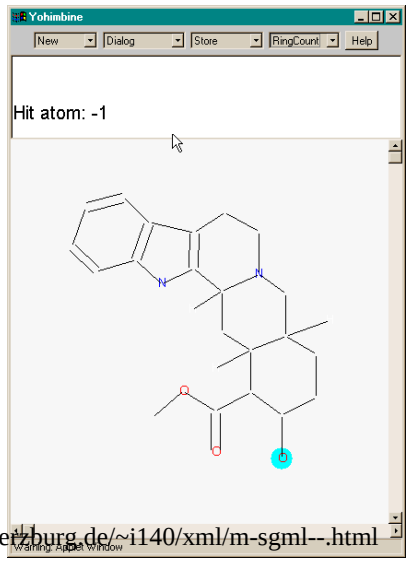
- DocBookなど
- 出版のための原稿を記述する

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
"http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd" >
<book>
  <title>文書タイトル</title>
  <chapter>
    <title>章タイトル</title>
    <section>
      <title>節タイトル</title>
      <para> 我輩は猫である。</para>
      <para> 名前はまだない。</para>
    </section>
  </chapter>
</book>
```

2-3 その他の応用

- SVG: ベクターグラフィックス
- SMIL: 動画の同期を記述
  - 音声に合わせてスライド表示など
- MPEG-7: 画像に説明文を付与
- 数式(MathML)
- 化学式(CML)

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



http://www.w3.org/TR/REC-MathML/chapter2.html

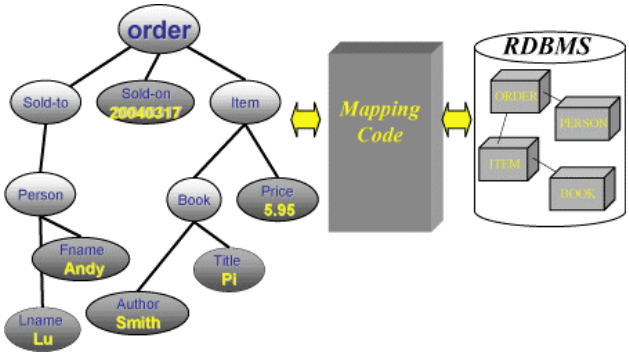
http://www.studi.fh-wuerzburg.de/~i140/xml/m-sgml--.html

2-4 メタデータ

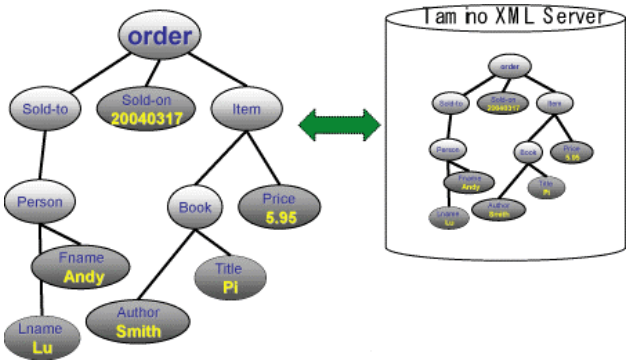
- データを説明するデータ
  - 記事の作者、日付、要約など

2-5 XMLデータベース

- 二種類の方式。一長一短
- RDB型
  - 表に分解する
  - 分解・組立
  - 項目検索速い



- ネイティブXML型
  - そのまま入れる
  - 分解・組立不要
  - 入出力速い



http://www.beacon-it.co.jp/products/pro_serv/e-business/tamino/tamino1.shtml

3.認証とセキュリティ

3-1 セキュリティ

(情報インフラ)

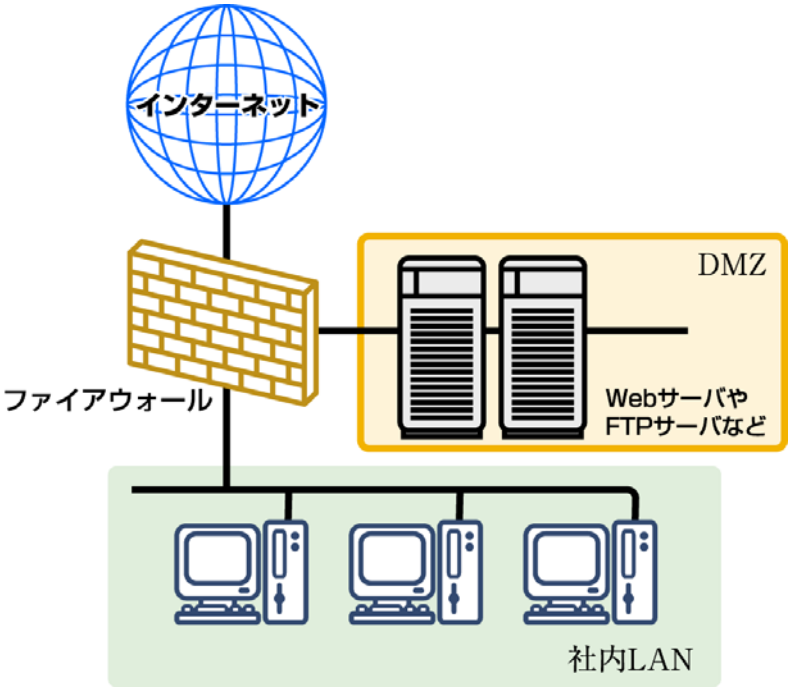
- 端末
  - ID/PW (+物理/生体的な特徴) で本人確認
- サーバー
  - 侵入の防御、データの暗号化
- 通信
  - 漏洩、改ざんなどの防止

(インターネットの商取引)

- 情報インフラに加えて対策が必要
  →今回の話題

3-2 ファイアウォールとDMZ

- DMZ: 非武装地帯(DeMilitarized Zone)



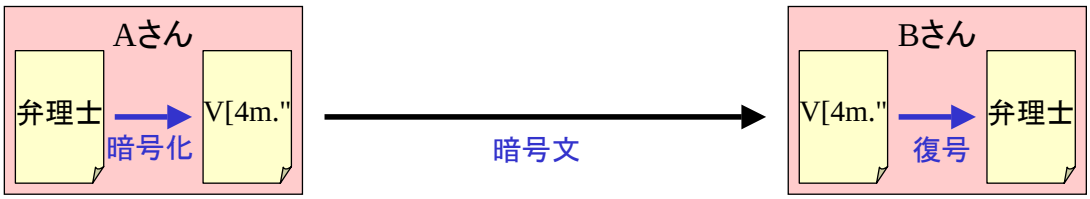
http://premium.nikkeibp.co.jp/security/special/index02_02_03.shtml

3-3 インターネット取引の危険要素

- 他人に情報が漏れていないか(漏洩)
  - 取引内容、顧客情報が流出
- 通信が書き換えられていないか(改ざん)
  - 1万円の商品で100万円の請求が来た
- 相手は本人か(なりすまし)
  - 送金したが商品が来ない
  - 商品を送ったが入金されない
- 取引を否定されることはないか(否認)
  - 商品を送ったら注文していないと言われた

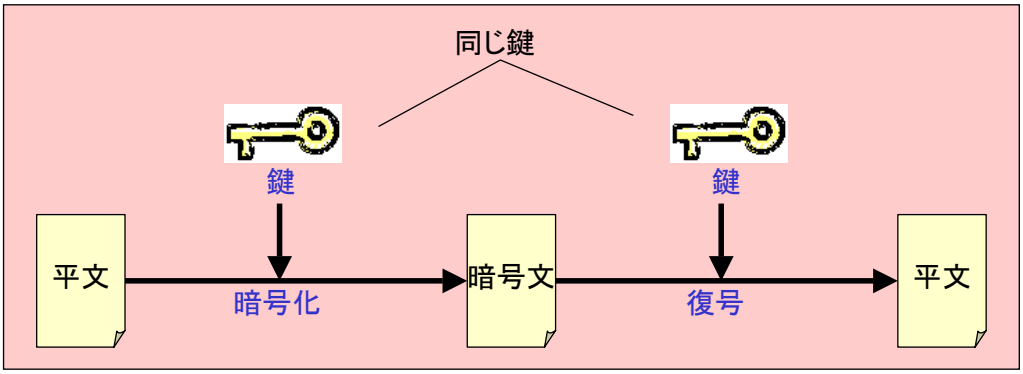
3-4 暗号

- 他人に読めなくする
- メッセージを暗号化して送る
  - 暗号文が漏れても解読できない(漏洩対策)
  - 暗号アルゴリズム: 複数の方式が存在
  - 鍵: 意味不明に見える文字列。総当りで実用的な時間内に解けない長さ



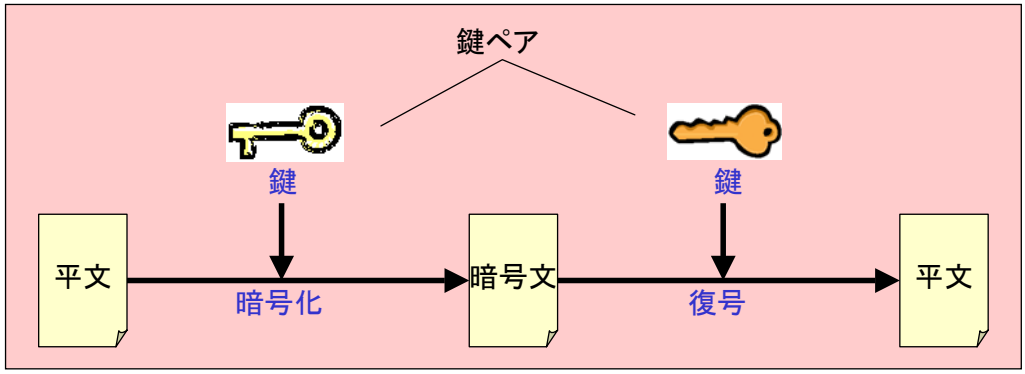
3-5 共通鍵暗号

- 特徴
  - 暗号化と復号に同じ鍵を使う
  - 速度は速い
  - 鍵の安全な受け渡しが必要
  - DES、RC5、AESなど



3-6 公開鍵暗号

- 特徴
  - 暗号化と復号に違う鍵を使う
  - 公開鍵は通信する相手に渡しておく(公開)
  - 秘密鍵は本人だけが持つ(秘密)
  - ペアの鍵で復号可能(公開→秘密、秘密→公開)
  - RSA、DSAなど

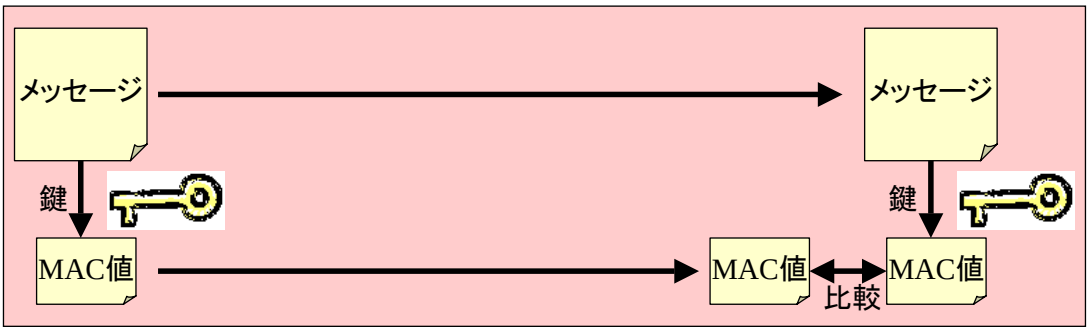


3-7 一方向ハッシュ関数

- 文書の指紋
  - 文書の特徴を数文字で表現
  - 文書が1文字でも変われば全く違う値
    →改ざんの検出
  - 元文書を復元不可能。特定の値を作れない
- 応用
  - 文書、ソフトウェアの改ざん検出
  - パスワードの暗号化
  - メッセージ認証コード
  - デジタル証明書

3-8 メッセージ認証コード

- メッセージ認証コード(MAC)
  - 「メッセージ+鍵」からハッシュ値を計算
  - 共通鍵を知らないとMAC値を作れない
    →改ざん防止
  - 共有鍵を安全に受け渡しておく必要がある



3-9 認証

- 本人であることを証明
- 本人しか知らない情報/特徴を使う
  - 記憶(パスワード)
  - 持ち物(ICカード)
  - 生体情報(指紋、網膜、…)
  - データ(秘密鍵)
- デジタル署名
  - 本人しか知らないデータ(秘密鍵)で暗号化
  - 公開鍵で復号できたら作った本人とわかる
  - 利用例
    - 公式なアナウンス(JPCERT/CCなど)
    - ソフトウェアの配布(署名付きアプレット)

3-10 否認

- 自分が送ったメッセージを否定すること
(それは私のメッセージではない)
  - 公開鍵による暗号化メール: 誰でも送れる
  - メッセージ認証コード: 相手も作れる
  - デジタル署名: 本人だけが作れる

- デジタル署名でよいか
  - 公開鍵を作った本人
  - それは本当に氏名まで含めた本人？

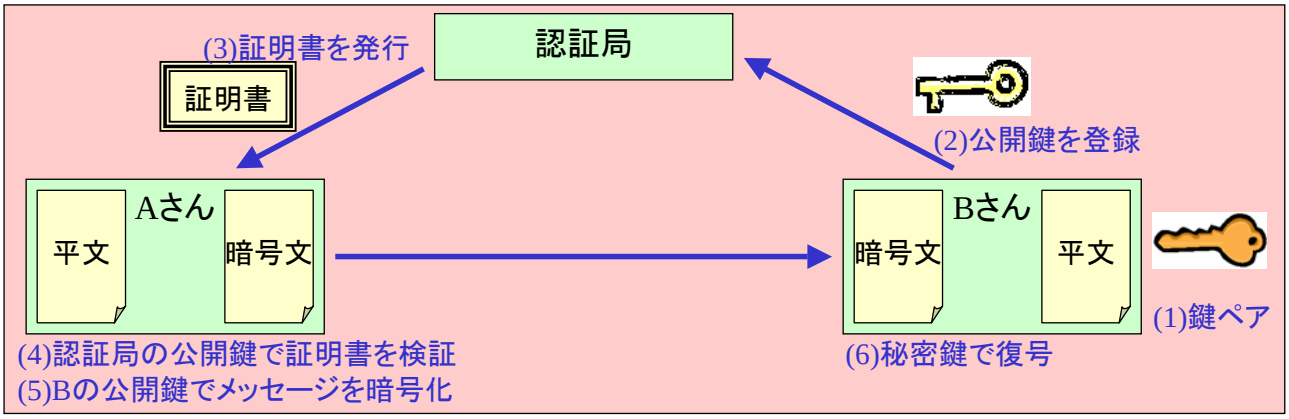
3-11 証明書

- 電子的な証明書
  - 個人情報、公開鍵、認証局のデジタル署名
  - 運転免許証と似ている
- 認証局(ＣＡ)
  - 証明書を登録・発行する

証明書
個人情報:
(氏名、所属、メールアドレスなど)
公開鍵
認証局のデジタル署名

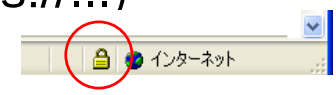
3-12 PKI

- 認証局(ＣＡ)
  - 誰でも作れる。どこまで信じるか(学生証は?)
  - 他の認証局に認証してもらう→ＣＡの階層
- 公開鍵基盤(ＰＫＩ)



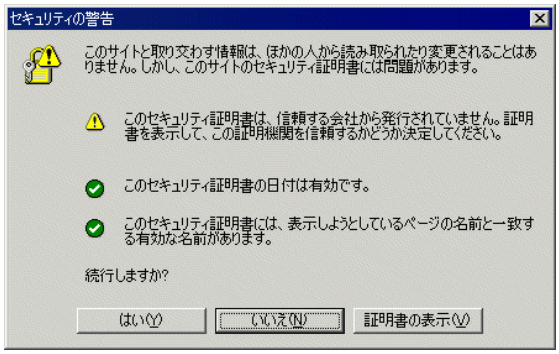
3-13 Webへの応用例

- ＳＳＬ
  - Webの認証/暗号化通信技術(https://...)
  - 個人情報、商取引などに利用
  - 公開鍵+共通鍵を組み合わせている



- オレオレ証明書(仮称)

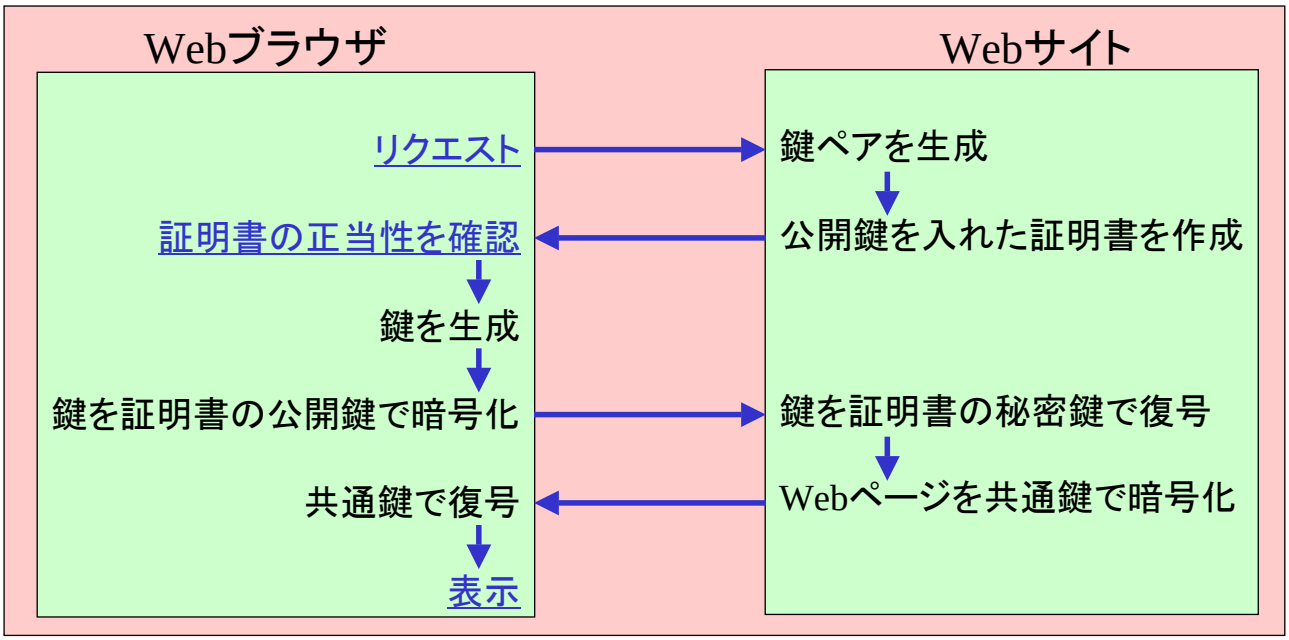
【クイズ】あるWebサイトに、
セキュリティの警告メッセージが表示されることがありますが、暗号化は正常に行われますので、「はい」でお進みください。
という案内文が書かれていたとします。そこにある「住所の入力」というリンクをクリックすると、右のような警告が表示されました。
(1) 案内文の前半にある「暗号化は正常に行われます」は正しいでしょうか。
(2) もしも「はい」で進み、住所を入力したとすると、どんな危険が考えられるでしょうか。



http://www.hyuki.com/diary/

3-14 SSLの仕組み

- 下線部はユーザーに見える処理



まとめ

- こんなことを学びました
  - データベース
  - XML
  - セキュリティ

総まとめ

- 今日は、こんなことを学びました
  - 計算機の構成、プログラミング
  - Webアプリケーション技術
  - データベース、セキュリティ

- 実践的な取り組み(次回以降)
  - オブジェクト指向設計
  - モデリング、インタビューなど

お楽しみに！