Internet Week '99 チュートリアル

暗号化/認証技術とその応用

稲村 雄

jane@vicus-oryzae.com / inamura@verisign.co.jp

日本ベリサイン (株) マーケティング部

構成

- ***暗号化技術概説***
- **認証技術の発展**
- **実プロトコルでの利用形態**

暗号化技術概説

ユリウス・カエサルの昔より

- ユリウス・カエサル
  - 共和政ローマ末期の政治家/将軍/文学者 etc.
- カエサルが知人に宛てた手紙を託す使者を信頼できなかった時に、暗号通信の歴史は始まった。

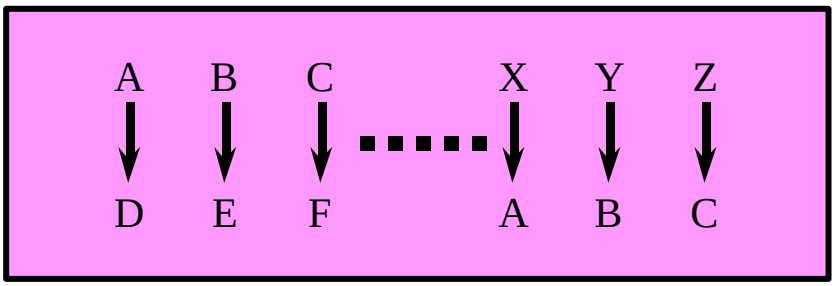
Gaius Julius Caesar
(B.C. 100 - B.C. 44)

http://www.uni-paderborn.de/Admin/corona/chris/Caesar_0.html より

暗号化技術概説

カエサル暗号

- 単純な**換字式**暗号
  - →現代ではほとんど実際の用に足りない
- NetNews などで利用例も



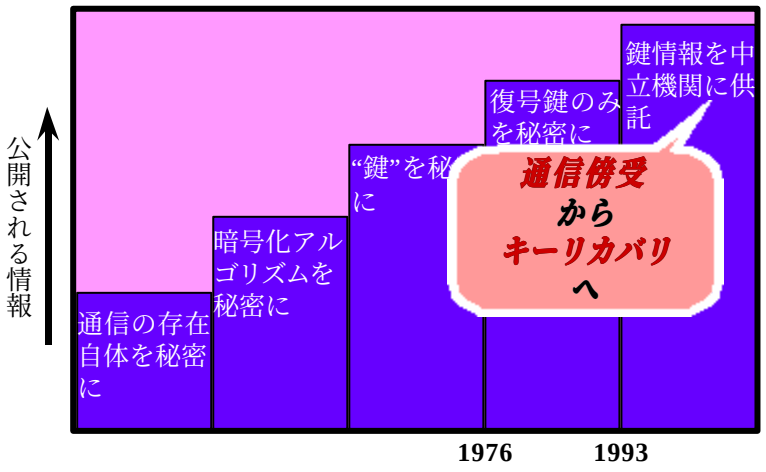
暗号化技術概説

信頼のおけない通信路を介して、如何に安全に？

- 二千年に渡る難題
- インターネットは現代の代表例
  - » 元来、研究者向けのネットワークなので、非常に性善説的
  - » 広大すぎて、誰も把握しきれていない

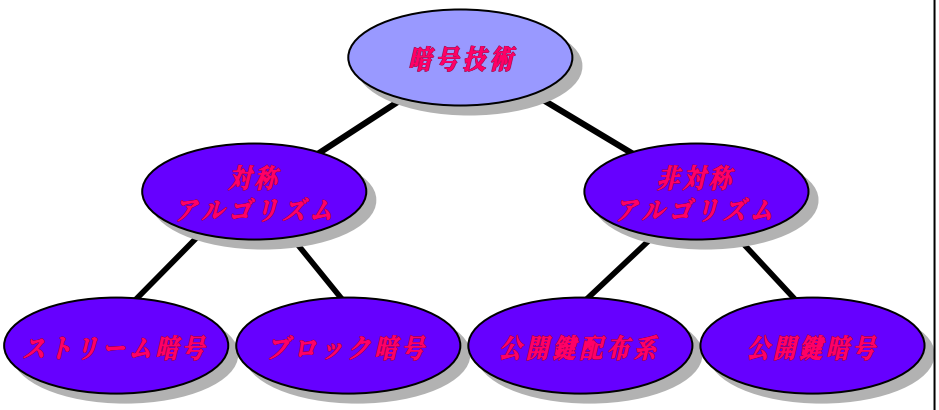
暗号化技術概説

暗号化技術の発達



暗号化技術
概説

暗号技術概要



暗号化技術
概説

暗号技術

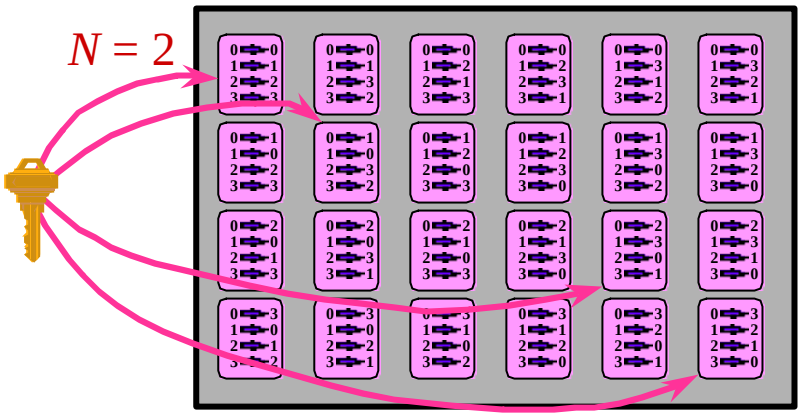
- **通信文など (= 平文) を第三者には意味不明な形 (= 暗号文) に変換することで、当事者以外にとっての有用性を失わせしめるための技術**
- 要は“**可逆な**”データ変換技術
  - ☑ 平文空間が N ビットならば $2^N!$ 通り

平文となり得るデータの集合

暗号化技術概説

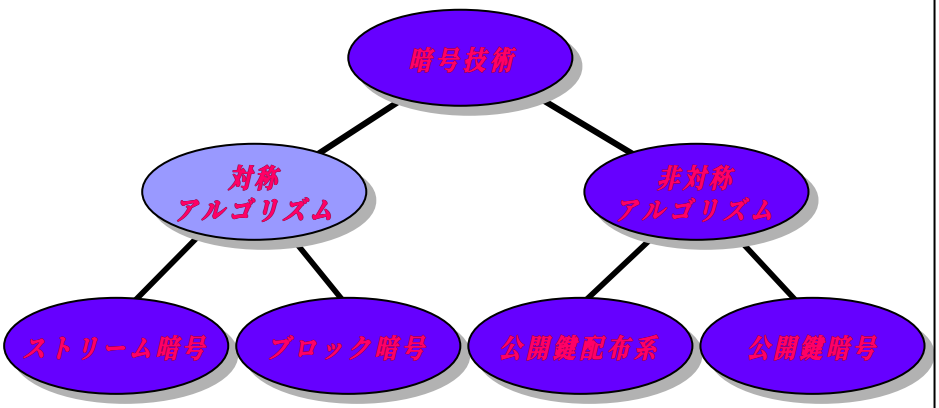
“可逆な”データ変換技術

- 平文空間が N ビットならば 2^N！通り



暗号化技術概説

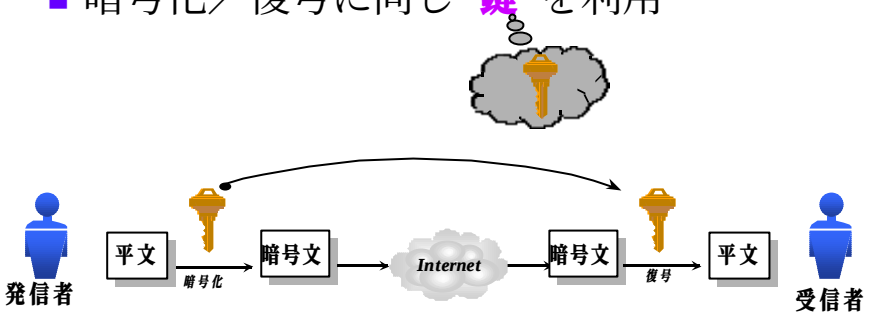
暗号技術概要



暗号化技術概説

対称アルゴリズム

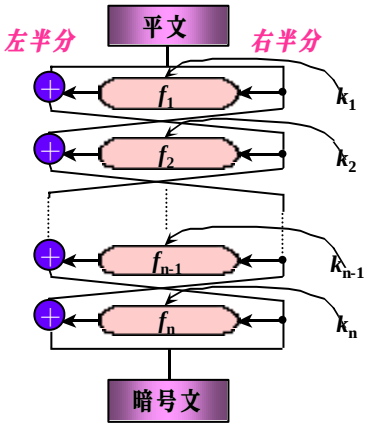
- ＝秘密鍵暗号／共通鍵暗号／慣用暗号
- 暗号化／復号に同じ**“鍵”**を利用



暗号化技術概説

対称アルゴリズム (2)

- ***Confusion* (混乱) & *Diffusion* (拡散)**
  - » ***Shannon*** の情報理論に基づく概念
  - » **置換**操作による混乱と**転置**操作による拡散
  - » SPN ***(Substitution-Permutation Network)***
    - ☒ 効果的に ***C-D*** を実現

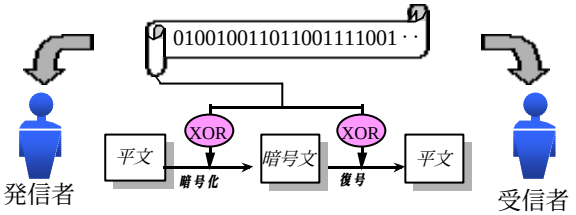


Feistel Network (SPNの実現)

暗号化技術概説

対称アルゴリズム (3)

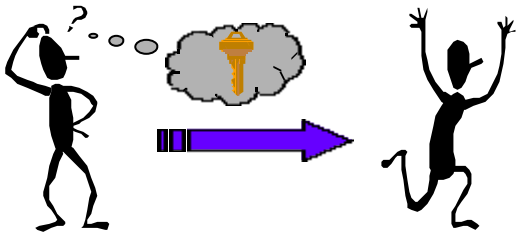
- 絶対に破られない暗号
  - **= *Onetime Pad***
  - メッセージと等長の乱数鍵を利用
  - 数学的に**安全性が証明されている**暗号方式



暗号化技術概説

対称アルゴリズムの問題点

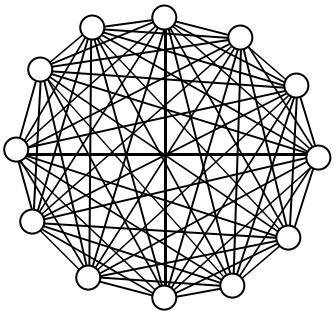
- 基本的方式として、鍵と平文／暗号文との間で複雑な演算を行なうことで暗号化／復号を実現
- ブロック暗号とストリーム暗号の二種類に大別
- **難点 1:** 鍵の安全な共有方法が大問題



暗号化技術概説

対称アルゴリズムの問題点

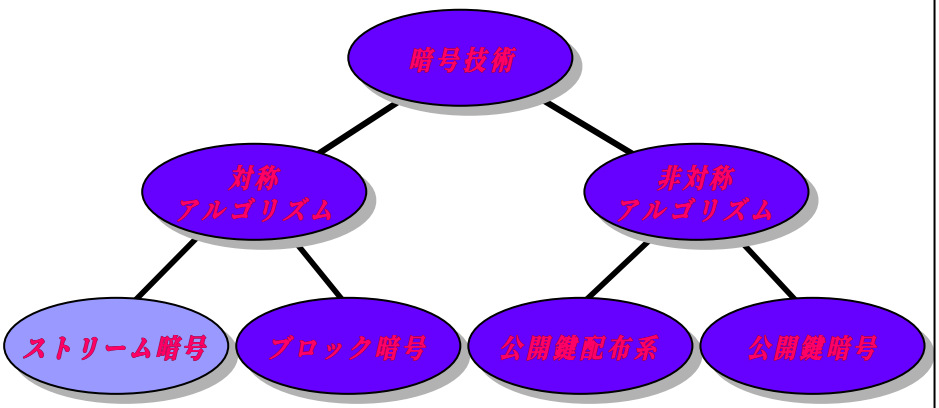
- **難点 2:** 相通信するペア毎に異なる鍵が必要
  - ⇒ 必要な鍵数の爆発



$$\frac{N(N-1)}{2}$$

暗号化技術概説

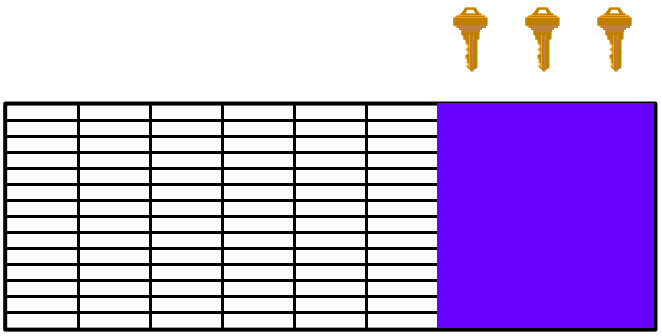
暗号技術概要



暗号化技術概説

ストリーム暗号

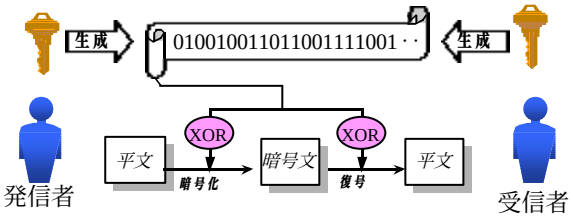
- 平文／暗号文を頭から順番に処理
  - 処理単位は bit/byte/word など、いろいろ



暗号化技術概説

ストリーム暗号 (2)

- 送信者／受信者ともに、鍵から擬似乱数列を生成
  - ***Onetime Pad*** の簡易版



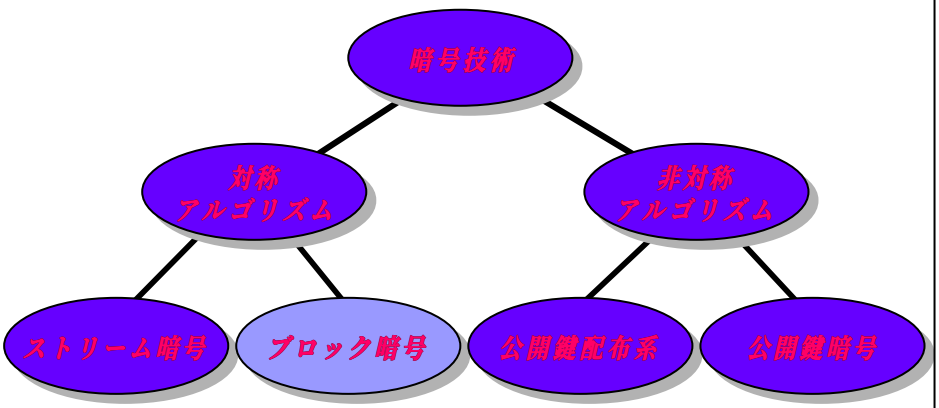
暗号化技術概説

ストリーム暗号 (3)

- RC4
  - ***Ron Rivest (RSA* の *R)*** が開発
  - 可変長鍵の利用が可能
  - **SSL *(Secure Sockets Layer)*** でのデフォルト
  - 本来は非公開アルゴリズムだったが…
- SEAL
- WAKE

暗号化技術概説

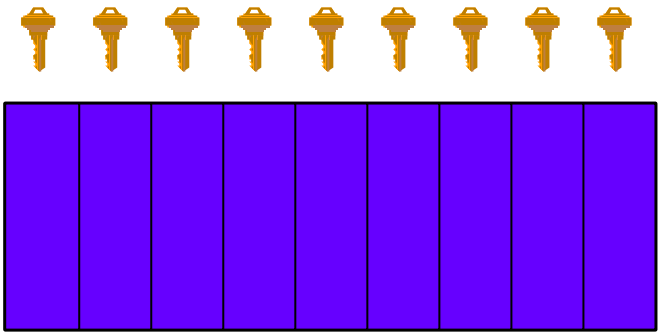
暗号技術概要



暗号化技術概説

ブロック暗号

- 平文／暗号文を一定サイズのブロックに分割して処理



暗号化技術概説

ブロック暗号 (2)

- ***DES*** ***(Data Encryption Standard)***
  - 米国 NBS (現 NIST) の公募に応じた IBM 提案に基づく暗号化アルゴリズム
  - 1976 年以来、標準的な暗号方式として利用が進む
  - 鍵長の短さのため、そろそろ寿命が尽きつつある
    - RSA 社による DES Challenge
    - AES ***(Advanced Encryption Standard)*** の公募開始

暗号化技術概説

ブロック暗号 (3)

- ***AES*** ***(Advanced Encryption Standard)***
  - DES に代わるアルゴリズムとして米国 NIST が 1997 年 9 月から公募を開始。
  - ***http://csrc.nist.gov/encryption/aes/aes_home.htm***
  - 最低条件:
    - 対称鍵暗号
    - ブロック暗号
    - 鍵-ブロック長として128-128, 192-128, 256-128 の組み合わせをサポート

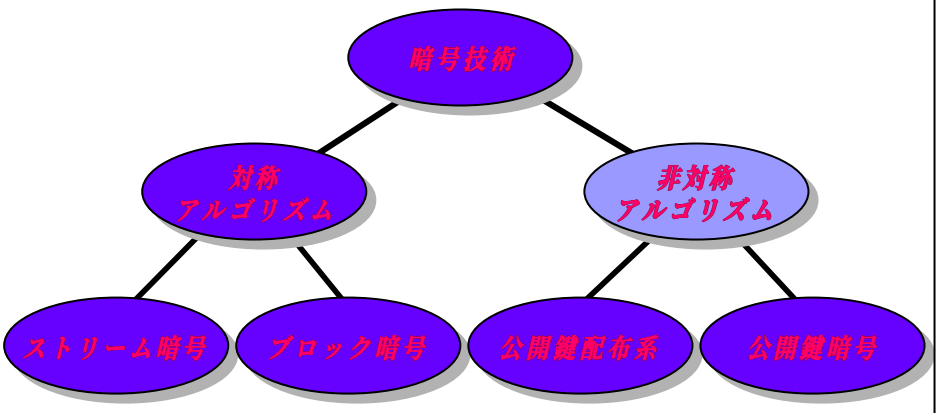
暗号化技術概説

AES 候補アルゴリズム

- ***CAST-256***
- ***CRYPTON***
- ***DEAL***
- ***DFC***
- ***E2***
- ***FROG***
- ***HPC***
- ***LOKI97***
- ***MAGENTA***
- ***MARS***
- ***RC6***
- ***RIJNDAEL***
- ***SAFER+***
- ***SERPENT***
- ***TWOFISH***

暗号化技術概説

暗号技術概要



暗号化技術概説

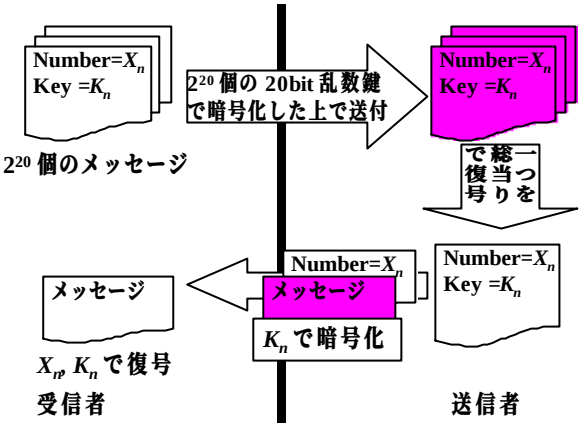
非対称アルゴリズム

- 1976 年に ***W.Diffie*** と ***M.Hellman*** が考案
  - **パブリックには。**
  - 1966 年 (*NSA*)、1970 年 (*CESG*) といった主張がある
- 基本は落とし戸 (*Trapdoor*) 付一方向関数
  - 片方向への計算は誰にでもできるが、逆方向の計算は非常に困難
  - 特別な知識 (= *Trapdoor*) を持った者は逆方向の計算ができる

暗号化技術概説

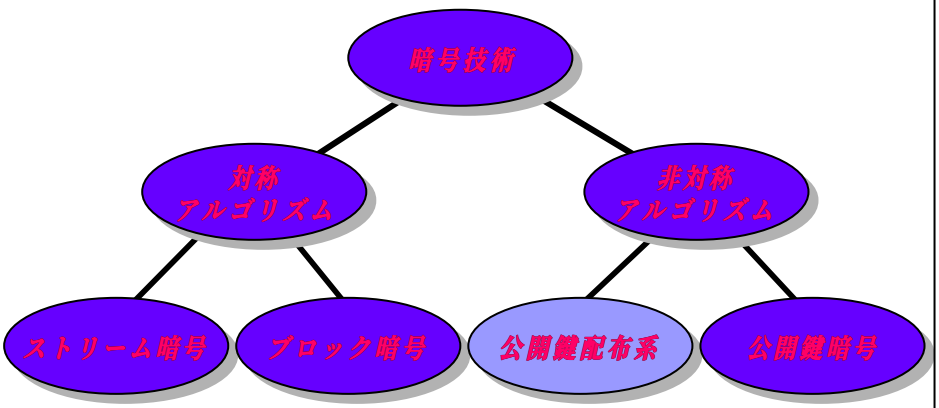
非対称アルゴリズムの先駆け

- ***R. Markle*** の提案 (1974)



暗号化技術概説

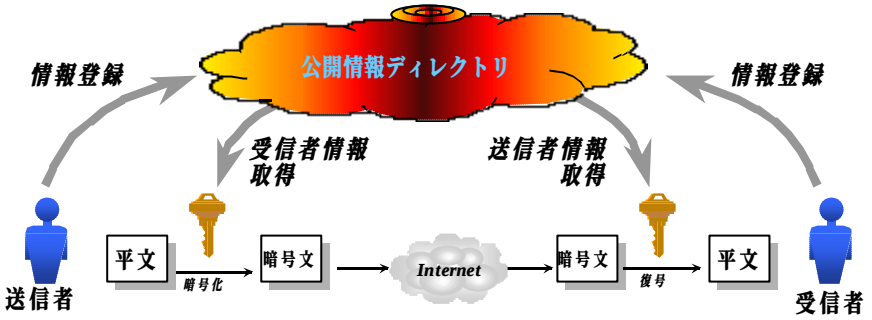
暗号技術概要



暗号化技術概説

公開鍵配布系

- **対称暗号の補助的役割**
  - » **各ユーザが公開している情報から対称暗号方式で用いられる共通鍵を生成する**



暗号化技術概説

公開鍵配布系 (2)

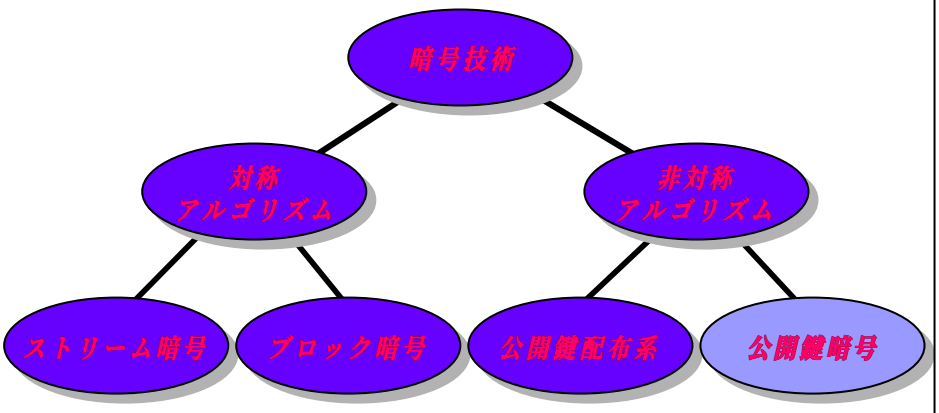
- ***Diffie-Hellman*** 方式
  - » ***W.Diffie*** と ***M.Hellman*** によって考案された世界初のアルゴリズム (1976)
  - » 離散対数問題の困難性に依拠

離散対数問題：

a, q, y が既知整数のとき、 $y = a^x \pmod q$ を満たすような整数 x を見付ける

暗号化技術概説

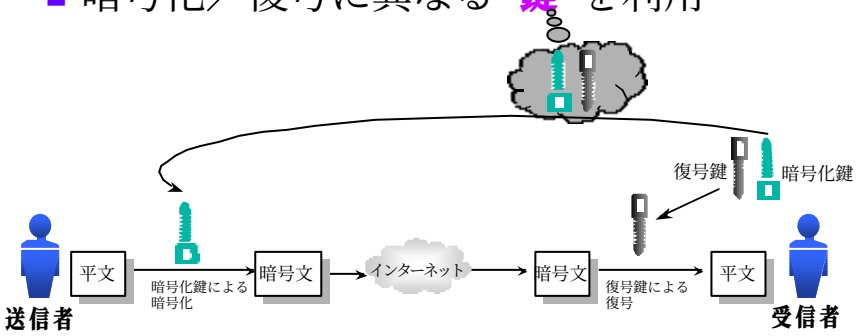
暗号技術概要



暗号化技術概説

公開鍵暗号

- ***W.Diffie*** と ***M.Hellman*** のアイデア (1976)
- 暗号化／復号に異なる**“鍵”**を利用



暗号化技術概説

公開鍵暗号 (2)

- 本システムのユーザは数学的に特殊な関係にある二種類の**“鍵”**を生成
  - 一方の鍵で暗号化されたデータは他方の鍵でしか復号できない
  - 一方の鍵データのみから他方の鍵を導き出すことは著しく困難
- 一つの鍵 (暗号化用鍵) を公開し、残り (復号用鍵) を秘密に保持する

暗号化技術概説

公開鍵暗号 (3)

- **対称暗号との比較**
- **難点 1:** 鍵の安全な共有方法が大問題
  - ☒ 暗号化用の鍵は公開してしまえるので問題なし
- **難点 2:** 相通信するペア毎に異なる鍵が必要
  - ☒ すべての送信者に対して一つの公開鍵／秘密鍵ペアのみで済む

インターネット環境で利用するには必須の技術

暗号化技術概説

RSA

- ***R. Rivest, A. Shamir, L. Adelman*** の三人が発明した暗号化アルゴリズム (1978 年)
- 大きな数の素因数分解の困難性に基づく
- 非対称暗号方式のデファクトスタンダード

暗号化技術概説

楕円曲線暗号

- ***N.Koblitz*** と ***V.S.Miller*** の二人が独立に考案 (1985 年)
- 新しい方式ではなく、***Diffie-Hellman*** などの既存システムを楕円曲線上に実装したものが主
- 他の非対称アルゴリズムと比較して短い鍵長で安全性を確保できるということで注目される

構成

- **暗号化技術概説**
- ***認証技術の発展***
- **実プロトコルでの利用形態**

電子認証技術とは

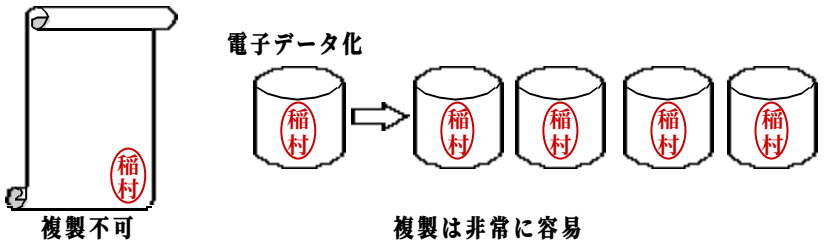
電子認証技術

- 誰が誰であるのかを、確実に判断する必要性
  - » ネットの向こうにいるのは誰？
    - ☒ 電子認証技術
- 基本は実世界での認証と同じ
  - » 物理的特徴の確認
  - » 所有物の確認
    - ☒ あらかじめ登録しておいたデータとの照合処理
- ただし、電子データ固有の事情も

電子認証技術

電子認証技術とは (2)

- 電子データ固有の事情
  - » 電子データは複製が容易なため、単純に実世界の仕組みを持ち込んでも機能しない



電子認証技術

電子認証技術とは (3)

- 認証手段 (ECOM[illegible])
  - » 本人に特[illegible]
    - ☒ 指紋
    - ☒ 声紋
  - » 所持品による：
    - ☒ クレジットカード、運転免許証
  - » 秘密情報による：
    - ☒ パスワード、暗証番号
    - ☒ デジタル署名、公開鍵証明書 (デジタル ID)

• ネットワーク経由の認証には不向き
• 入退室管理など特定用途向け

• ネットワーク経由の認証には不向き
• 入退室管理など特定用途向け

電子認証技術

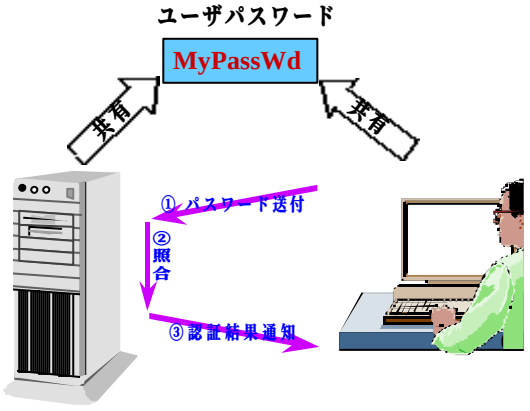
電子認証技術とは (4)

- パスワード認証
  - 単純パスワード
  - 暗号化パスワード
- チャレンジ & レスポンス
- ワンタイムパスワード
- ***Kerberos***
- 電子署名

電子認証技術

各種認証技術

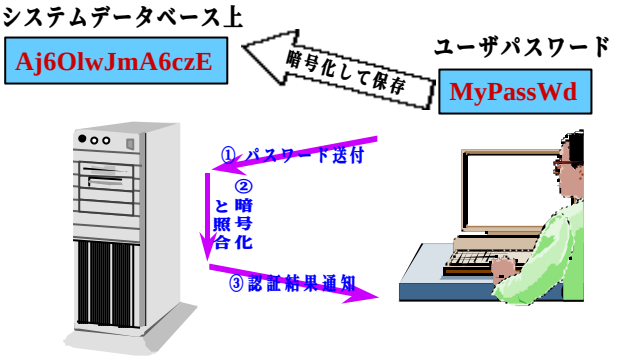
- 単純パスワード方式



各種認証技術
続き

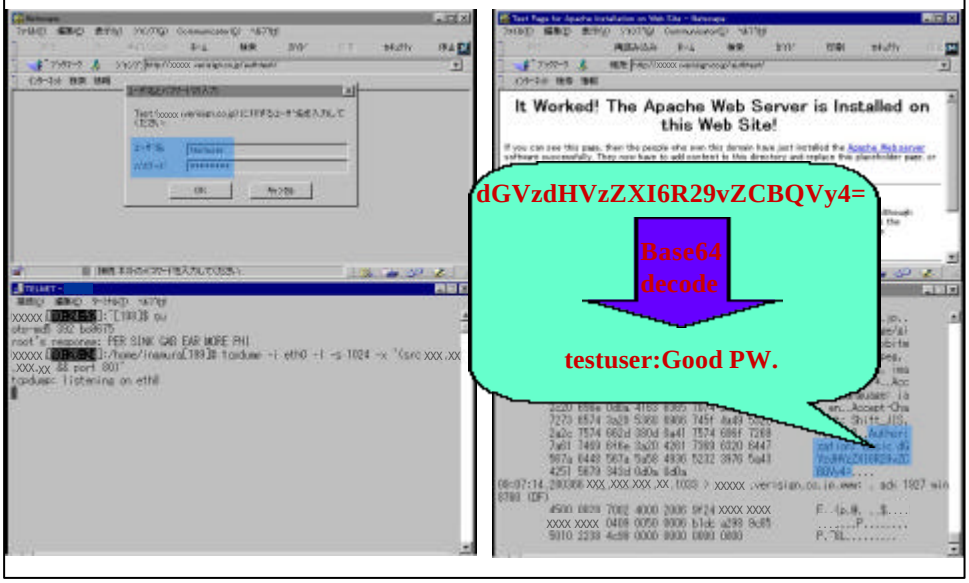
電子認証技術

- 暗号化パスワード方式



パスワード認証の危険性

電子認証技術



電子認証技術

メッセージダイジェスト

- 任意のデータから、そのデータ特有とみなせる短い (百数十ビット程度) 情報 (= メッセージダイジェスト) を抽出する技術
- 暗号学的一方向ハッシュ関数を利用
  - 逆演算不可
  - ***Collision Proof*** 性
- メッセージダイジェストは、元データの“指紋”として扱うことが可能

電子認証技術

メッセージダイジェスト (2)

- **暗号学的一方向ハッシュ関数が持つべき特性**
  - **元データが 1bit 異なっただけで、メッセージダイジェスト中の多くの(半数程度) bit に影響**
  - **あるメッセージダイジェストに対応する元データを見つけ出すのが非常に困難**
  - **同じメッセージダイジェストを得られるような二種類のメッセージを見付けるのが非常に困難**

Birthday Attack
同じ誕生日の人間を見付けるのに何人必要か

電子認証技術

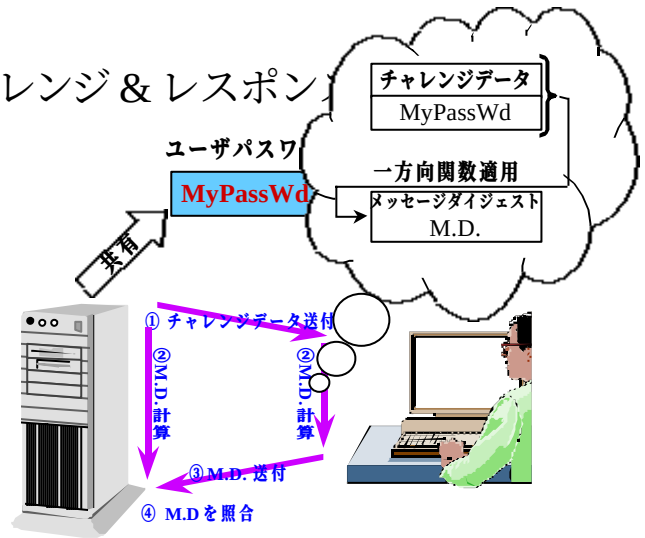
メッセージダイジェスト (3)

- MD2/4/5
  - ***R. Rivest*** らによるアルゴリズム
  - 128bit 長のメッセージダイジェストを抽出
- SHA-1
  - 米国 NIST が NSA とともに開発したアルゴリズム
  - 160bit 長のメッセージダイジェストを抽出

電子認証技術

各種認証技術
続き

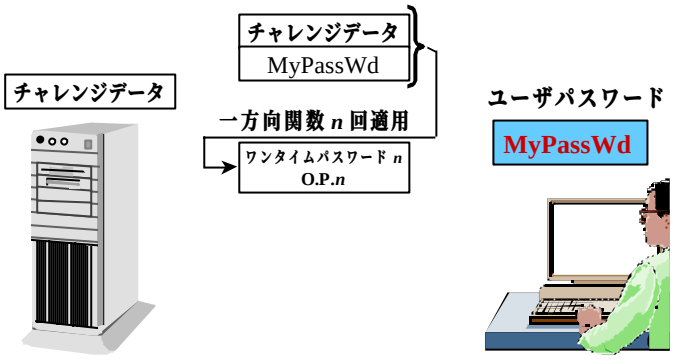
- チャレンジ & レスポンス



電子認証技術

各種認証技術
続き

- ワンタイムパスワード方式 (S/Key 等)
  - 前準備 (1)



電子認証技術

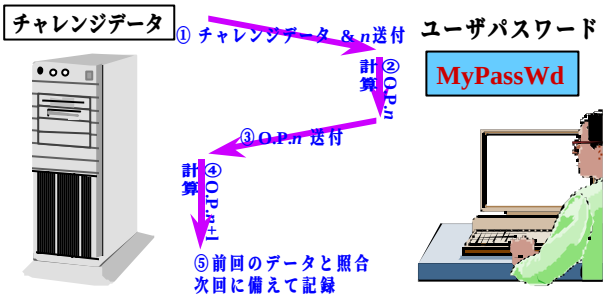
各種認証技術
続き

- ワンタイムパスワード方式 (S/Key 等)
  - 前準備 (2)

電子認証技術

各種認証技術
続き

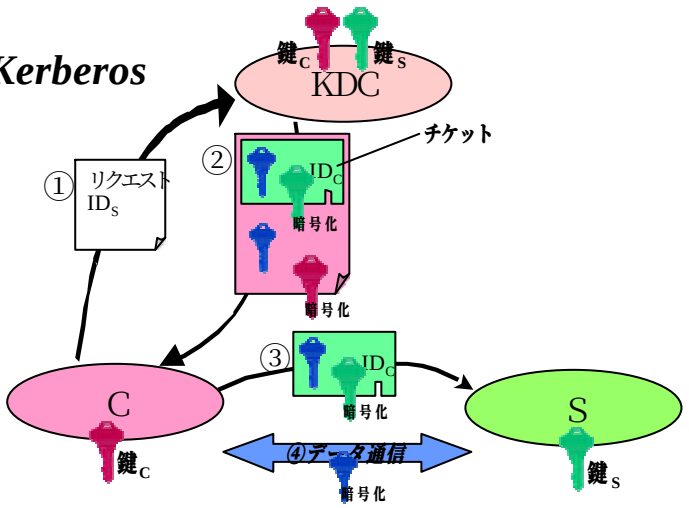
- ワンタイムパスワード方式 (2)



電子認証技術

各種認証技術
続き

- ***Kerberos***



なぜ電子署名方式か？

電子認証技術

パスワード	秘密情報がそのままネットを流れる
チャレンジ&レスポンス	システム中のパスワード漏洩
ワンタイムパスワード	保護は認証時のみ
Kerberos	KDC（中心的な鍵管理センター）の存在

- いずれも、否認防止には役立たない

電子署名

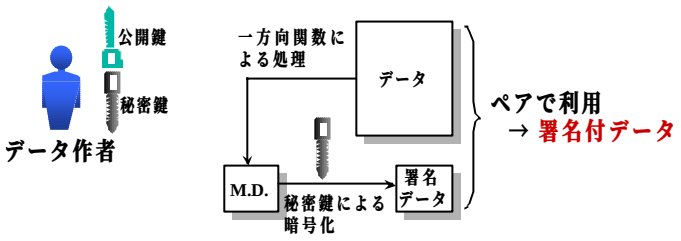
電子認証技術

- ＝デジタル署名
- データ作成者の身元同定を可能にするために、データに付加される“印”
  - 実世界における署名／印鑑押捺などに相当
- 通常は、メッセージダイジェストと非対称暗号技術の併用によって実現
  - 内容が改竄されていないことの保証も可能

電子認証技術

電子署名 (2)

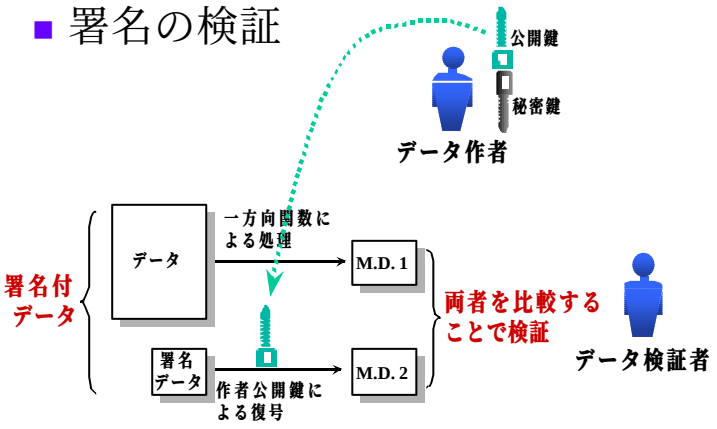
- メッセージダイジェストと非対称暗号技術の併用による実現
  - » メッセージダイジェストの***Collision Proof***性と暗号化で署名データの一意性を保証



電子認証技術

電子署名 (3)

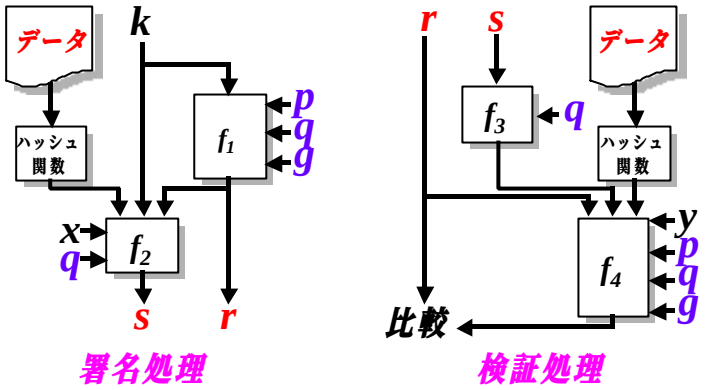
- 署名の検証



電子認証技術

電子署名 (4)

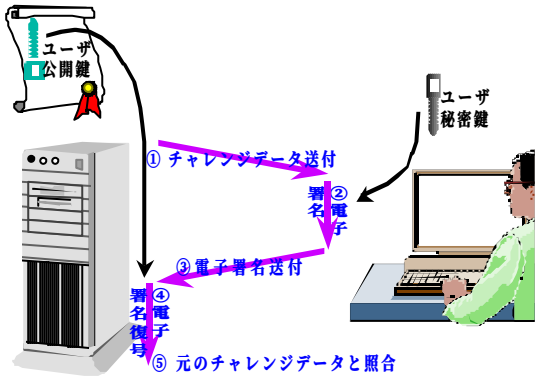
- **DSA** 風アプローチ



電子認証技術

各種認証技術
続き

- 電子署名を用いた認証



構成

- **暗号化技術概説**
- **認証技術の発展**
- ***実プロトコルでの利用形態***

実プロトコルでの利用形態

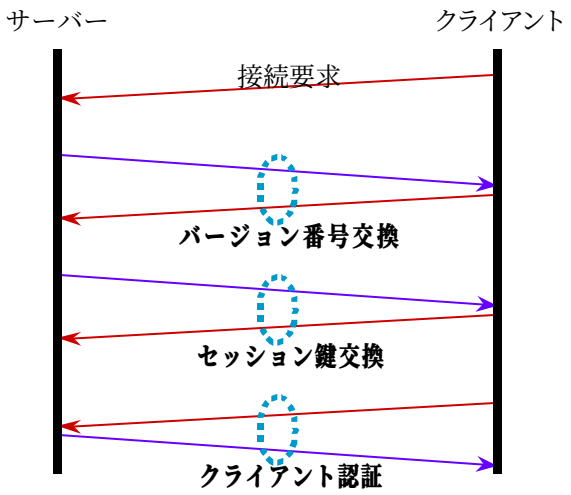
- SSH (*Secure Shell*)
- S/MIME (*Secure/Multipurpose Internet Mail Extensions*)
- SSL (*Secure Sockets Layer*)
- IPSec (*Security Architecture on Internet Protocol*)

SSH (*Secure SHell*)

- *Tatu Ylonen* 氏考案のセキュリティ・プロトコル
- BSD UNIX の rcmd 機能の置き換え
- 特徴
  - サーバ／クライアントに同じユーザ・アカウントが存在することが前提
  - 対称／非対称の両アルゴリズムを併用
  - UNIX 上の実装も、商用利用の場合は有償に (2.0 版から)

SSH

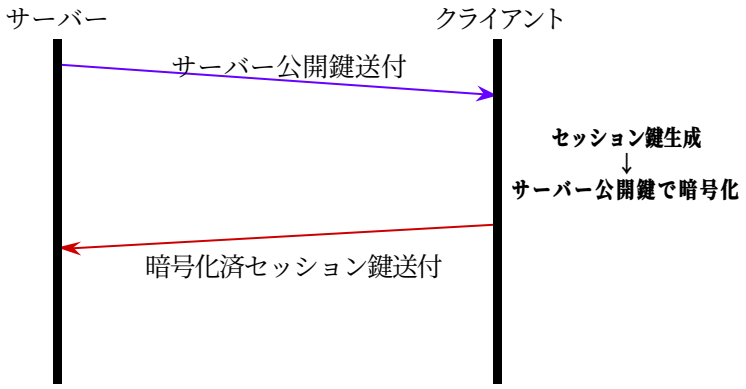
SSH プロトコル概要 (1)



SSH

SSH プロトコル概要 (2)

- セッション鍵交換



SSH

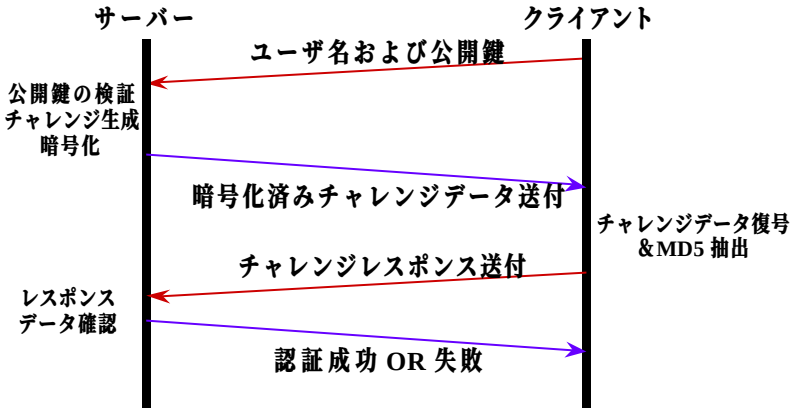
SSH プロトコル概要 (3)

- クライアント認証方式
  - ホスト RSA 認証
    - クライアントホスト RSA 鍵を利用
  - ユーザ RSA 認証
    - ユーザ RSA 鍵を利用
  - パスワード認証
    - 通常ログイン時と同じ認証
    - ただし、パスワード情報はセッション鍵で暗号化

SSH

SSH プロトコル概要 (4)

- ユーザ RSA 認証方式



SSH

処理概要とその他の特徴

- ユーザ・アカウントの存在を仮定できるため、認証の実現は比較的容易
- ***Man-in-the-Middle*** 型の攻撃に対する対処が問題
- 他のプロトコルに対して安全なコネクションを提供可能
  - X, FTP, MAIL (SMTP, POP), TELNET
  - VPN (*Virtual Private Network*) 類似機能を簡便に実現

SSH

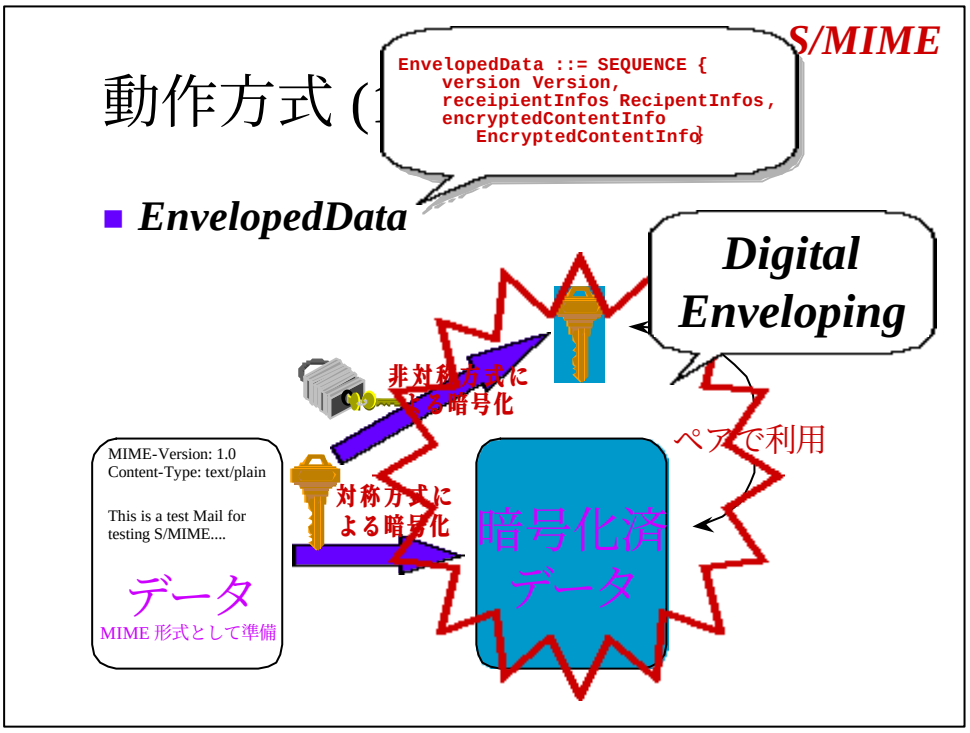
簡易 VPN スクリプト

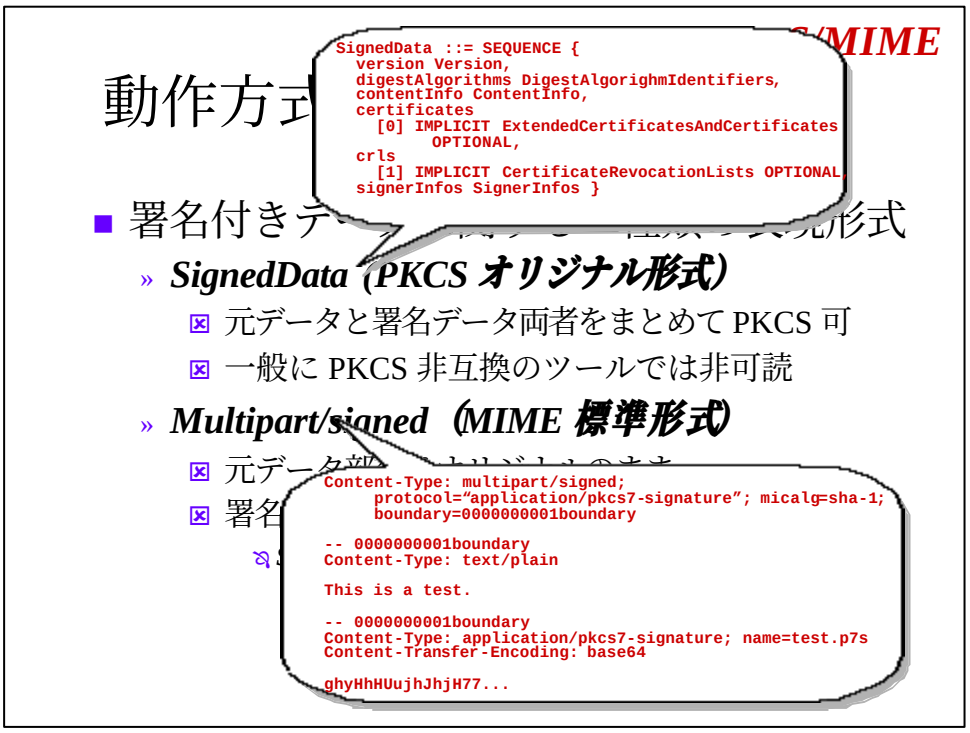
```
#! /bin/sh
while true; do
  ssh -R 8022:localhost:22 ¥
  -R 8025:localhost:25 -R 8110:localhost:110 ¥
  -L 8025:remote:25 -L 8110:remote:110 ¥
  remote
  sleep 10
done
```

S/MIME

S/MIME (*Secure/Multipurpose Internet Mail Extensions*)

- *RSA* 社提案のセキュア・メール規格
- 電子メールへの暗号化／電子署名処理
- 特徴
  - ユーザと公開鍵との結び付きを証明書で保証
  - 対称／非対称の両アルゴリズムを併用
  - Netscape 社、Microsoft 社のメール・ソフトにネイティブに実装


S/MIME
動作方式 (
EnvelopedData ::= SEQUENCE {
version Version,
receipientInfos RecipentInfos,
encryptedContentInfo
EncryptedContentInfo }
EnvelopedData
Digital Enveloping
非対称方式による暗号化
ペアで利用
MIME-Version: 1.0
Content-Type: text/plain
This is a test Mail for testing S/MIME....
データ
MIME 形式として準備
対称方式による暗号化
暗号化済データ


S/MIME
動作方式
SignedData ::= SEQUENCE {
version Version,
digestAlgorithms DigestAlgorighmIdentifiers,
contentInfo ContentInfo,
certificates
[0] IMPLICIT ExtendedCertificatesAndCertificates
OPTIONAL,
crls
[1] IMPLICIT CertificateRevocationLists OPTIONAL,
signerInfos SignerInfos }
署名付きデータ
SignedData（PKCS オリジナル形式）
元データと署名データ両者をまとめて PKCS 可
一般に PKCS 非互換のツールでは非可読
Multipart/signed（MIME 標準形式）
元データ部
署名
Content-Type: multipart/signed;
protocol="application/pkcs7-signature"; micalg=sha-1;
boundary=0000000001boundary
-- 0000000001boundary
Content-Type: text/plain
This is a test.
-- 0000000001boundary
Content-Type: application/pkcs7-signature; name=test.p7s
Content-Transfer-Encoding: base64
ghyHhHUujhJhjH77...

S/MIME

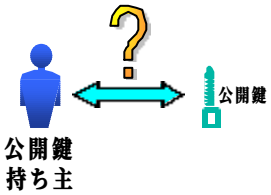
考慮ポイント

- 公開鍵の入手をどうするか?
  - 暗号化メールの場合は受信者の
  - 署名メールの場合は送信者の
- デジタル証明書を用いることで、スケーラブルな相互運用性を確保

S/MIME

デジタル証明書

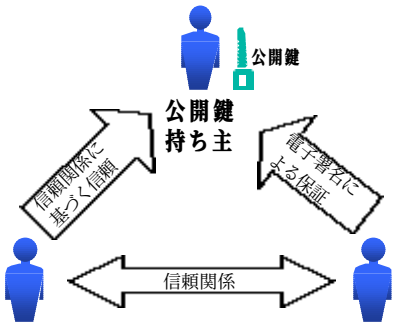
- 公開暗号技術の補完的役割
- ある公開鍵の持ち主が本当に申告通りの人間であるかどうかを保証するための機構



S/MIME

デジタル証明書

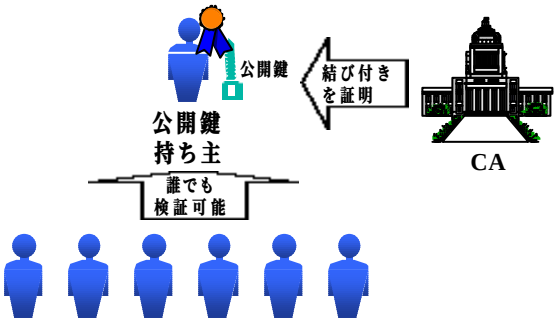
- 1. 草の根的解決 (PGP)
  - 『友達の友達は友達』方式



S/MIME

デジタル証明書

- 2. 信頼された第三者機関による保証
  - 認証機関 (= Certification Authority, CA) がユーザと公開鍵との結びつきを証明



S/MIME

デジタル証明書

- 証明書の内容

X.509 バージョン番号	・・・	X.509 のバージョン
認証証明書のシリアル番号	・・・	認証証明書ごとのユニークな番号
署名方法 (アルゴリズム名)	・・・	この認証証明書の署名方法
CA の名前	・・・	この認証証明書を発行した機関名
有効期間	・・・	この認証証明書の有効期間
認証証明書の持ち主の名前	・・・	登録された公開鍵の申請者の名前
認証証明書の持ち主の公開鍵情報	・・・	登録された申請者の公開鍵
拡張 (X.509 Ver3 のオプション)	・・・	X.509 の拡張フィールド
CA による署名	・・・	上記全項目に対して一括して施した電子署名

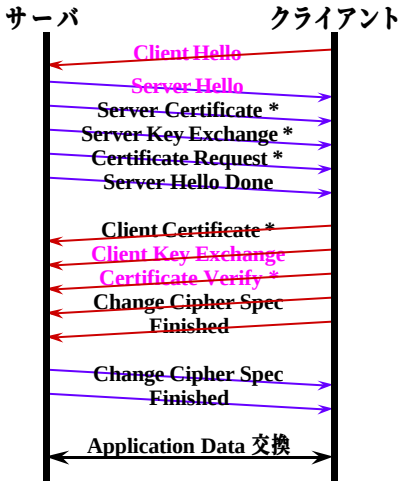
SSL

SSL *(Secure Sockets Layer)*

- *Netscape* 社提案のセキュリティ・プロトコル
- 特徴
  - » サーバ／クライアント・モデルでの利用
    - ☒ 一般に、認証用データを前もって準備しておくことは不可能
  - » 相互運用性確保のためデジタル証明書を利用
  - » 対称／非対称の両アルゴリズムを併用
  - » 多くのウェブ・サーバ／クライアントに実装

SSL

SSL プロトコル概要



SSL

SSL プロトコル概要 (1)

- C/S Hello
  - 暗号化／圧縮アルゴリズムの決定
  - 時間データを含み、接続のフレッシュさを保証

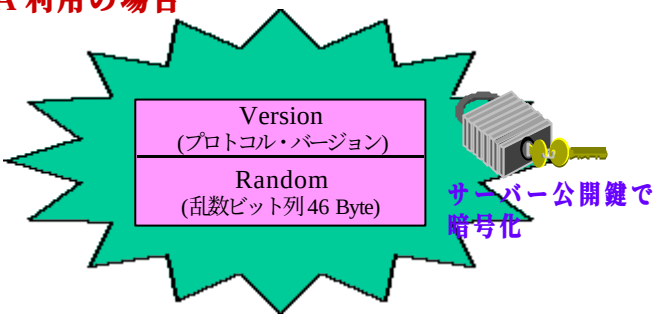
= Random
マスタ・シークレット生成に利用

SSL

SSL プロトコル概要 (2)

- Client Key Exchange
  - » マスタ・シークレットを生成する元となるデータ (プリ・マスタ・シークレット)をサーバーに送付

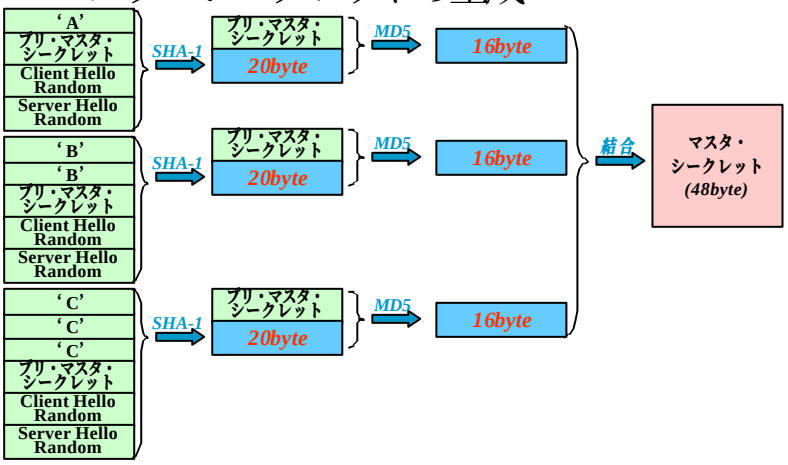
RSA 利用の場合



SSL

SSL プロトコル概要 (3)

- マスタ・シークレットの生成

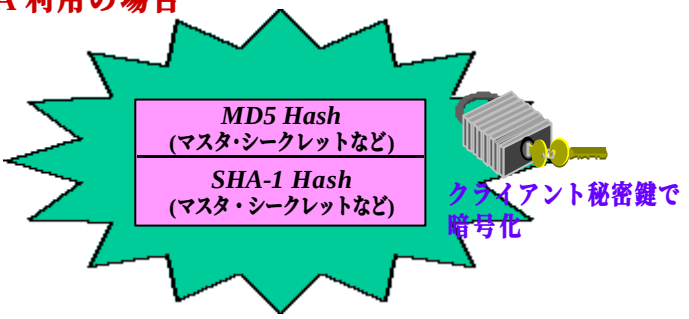


SSL

SSL プロトコル概要 (4)

- Certificate Verify
  - » サーバーがクライアント証明書の認証を行なうのを補助する目的でクライアントが送付

RSA 利用の場合



SSL

SSL プロトコル概要 (5)

- キーブロックの生成
  - » 暗号化鍵、初期化ベクタ、MAC 計算用秘密データなどとして利用するデータの生成処理
  - » 各種データに十分な量に達するまで、以下の計算を行う

$$Key\_Block = MD5(MasterSecret + SHA(MasterSecret + ServerHello.Random + ClientHello.Random + 'A')) + MD5(MasterSecret + SHA(MasterSecret + ServerHello.Random + ClientHello.Random + 'BB')) + MD5(MasterSecret + SHA(MasterSecret + ServerHello.Random + ClientHello.Random + 'CCC')) + \cdots$$

+: 結合演算

SSL

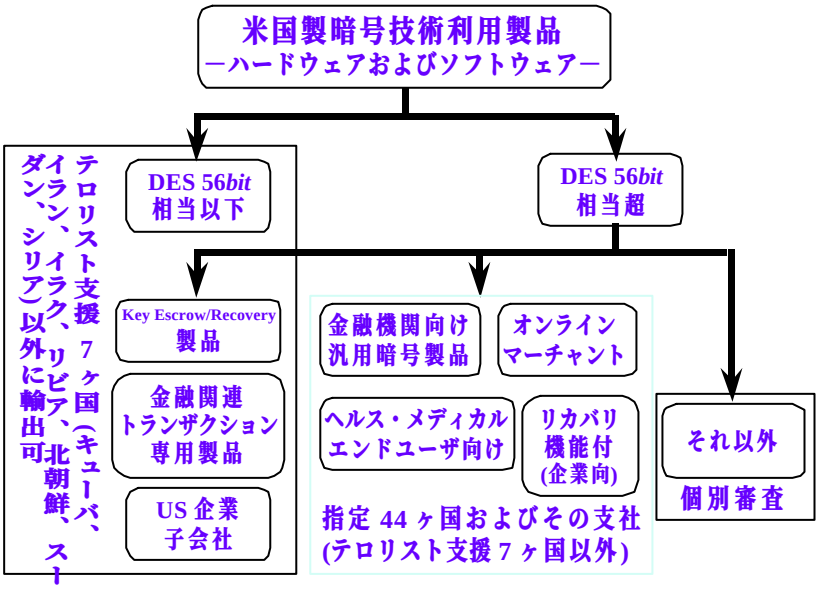
処理概要とその他の特徴

- ユーザ・アカウントの存在が仮定できないため、認証のためには証明書が必須
  - ***Man-in-the-Middle*** 型の攻撃への防御にも
  - オプションで証明書によるクライアント認証も可能
- 暗号強度を変えることで、米国輸出規制をクリア
  - 米国国内版・国際版の存在
  - 米国外で高強度暗号を利用する手段も提供
    - グローバル・サーバ ID 等

SSL

米国輸出管理規制

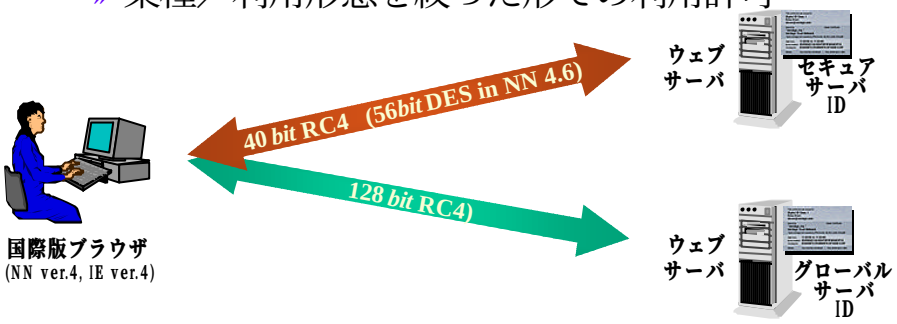
1999 年 11 月 1 日現在



SSL

グローバル・サーバ ID

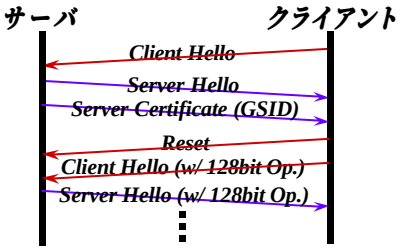
- 米連邦政府輸出管理規制を実現するための切札的存在
  - » 業種／利用形態を絞った形での利用許可



SSL

グローバル・サーバ ID

- グローバル・サーバ ID 利用時の SSL ネゴシエーション・シーケンス



SSL

グローバル・サーバ ID

ちなみに、現時点で、公の場での 56*bit* クラッキング・レコードは 22 時間 15 分

- **56*bit* vs 128*bit* コスト比較**
  - 1995 年時点での推測値

$100K	35 時間	10^{19} 年
$1M	3.5 時間	10^{18} 年
$10M	21 分	10^{17} 年
$100M	2 分	10^{16} 年
$1G	13 秒	10^{15} 年
$10G	1 秒	10^{14} 年
$100G	0.1 秒	10^{13} 年

Bruce Schneier, Applied Cryptography 2nd Ed. John Wiley & Sons, Inc. より

IPSec

IPSec *(Internet Protocol)*

- *IETF* を中心として標準化作業中のセキュリティ・プロトコル
- 特徴
  - ***Internet Protocol*** 自体へセキュリティ機能付与
    - コネクションレスの IP では、SSH/SSL のように通信毎に認証を行なうことは非現実的
  - 二者間で前もって暗号化等に用いるパラメータの交渉を済ませておく
    - 認証／鍵交換などを別プロトコル化

IPSec

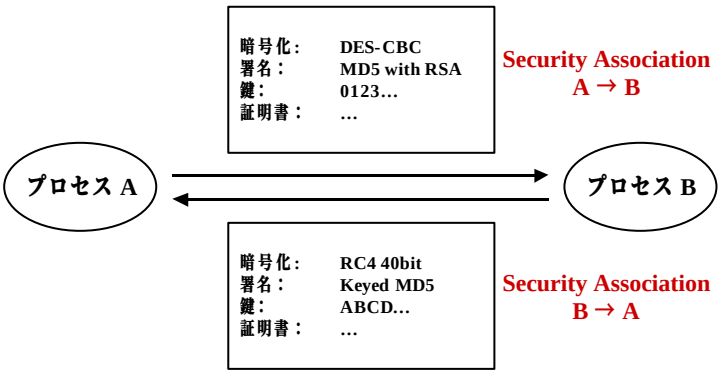
鍵管理方式との分離 (1)

- 鍵管理方式は任意に利用可能
  - » 手動配布, ***Diffie&Hellman***, ***Kerberos***, etc.
- **Security Association** (SA) と **Security Parameters Index** (SPI) によって指定

IPSec

鍵管理方式との分離 (2)

- **Security Association (SA)**



IPSec

鍵管理方式との分離 (3)

- **Security Parameters Index (SPI)**

SPI for Host X

送信者	暗号方式	署名方式	鍵	…
Host A	DES-CBC	MD5withRSA	0123…	
Host B	IDEA-CBC	Keyed MD5	ABCD…	
Host C	RC5-CFB	SHAwithDSS	FECD…	
…				
Host X	RC4	MD5withRSA	5678…	
…				

Security Association Table

IPSec

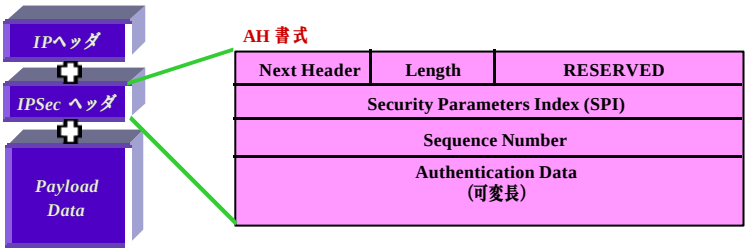
自動鍵管理方式の例

- ***IKE (Internet Key Exchange)***
  - 通信相手との間で、利用する暗号アルゴリズムや鍵などのパラメータ (セキュリティ･アソシエーション) を定めるためのプロトコル
  - ***Diffie & Hellman*** 方式(公開鍵アルゴリズム)ベースで、安全に鍵を交換

IPSec

AH (Authentication Header)

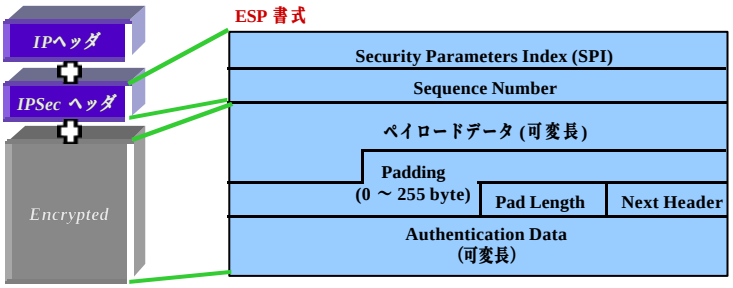
- IP パケットに対して完全性保証用データを付加するための新しい IP ヘッダを定義
- HMAC-MD5 と HMAC-SHA-1



IPSec

ESP (Encapsulating Security Payload)

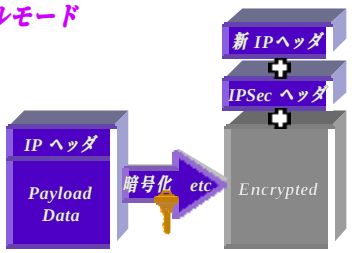
- IP パケットを暗号化 (オプションで完全性保証用データを付加) した上で送付するための新しい IP ヘッダを定義
- DES-CBC



IPSec

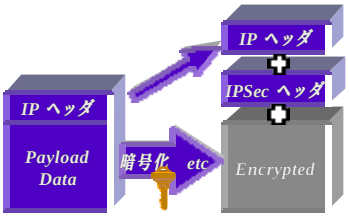
動作モード

・トンネルモード



- 元の IP ヘッダを含めた全体に対して処理を実施
  - 本来の宛先・発信アドレスなどに関する情報まで保護可能
  - 新 IP ヘッダが必要な分、データ量が増加

・トランスポートモード



- 元の IP ヘッダを除く部分に対して処理を実施
  - データ量の増加は IPSec ヘッダの分のみ
  - 本来の宛先・発信元アドレスに関する情報は保護不可

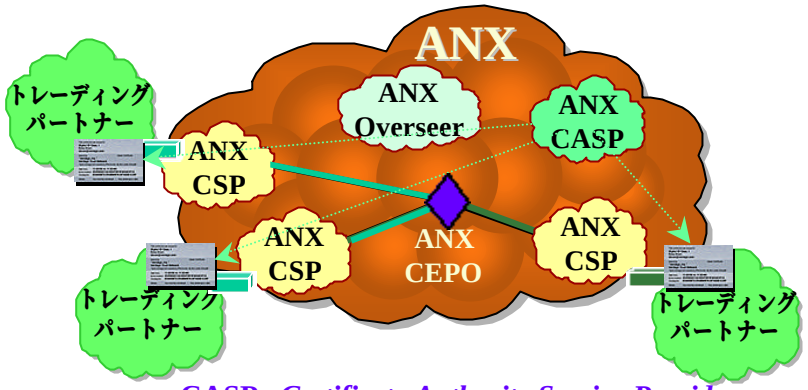
IPSec

処理概要とその他の特徴

- IP のコネクションレスという特性から、鍵交換と実際の運用を完全に分離
  - » **利用環境に応じて柔軟な構成が可能**
    - 小規模な LAN 環境では手作業でのパラメータ設定が可能
    - 大規模なエクストラネットなどでは証明書を用いた相互運用性の確保
      - ANX (Automobile Network eXchange)

IPSec

ANX 概要



CASP *Certificate Authority Service Provider*
CEPO *Certified Exchange Point Operator*
CSP *Certified IP Service Provider*
IPSec Gateway