



PKI: 基礎と応用


稲村雄
International Network Security, Inc.
mailto:jane@insi.co.jp



本日の講演内容

- ◆ *PKI (Public Key Infrastructure) という概念*
- ◆ PKI要素技術
- ◆ PKIとは何か
- ◆ *PKI Protocols/Applications*
- ◆ 実社会への *implications*


2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 2



PKI (*Public Key Infrastructure*) という概念

- ◆ 話者／文脈により、意味のぶれ幅が非常に大きい単語
 - 認証機関(CA)の存在およびその構造自体
 - 公開鍵暗号技術(*Public Key Cryptography*)利用環境(HW&SW)の総体
 - *Infrastructure*=社会基盤としての公開鍵暗号技術利用可能性
 - 公開鍵暗号技術そのもの
 - ECで儲けるための新手法の魔法のことば
 - *FW, Vaccine Soft, IDS, etc.*に続く


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 3



PKI (*Public Key Infrastructure*) という概念 *contd.*

- ◆ 話者／文脈により、意味のぶれ幅が非常に大きい単語
 - 認証機関(CA)の存在およびその構造自体
 - 公開鍵暗号技術(*Public Key Cryptography*)利用環境(HW&SW)の総体
 - *Infrastructure*=社会基盤としての公開鍵暗号技術利用可能性
 - ~~公開鍵暗号技術そのもの~~
 - ~~ECで儲けるための新手法の魔法のことば~~
 - ~~• *FW, Vaccine Soft, IDS, etc.*に続く~~


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 4



社会基盤としての公開鍵暗号技術利用可能性

- ◆ 交通機関／通信／電力など、他の社会基盤と同じレベルの公開鍵暗号技術を(主として)インターネット上に提供
 - 安全な要素技術の設計および実装
 - 対称／非対称暗号
 - ハッシュ関数
 - 安全なProtocol/Applicationの設計および実装
 - 運用
 - ユーザ(=セキュリティに関する最も弱いリンク)
 - 利用環境整備
 - 教育


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You



PKIに関するよくある誤解

- ◆ 公開鍵暗号は共通鍵暗号より優れている
- ◆ 公開鍵暗号ならば、鍵の配布は問題ない
- ◆ デジタル証明書は秘密にしないと危険
- ◆ デジタル証明書を送付すると認証される
- ◆ PKIさえ導入すれば、セキュリティは万全
- ◆ 暗号って難しい


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 6



PKIに関するよくある誤解 *contd.*

- ◆ 暗号は難しい?
 - NO !
 - 理論⇒実践のギャップが小さい
 - 量的には共通鍵系で二つ、公開鍵系で二つ、計4項目も覚えればそれで十分
 - 他のセキュリティ技術に比べれば遥かに取っ付き易いはず


2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 7



PKI関連の今年のトピック

- ◆ 米ベリサイン社による証明書誤発行
 - Microsoft社の名を騙った証明書
 - 証明書有効性検証の難しさを露呈
- ◆ PKIベンダのウェブサイト書き換え事件
 - ベリサイン&ボルチモア
 - PKIが万能なわけではない
- ◆ 電子署名法施行&特定認証業務運用開始

2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 8



PKI(?)関連の今年のトピック *contd.*

- ◆ Timing Attack on SSH
 - サーバ⇔クライアント間を流れるデータのタイミングを測ることでキー入力を推測
 - Touch Typingできない方が安全という逆説性
- ◆ WEP脆弱性
 - いくら “*Wired Equivalent*”程度と言っても...
 - ある意味今年最大の事件かも


2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 9



本日の講演内容

- ◆ PKI (*Public Key Infrastructure*)という概念
- ◆ *PKI*要素技術
- ◆ PKIとは何か
- ◆ *PKI Protocols/Applications*
- ◆ 実社会への *implications*


2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 10



PKI要素技術

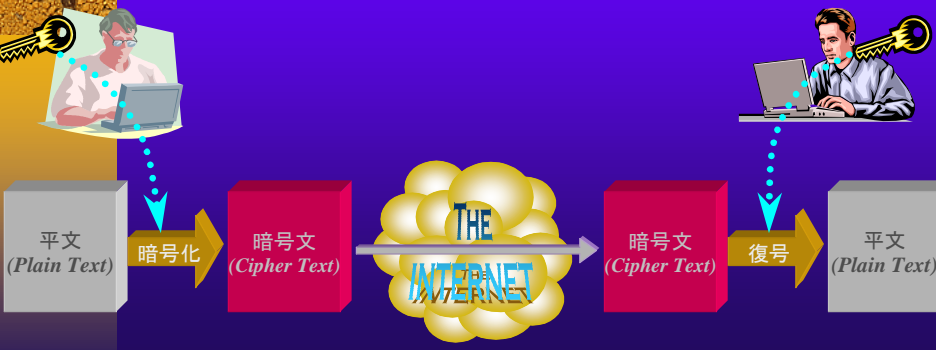
- ◆ 対称暗号
- ◆ 公開鍵配布系
- ◆ 非対称暗号
- ◆ 一方向ハッシュ関数

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 11



対称暗号





- ◆ = 秘密鍵暗号 / 共通鍵暗号 / 慣用暗号
- ◆ 暗号化 / 復号に同じ鍵が用いられる



2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 12

対称暗号の問題点


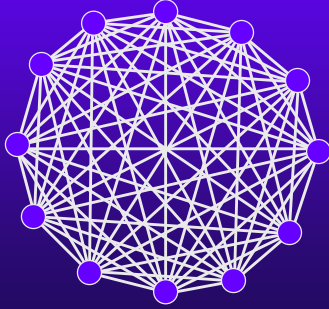
- ◆ 基本的な方式として、鍵と平文／暗号文との間で複雑な演算を行なうことで暗号化／復号を実現
- ◆ ブロック暗号とストリーム暗号の二種類に大別
- ◆ **問題点 1**: 如何にして鍵を安全に共有できるか?

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 13

対称暗号の問題点 *contd.*

- ◆ **問題点 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

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

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 14

公開鍵配布系

- ◆ 送受信者双方が個々に公開している情報から、その二者のみしか知り得ない秘密の鍵 (for 対称暗号) を生成

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 15


公開鍵配布系 *contd.*

- ◆ *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：

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

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 16




公開鍵配布系 *contd.*

◆ Diffie-Hellman 方式 *contd.*

大きな素数 q と $GF(q)$ 上の原始元 α を決定
 各ユーザは $GF(q)$ から任意の整数 X_p を選び、
 $K_p \equiv \alpha^{X_p} \pmod{q}$ を公開情報とする
 AがBにメッセージを送る:
 $K_{AB} \equiv (K_B)^{X_A} \pmod{q}$ を暗号化鍵とする
 BがAからメッセージを受ける:
 $K_{BA} \equiv (K_A)^{X_B} \pmod{q}$ を復号鍵とする
 K_{AB} と K_{BA} は等しく、また、A, B 以外がこの値
 を計算するには K_A もしくは K_B の対数計算が必要

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 17



公開鍵配布系 *contd.*

◆ Diffie-Hellman 方式 *contd.*

$q=11, GF(q) = \{1,2,3,4,5,6,7,8,9,10\}, \alpha=2$
 $X_A=3, X_B=4, K_A \equiv 2^{X_A} \pmod{11} = 8, K_B \equiv 2^{X_B} \pmod{11} = 5$
 AがBにメッセージを送る:
 $K_{AB} \equiv (K_B)^{X_A} \pmod{11} = 5^3 \pmod{11} = 4$
 BがAからメッセージを受ける:
 $K_{BA} \equiv (K_A)^{X_B} \pmod{11} = 8^4 \pmod{11} = 4$

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 18

非対称暗号

- ◆ = 公開鍵暗号
- ◆ 暗号化／復号に異なった鍵が用いられる



⇒ 公開鍵



⇒ 秘密鍵



2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 19

非対称暗号 contd.

- ◆ RSA 方式

大きな素数 P, Q を選択

$N \equiv P \cdot Q, \quad \Phi(N) = (P-1)(Q-1)$


$\Phi(N)$ に関して互いに素な数 E を選ぶ

$E \cdot D = 1 \pmod{\Phi(N)}$ を満たす数 D を選ぶ

暗号化処理: $C = M^E \pmod{N}$

復号処理: $M = C^D \pmod{N}$

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 20




非対称暗号 *contd.*

◆ *RSA* 方式 *contd.*

E を選ぶ (通常 $2^{16} + 1$)
 大きな素数 P, Q を選択
 条件: “ $P-1, Q-1$ がそれぞれ E と互いに素 ” を満たす
 $N \equiv P \cdot Q, \quad \Phi(N) = (P-1)(Q-1)$
 $E \cdot D \equiv 1 \pmod{\Phi(N)}$ を満たす数 D を選ぶ
 暗号化処理: $C = M^E \pmod{N}$
 復号処理: $M = C^D \pmod{N}$

2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 21



非対称暗号 *contd.*

◆ *RSA* 方式 *contd.*

$E = 3, P = 5, Q = 11$
 $N \equiv P \cdot Q = 55, \quad \Phi(N) = (P-1)(Q-1) = 40$
 $E \cdot D \equiv 1 \pmod{\Phi(N)}$ を満たす数 $D = 27$
 $M = 5$
 暗号化処理: $C = 5^3 \pmod{55} = 15$
 復号処理: $M = 15^{27} \pmod{55} = 5$

2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 22

非対称暗号 contd.

- ◆ デジタル署名
- ◆ 署名者の身許証明およびデータ完全性保証が可能
 - ⇒ 公開鍵
 - ⇒ 秘密鍵


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 23

非対称暗号 contd.

- ◆ 対称暗号との比較
 - 問題点 1: 如何にして鍵を安全に共有できるか
 - 暗号化用の鍵は公開してしまえるので問題なし
 - 問題点 2: 相通信するペア毎に異なる鍵が必要
 - 一人の受信者は、すべての送信者に対して一つの公開鍵/秘密鍵ペアのみを準備すればよい

インターネット環境で利用するには
必須の技術

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 24



非対称暗号 contd.

- ◆ それでは、非対称暗号は万能なのか?
- ◆ 答えは **NO!**
 - 問題点 1: 処理速度
 - 対称暗号の方が圧倒的に速い
 - 問題点 2: 公開された公開鍵の本当の持主は?
 - PKIなどの仕組みが別途必要(詳細は後述)

対称／非対称両システムの 補完的利用法の確立

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 25



デジタル・エンベロープ

- ◆ 対称アルゴリズムの速度と非対称アルゴリズムの柔軟性
- ニデジタル・エンベロープ(封筒)方式
- 一対称／非対称両システム補完的利用法の一例

デジタル・エンベロープ
暗号文 (Cipher Text)

暗号文 (Cipher Text)

平文 (Plain Text)

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 27

⇒ 公開鍵

⇒ 秘密鍵

~~平文 (Plain Text) → 署名 (Signature) → 署名文 (Signed Text) → 変換 (Transformation) → 署名文 (Signed Text) → 署名 (Signature) → 平文 (Plain Text)~~

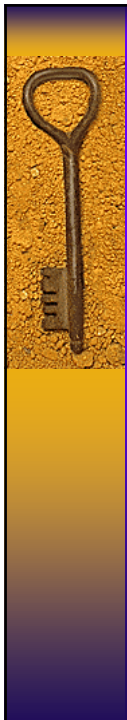
⇒ 公開鍵

⇒ 秘密鍵

平文 (Plain Text) → ハッシュ関数適用 (Hash Function Application) → 署名データ (Signature Data) → 署名 (Signature) → 署名文 (Signed Text) → 変換 (Transformation) → 署名文 (Signed Text) → ハッシュ関数適用 (Hash Function Application) → メッセージダイジェスト (Message Digest) ~160bit

OK? NG? → 検証関数 (Verification Function)

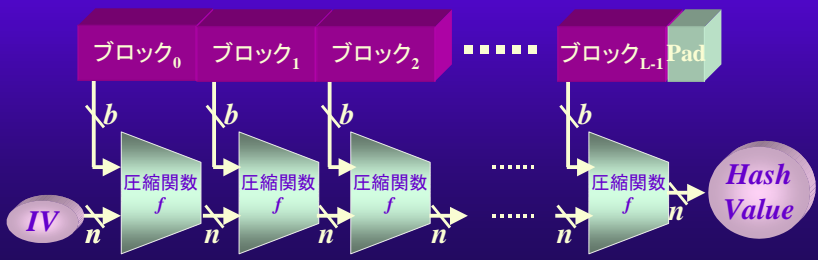
2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You




一方向ハッシュ関数

- ◆ =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

$b=512bit$
 $n=128\sim 160bit$
 あたりが一般的




2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 29



一方向ハッシュ関数 contd.

- ◆ 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
 - 通常攻撃に 2^{128} 、誕生日攻撃に 2^{64}
- ◆ 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
 - 通常攻撃に 2^{160} 、誕生日攻撃に 2^{80}
- ◆ より長いメッセージダイジェスト(256bit~512bit)も近年提案されつつある



2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 30

デジタル署名方式

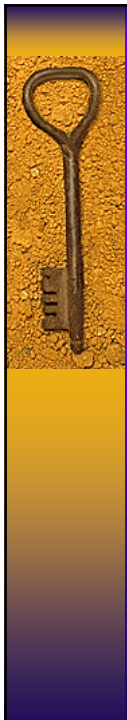
◆ RSA型

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 31

デジタル署名方式 contd.

◆ DSA型

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 32



デジタル署名方式 contd.

◆ DSA

p : 素数, q : 素数 $\wedge q | (p-1)$, $g = \alpha^{\frac{p-1}{q}} \bmod p > 1$
 x : 秘密鍵, $y = g^x \bmod p$: 公開鍵
 署名処理: k : 任意の数
 $r = g^k \bmod p \bmod q$
 $s = k^{-1}(r \cdot x + \text{Hash}(\text{Msg})) \bmod q$
 検証処理: $w = s^{-1} \bmod q$
 $u1 = w \cdot r \bmod q, u2 = w \cdot \text{Hash}(\text{Msg}) \bmod q$
 $y^{u1} \cdot g^{u2} \bmod p \bmod q = r?$

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 33



本日の講演内容

- ◆ PKI (*Public Key Infrastructure*)という概念
- ◆ PKI要素技術
- ◆ PKIとは何か
- ◆ PKI Protocols/Applications
- ◆ 実社会への *implications*

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 34

非対称暗号 *reprise*

- ◆ それでは、非対称暗号は万能なのか?
- ◆ 答えは **NO!**
 - 問題点 1: 処理速度
 - 対称暗号の方が圧倒的に速い
 - 問題点 2: 公開された公開鍵の本当の持主は?
 - PKIなどの仕組みが別途必要(詳細は後述)

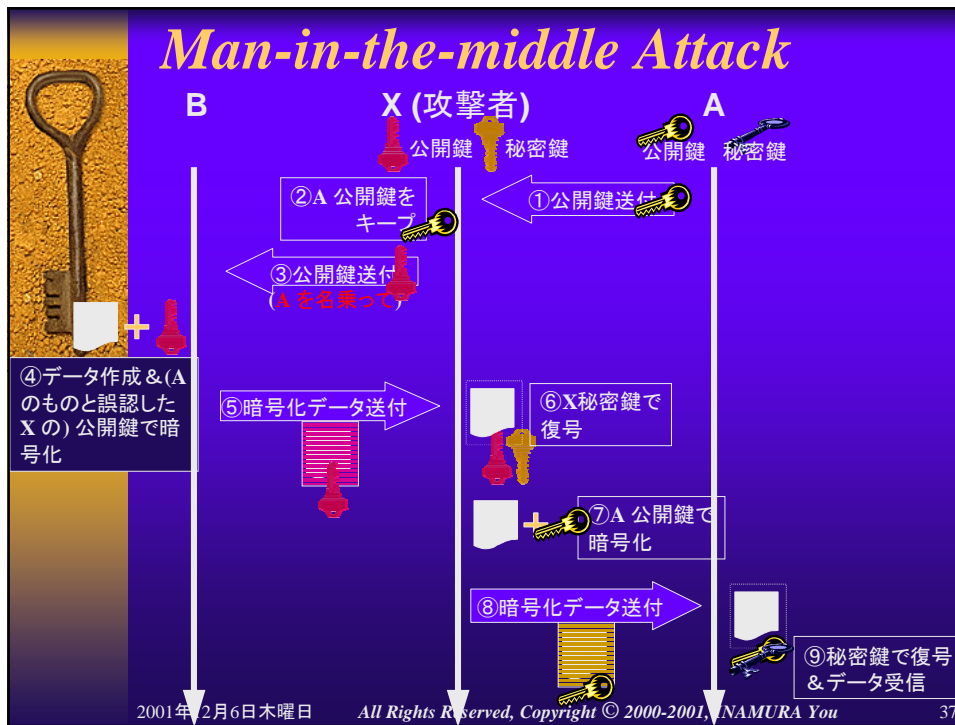
2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 35

犯しやすい過ち

The diagram illustrates the asymmetric encryption process between two parties, B and A:

- Party A:** Possesses a **公開鍵 (Public Key)** and a **秘密鍵 (Private Key)**.
- Step 1:** A sends the **公開鍵 (Public Key)** to B.
- Party B:** Takes a piece of data and a key, and performs **② データ作成 & 公開鍵で暗号化 (Data creation & encryption with public key)**.
- Step 3:** B sends the **暗号化データ (Encrypted data)** to A.
- Party A:** Uses the **秘密鍵 (Private Key)** to perform **④ 秘密鍵で復号 & データ受信 (Decryption with private key & data reception)**.

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 36



非対称暗号のアキレス腱

- ◆ ある公開鍵の持ち主が本当に申告通りの人物であるかどうか？
 - インターネットEC等の成否は、この結び付きを誤りなく判断できるかどうかにかかっている
- ◆ 虚偽の申告が世間で受け入れられた場合、以降その人物になりすますことが可能に

公開鍵 ? [Person at Computer]

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 38




草の根的解決 (PGP型)

◆ 『友達の友達は友達』方式



2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 39



草の根的解決 (PGP型) *contd.*

◆ “草の根的解決”の特徴

- とりあえず他の仕組が要らない
 - 初期段階での普及は容易
- ユーザ層が拡大すると、信頼性に不安が
 - 友達の友達の友達の...が信頼する鍵は信頼できる？
 - 個人への信頼と、その個人による保証への信頼とを区別する

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 40

草の根的解決 (PGP型) contd.

◆ PGP の信頼モデル

```

    graph TD
      Self(本人) --> A(A)
      Self --> B(B)
      Self --> C(C)
      Self --> D(D)
      A --> E(E)
      B --> F(F)
      B --> G(G)
      C --> G(G)
      D --> H(H)
      X(X) --> Y(Y)
      style Self fill:#fff,stroke:#fff
      style A fill:#fff,stroke:#fff
      style B fill:#fff,stroke:#fff
      style C fill:#fff,stroke:#fff
      style D fill:#fff,stroke:#fff
      style E fill:#0000FF,stroke:#0000FF
      style F fill:#0000FF,stroke:#0000FF
      style G fill:#0000FF,stroke:#0000FF
      style H fill:#0000FF,stroke:#0000FF
      style X fill:#FF0000,stroke:#FF0000
      style Y fill:#FF0000,stroke:#FF0000
    
```

→	X が Y に署名		署名者として信頼
	信頼できる鍵		署名者として部分的に信頼
	信頼できない鍵		署名者として信頼しない

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 41

デジタル証明書/CA/PKI

◆ CA (Certification Authority, 認証機関) が、
ユーザの身許を確かめた上で、“お墨付き”
(= デジタル証明書) を発行

- 誰でもユーザ ⇔ 公開鍵の関係が検証可能に

結び付きを証明

公開鍵

誰でも検証可能

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 42

デジタル証明書/CA/PKI *contd.*

◆ X.509v3 証明書

記載されるのは基本的に公開情報のみ
∴ 誰にでも公開可能

- ITU-T X.509 勧告で定義されたフォーマット
- 同勧告第3版が現在事実上の証明書標準
- Internet上の標準“PKIX”もX.509v3ベース

X.509 バージョン番号	… … … X.509 のバージョン (現行はv3)
シリアル番号	… … … CAごとに一意に証明書を指定
署名方式 (アルゴリズム名)	… … … 証明書に対する署名方式
発行CA名	… … … 証明書を発行したCAの名前
有効期間	… … … 証明書の有効期間 (開始/終了日時)
証明書所持者名	… … … 証明書および公開鍵所有者の名前
証明書所持者の公開鍵情報	… … … 証明書所持者の公開鍵データそのもの
拡張 (X.509 Ver3 のオプション)	… … … 拡張フィールド(パス検証処理補助情報などに利用)
CAによる署名	… … … 上記全項目に対してCAが施した電子署名

2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 43

デジタル証明書/CA/PKI *contd.*

◆ 証明書の検証



CA

検証結果がOK
であれば
がユーザの正しい公開鍵である
ことが明らかに

ハッシュ関数
適用

検証
関数

X.509 バージョン番号
シリアル番号
署名方式 (アルゴリズム名)
発行CA名
有効期間
証明書所持者名
証明書所持者の公開鍵情報
拡張 (X.509 Ver3 のオプション)
CAによる署名

2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 44

デジタル証明書/CA/PKI contd.

◆ CA 自体の公開鍵の保証は？

検証結果がOK
であれば
がユーザの正しい
公開鍵である
ことが明らかに

ハッシュ関数
適用

検証
関数

X.509 バージョン番号

シリアル番号

署名方式 (アルゴリズム名)

発行CA名

有効期間

証明書所有者名

証明書所有者の
公開鍵情報

拡張 (X.509 Ver3 のオプション)

CAによる署名

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 45

PKI の定義

CA の存在および構造自体

CA1 (TRUSTED) → CA2 (UNTRUSTED)

↓ TRUSTED ↓ UNTRUSTED ↓

検証可能 ← 検証可能 →

階層型

ブリッジ
CA型

BCA

クロス
認証型

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 46

PKI の定義 contd.

◆ 公開鍵暗号技術利用環境総体(HW&SW)

The diagram illustrates the overall environment for Public Key Infrastructure (PKI) technology. At the center is a person sitting at a computer workstation, labeled 'アプリケーション (ブラウザ、メール、...)' (Application (Browser, Mail, ...)). This central node is connected to several other components: a 'CA' (Certificate Authority) represented by a building icon; a 'ディレクトリサーバ' (Directory Server) with a 'QUERY' label; a 'ウェブサーバ' (Web Server); an 'Exchangeサーバ' (Exchange Server); an 'ERPサーバ' (ERP Server); a 'ルータ' (Router); and a 'ファイアウォール' (Firewall). An 'ICカード' (IC Card) is also shown at the bottom. The entire system is enclosed in a cloud-like shape.

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 47

本日の講演内容

- ◆ PKI (*Public Key Infrastructure*)という概念
- ◆ PKI要素技術
- ◆ PKIとは何か
- ◆ *PKI Protocols/Applications*
- ◆ 実社会への *implications*


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 48



PKI *Protocols/Applications*

- ◆ S/MIME (*Secure/MIME*)
 - 電子メールに対する暗号化/署名付与
- ◆ Code Signing
 - Java/ActiveX等に対する保護
- ◆ Document Signing
 - PDF等電子書類に対する保護
- ◆ SSL (*Secure Sockets Layer*)
 - セキュアな汎用プロセス間通信機構
- ◆ TLS (*Transport Layer Security*)
 - SSL後継規格
- ◆ IPSec (*IP Security Protocol*)
 - IPレベルでのセキュリティ機能付与
- ◆ PKIX (*Public Key Infrastructure X.509*)
 - Internet上での相互運用可能なPKI構築

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 49



SSL プロトコル概要

OSI 参照モデル

- アプリケーション層
- プレゼンテーション層
- セッション層
- トランスポート層
- ネットワーク層
- データリンク層
- 物理層

Handshake

Change Cipher Spec

Alert

Application Data

Record Layer Protocol

- ・データの分割/組立
- ・圧縮/伸長
- ・暗号化/復号

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 50

SSL プロトコル概要 contd.



◆ Record Layer Protocol

- 2¹⁴ Byte 以下にデータを分割 (もしくは複数を統合)
- (必要なら) 圧縮処理
- (必要なら) 認証データ付加
- (必要なら) 暗号化処理
- 右図のようなデータ構造体を生成して送受



2001年12月6日木曜日
All Rights Reserved, Copyright © 2000-2001, INAMURA You
51

SSL プロトコル概要 contd.



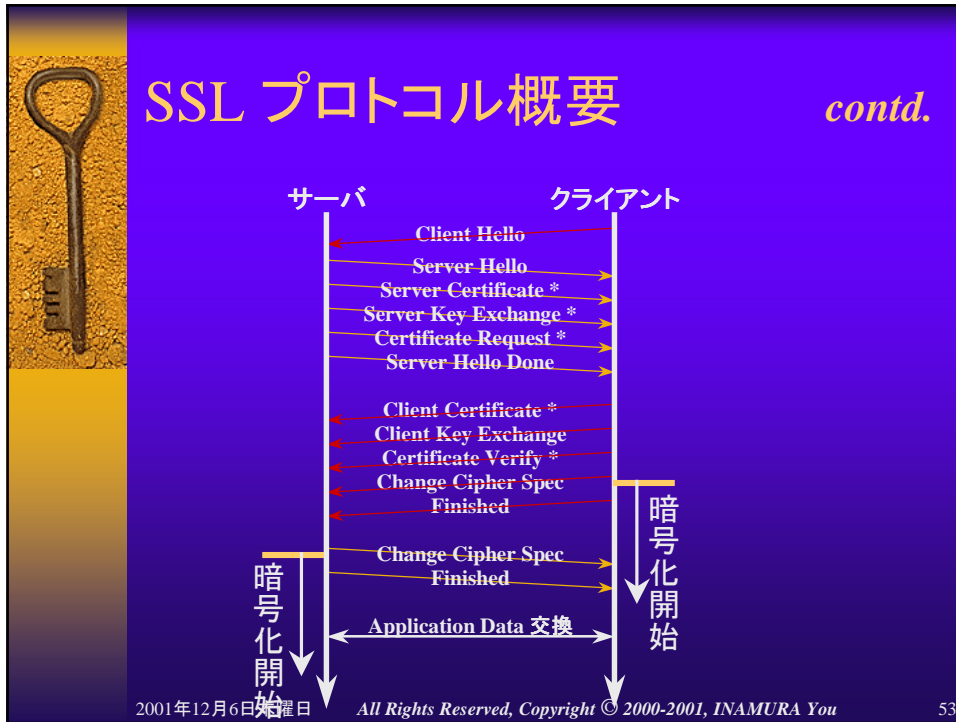
◆ Handshake Protocol

- SSL の実質的な要
- Client/Server 間の接続を確立
 - 暗号化アルゴリズム決定
 - セッション鍵の生成
 - 互いの認証

サーバ
クライアント



2001年12月6日木曜日
All Rights Reserved, Copyright © 2000-2001, INAMURA You
52



SSL プロトコル概要 contd.

◆ C/S Hello

- 暗号化/圧縮アルゴリズムの決定
 - Clientが利用可能なアルゴリズムをリストとして通知し、Serverはその中から実際に使用するアルゴリズムを選択
- 時間データを含み、接続のフレッシュさを保証

Version
(プロトコルバージョン: 2byte)

Time
(10000 毎秒更新: 4byte)

Random Bytes
(16 個の random: 16byte)

Session ID
(通常はS Hello のみ)


暗号化仕様
(C Hello の場合、可能な方式のリスト)

圧縮方式
(C Hello の場合、可能な方式のリスト)

= C/S Random
 マスタ・シークレット
 生成に利用

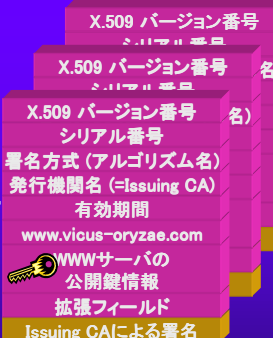
2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 54

SSL プロトコル概要 contd.




◆ **Server Certificate**

- サーバ自身⇒CAの順番にすべての証明書を羅列
- 証明書列を順番に利用することで、最終的にサーバ自身の証明書を検証



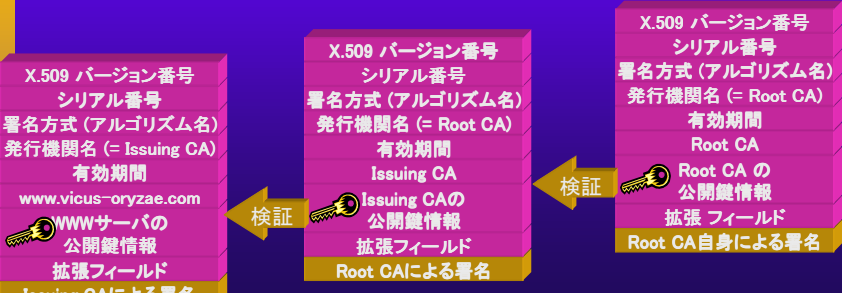
2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 55

SSL プロトコル概要 contd.



◆ **検証**

- サーバ名が証明書のも的一致するか
- 有効期限は切れていないか
- 証明書チェーンにより順番に署名検証できるか
- **有効期限内に破棄されていないか**



2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 56

SSL プロトコル概要 contd.

X.509 バージョン番号
署名方式 (アルゴリズム名)
発行機関名
発行/次回発行日時
破棄された証明書
のリスト
シリアル番号/破棄日時
シリアル番号/破棄日時
:
拡張フィールド
CAによる署名

◆ 破棄の検証

- 秘密鍵漏洩などのため、有効期限到達前に証明書の破棄が必要になる可能性がある
- 一般的なのは CRL (*Certificate Revocation List*) の利用
 - 中途破棄された証明書をCAが署名付リストで公表
 - 差分CRL, CRL配布ポイントなどの改良版が考案中
- オンライン確認プロトコルも
 - OCSP (*Online Certificate Status Protocol*)
 - SCVP (*Simple Certificate Validation Protocol*)
 - 正しい情報源からの回答かどうか確かめるのが厄介
 - CRT (*Certificate Revocation Tree*) というかなり賢い方式もある

CRL

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 57

SSL プロトコル概要 contd.

X.509 バージョン番号
署名方式 (アルゴリズム名)
発行機関名
発行/次回発行日時
破棄された証明書
のリスト
シリアル番号/破棄日時
シリアル番号/破棄日時
:
拡張フィールド
CAによる署名

◆ CRT

- CAが破棄証明書のハッシュ値を元に木を生成後、ルートに署名
- 部分木を知るだけで破棄されたか否かを確実に判断可能
- 特許で保護されているため、一部製品でしか見かけないのが残念

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 58

SSL プロトコル概要 contd.



◆ 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) をクライアントが生成し、サーバ公開鍵で暗号化した上で送付

RSA 利用の場合

Pre-Master Secret
Client Public Key
Server Public Key
Server Random




サーバ公開鍵
で暗号化

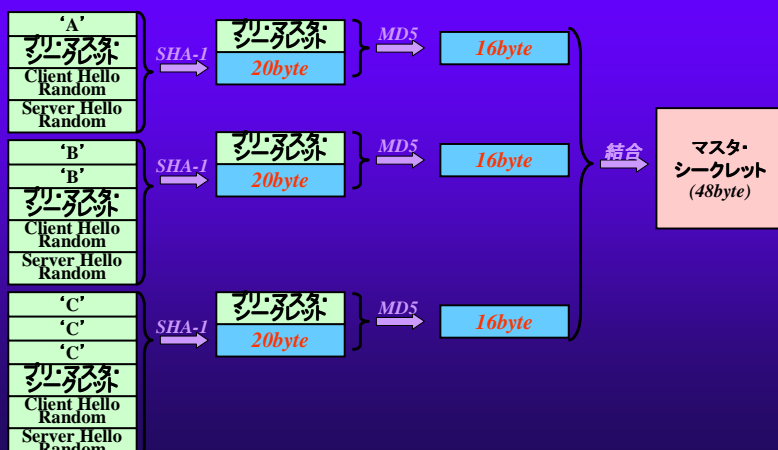
X.509 バージョン番号
シリアル番号
署名方式 (アルゴリズム名)
発行機関名 (= Issuing CA)
有効期間
www.vicus-oryzae.com
WWWサーバの公開鍵情報
拡張フィールド
Issuing CAによる署名

2001年12月6日木曜日
All Rights Reserved, Copyright © 2000-2001, INAMURA You
59

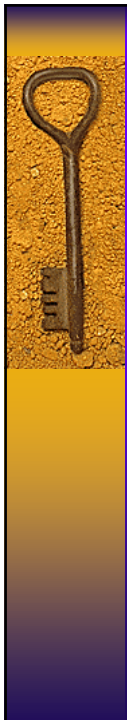
SSL プロトコル概要 contd.



◆ マスタ・シークレットの生成



2001年12月6日木曜日
All Rights Reserved, Copyright © 2000-2001, INAMURA You
60



SSL プロトコル概要 *contd.*

- ◆ 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 + ServerHelloRandom +
ClientHelloRandom+'A'))+
MD5(MasterSecret + SHA(MasterSecret + ServerHelloRandom +
ClientHelloRandom+'BB'))+
MD5(MasterSecret + SHA(MasterSecret + ServerHelloRandom +
ClientHelloRandom+'CCC'))+...

```

+: 結合演算

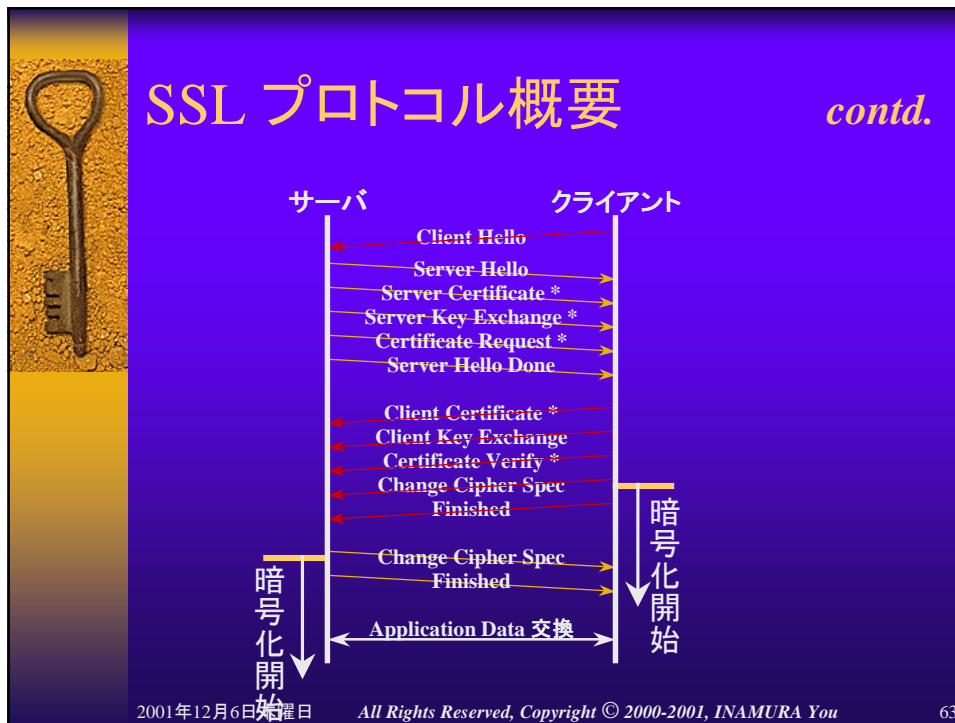
2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 61



SSL プロトコル概要 *contd.*

- ◆ *Change Cipher Spec*
 - それまでのネゴシエーションで合意を得た暗号化方式の利用を開始するという合図
 - 特に相手の*ack*を待ったりはせず、次のメッセージ(*Finished*)から構わず暗号化してしまう
 - 受信側もその旨対処
 - この辺りで、下位プロトコルTCPのパケット順保証に頼っている

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 62



SSL プロトコル概要 contd.

- ◆ **Certificate Request**
 - 所謂SSLクライアント認証を実現するための仕組み
 - クライアントに証明書の送付を要求
- ◆ **Client Certificate**
 - クライアント自身⇒CAの順番にすべての証明書を羅列
 - 証明書列を順番に利用することで、最終的にクライアント自身の証明書を検証可能

X.509 バージョン番号
シリアル番号

X.509 バージョン番号 (名)
シリアル番号 (名)

X.509 バージョン番号 (名)
シリアル番号 (名)

署名方式 (アルゴリズム名)
発行機関名 (= Issuing CA)
有効期間
INAMURA You
証明書所持者の公開鍵情報
拡張フィールド
Issuing CAによる署名

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 64

SSL プロトコル概要 *contd.*

◆ Certificate Verify

- サーバがクライアント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
- **証明書の検証=クライアントの認証ではない**
 - 証明書自体は誰でも入手可能
 - 秘密鍵を使うことを検証して始めて身許が確認
 - 同データをクライアント公開鍵で正しく復号できるか?



クライアント秘密鍵
暗号化

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 65

SSL プロトコル概要 *contd.*

◆ Certificate Verify *contd.*

- それでは、何故 *Server Verify* は必要ないのか?
- 鍵は *Client Key Exchange* の構成方法
 1. サーバ公開鍵で暗号化されたデータを復号するためには、対応する秘密鍵の利用が必須
 2. 同データを元にセッション鍵などが生成される
 3. その後の通信が意味をなす ⇒ 通信相手が正規サーバ秘密鍵を持っていることが保証



サーバ公開鍵
暗号化

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 66



本日の講演内容

- ◆ PKI (*Public Key Infrastructure*)という概念
- ◆ PKI要素技術
- ◆ PKIとは何か
- ◆ PKI *Protocols/Applications*
- ◆ 実社会への *implications*

2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 67



PKI の定義 *contd.*

- ◆ PKI = *Public Key Infrastructure*
 - インフラストラクチャとしての公開鍵暗号技術
 - 交通機関／通信／電力など、他のインフラと同じレベルでの公開鍵暗号技術利用性の提供



2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 68

PKIと実社会取引との比較

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 69

PKIに何が求められてしまうか

機能	実現方法	代替手段
機密情報の保護	暗号化による	共通鍵暗号
取引相手の身許確認	デジタル署名による	Challenge&Response等
取引内容の完全性保証	デジタル署名による	KeyedMAC等
取引相手による否認行為の防止	デジタル署名による	代替手段なし

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 70

PKIにおける否認防止とは

1. 本人の秘密鍵を用いずに署名データを生成することは(計算量的に)不可能
2. 秘密鍵を使い得るのは本人のみ
3. 本人の公開鍵を用いて検証できるデジタル署名が存在するならば、その署名を生成したのは本人

⇒公開鍵

⇒秘密鍵

本来、純粹に技術上の概念

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 71

PKIにおける否認防止とは contd.


1. 十分長い鍵を使えばOK
2. 秘密鍵をどう保護するか?
現状の一般的利用環境
 - ① Windows 9X/ME⇒OSによる保護が弱い
 - ② Windows Logon Passwordのみで保護⇒運用面に関する保護が弱い
3. “デジタル署名の存在⇔本人が署名を生成”という図式は成り立ち得ない

⇒公開鍵

⇒秘密鍵

成立させるためには、社会的普及活動が必須


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 72



PKI/電子署名の実社会への implication

- ◆ 各種行政手続きの電子化⇒電子署名に基づく本人認証
- ◆ 電磁的記録について本人による電子署名が行われているときは、真正に成立したものと推定する(電子署名法)
- ◆ **電子署名に印鑑と同じ効力が持たされてしまう!**
 - 現状では、社会的に印鑑と同程度の認識／受容性が持てていないにも関わらず


2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 73



PKI/電子署名の実社会への implication *contd.*

- ◆ では何が必要か?
 - 安全性
 - 秘密鍵情報の盗用が行なえないように
 - 秘密鍵の利用＝本人性の推定
 - 実的な知識の普及が急務
 - 使いやすさ
 - 非技術者層でも扱えるように
 - **秘密鍵をパスワードで保護してHDに保存するのでは、絶対的に不十分**
 - といった状況下で電子署名法／GPKI が動き出そうとしている
 - さらに、儀式性の確保といった問題も
 - Real world における署名／印鑑押捺と同等の

2001年12月6日木曜日 All Rights Reserved, Copyright © 2000-2001, INAMURA You 74




PKI/電子署名の実社会への implication contd.

- ◆ こんなものが欲しいなあ
 - 電子印鑑
 - 署名機能をすべて含んだ自己充足型耐タンパ・デバイス (with private key activation by biometrics)
 - 仕様のには Java Ring 相当
 - FIPS PUB 140-1 Level 3 耐タンパ性
 - 認証局署名鍵保護用暗号モジュール級
 - 1024bit RSA 署名 ≤ 1sec
 - @ \$65
 - 印鑑型デバイスで用紙型リセプタを撞くと、電子署名がなされる

2001年12月6日木曜日

All Rights Reserved, Copyright © 2000-2001, INAMURA You

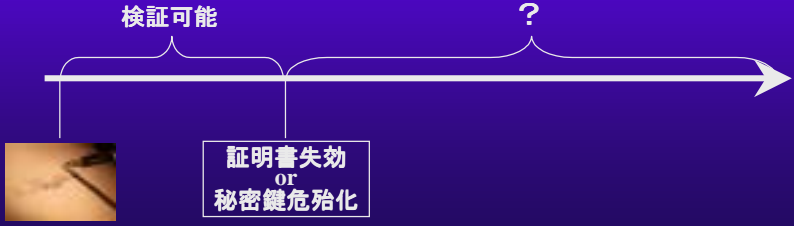
75



さらなる問題

- ◆ 長期間に渡る署名有効性検証
- ◆ 秘密鍵危殆化=偽造署名の作成可能性
 - 証明書に有効期限を定めてはいるが...
- ◆ 一般に、契約行為等に関して争議が発生するのは署名後長い時間が経ってから

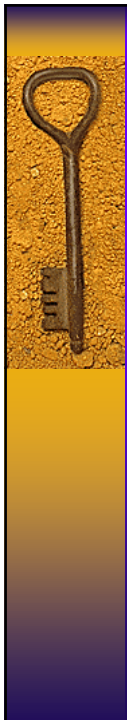
検証可能 ?



2001年12月6日木曜日

All Rights Reserved, Copyright © 2000-2001, INAMURA You

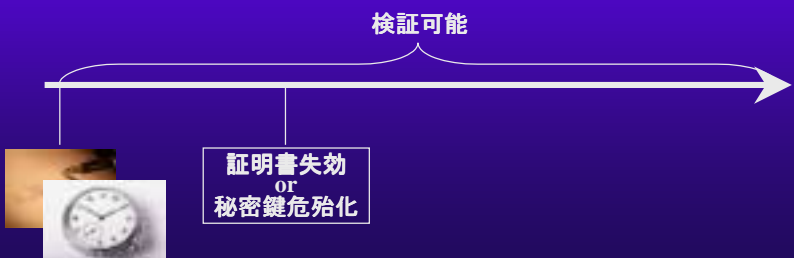
76




デジタル刻時処理

- ◆ 時間軸上のある点である電子データが存在していたことを証明するための技術
 - デジタル署名の補完的役割
 - 刻時処理の時点で有効な証明書で署名されていることさえ確認できれば十分

検証可能



2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 77



デジタル刻時処理 *contd.*

- ◆ 基本は対象データのハッシュ(=指紋)の保管
 - 信頼できる源から得た時刻情報とともに
- ◆ どのように安全・確実に長期保管を実現するか
- ◆ 代表的な手法
 - 十分長い鍵を用いてデジタル署名
 - 紙媒体への記録・公表
 - Surety社によるNewyork Times誌への掲載
 - 物理的に安全な施設を利用
 - ヒステリシス署名
 - 過去の刻時情報・署名を次々に連鎖

2001年12月6日木曜日 *All Rights Reserved, Copyright © 2000-2001, INAMURA You* 78



結論(らしきもの)

- ◆ PKIは何でもこなす、ある意味、万能選手
 - 特に、否認防止機能
- ◆ ただし、もちろん、万全ではない
 - セキュリティとは、プロセスであり、一つの魔法のシステムで賄えるものではない
- ◆ Scalability/Flexibility などの点で、PKIを凌駕(、もしくはそれに匹敵する)仕組は、他には見当たらない
 - 万一、このシステムがこけた場合、*Internet EC* なる代物も終わり、もしくは大きく後退するだろう