認証
―認証プロトコル、PKI―

野口　拓
Taku NOGUCHI

- 設問1：パスワードの管理について記述している以下の文章のうち、最も適切なものはどれか
  - (a) パスワードは忘れてしまうと大変なので、確実に憶えている同じものを数年間使い続けている。
  - (b) いろんなWebサービスを利用しているが、パスワードをいちいち憶えられないので全部同じものを使っている。
  - (c) パスワードは自分の名前と誕生日にしておけば、忘れないので安心だ。
  - (d) パスワードは、毎年新しいものに更新している。
- 答え：

- 設問2：多数のホストを協調させて、大量のパケットを送りつけるなどする攻撃を何というか
  - (a) DDoS
  - (b) DNSキャッシュポイズニング
  - (c) War Driving
  - (d) Brute Force Attack
- 答え：

- 設問3：サーバにJavaScriptのプログラムを送り込み、それをユーザのブラウザで実行させてCookieなどに書かれたセッションIDなどの情報を盗む攻撃を何というか
  - (a) クロスサイトリクエストフォージェリ
  - (b) クロスサイトスクリプティング
  - (c) SQLインジェクション
  - (d) マルウェア
- 答え：

- 設問4：Webにおける通信について説明した以下の文章のうち、間違っているものはどれか
  - (a) HTTPは通常1回の通信では処理が完結しないので、セッションIDなどを使ってTCP接続間に跨がった認証をする必要がある。
  - (b) サーバから送られたセッションIDは、URLリンクやHTMLのフォーム、Cookieなどの形式でブラウザに保存される。
  - (c) Cookieは保存されたら二度と消すことができない。
  - (d) サーバに脆弱性があると、クロスサイトスクリプティングなどの攻撃を通じてセッションIDが盗まれ、通信を乗っ取られる可能性がある。
- 答え：

認証
―認証プロトコル、PKI―

野口　拓
Taku NOGUCHI

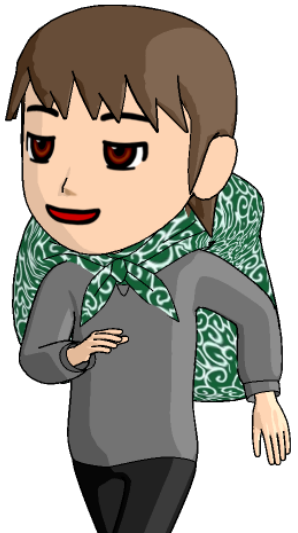
アクセス制御には認証が必要

- 認証とは？
  - 相手に自分が「誰か」を伝えること
- 何を使って伝えるか？
  - 自分しか持ち得ない「何か」＝認証要素
    - 名前、電話番号、ID、指紋、声、顔写真・・・
    - パスワード、暗号表、認証デバイス・・・
- 所有していることを「証明」するには？
  - PKIのクライアント証明書、ゼロ知識証明・・・

もっとも簡単な認証：パスワード

- ユーザ認証に最もよく使われる「合言葉」
  自分と相手しか知らないのが前提

- よいところ：使い方が簡単で広く知られている
- 悪いところ：盗まれやすい・盗まれても気付かない

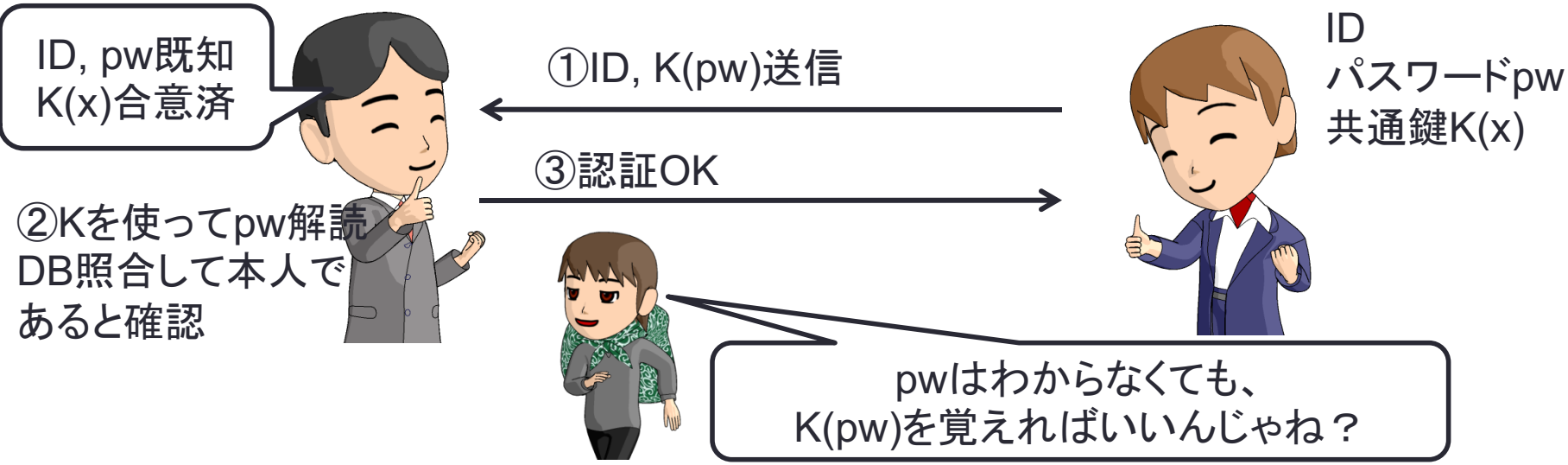


パスワードを使う認証プロトコル

プロトコル：protocol

- 暗号化されていればパスワードは漏れない
  - LANなど盗聴の危険が少ない場合も
- 単純なプロトコルはリピート攻撃に弱い！

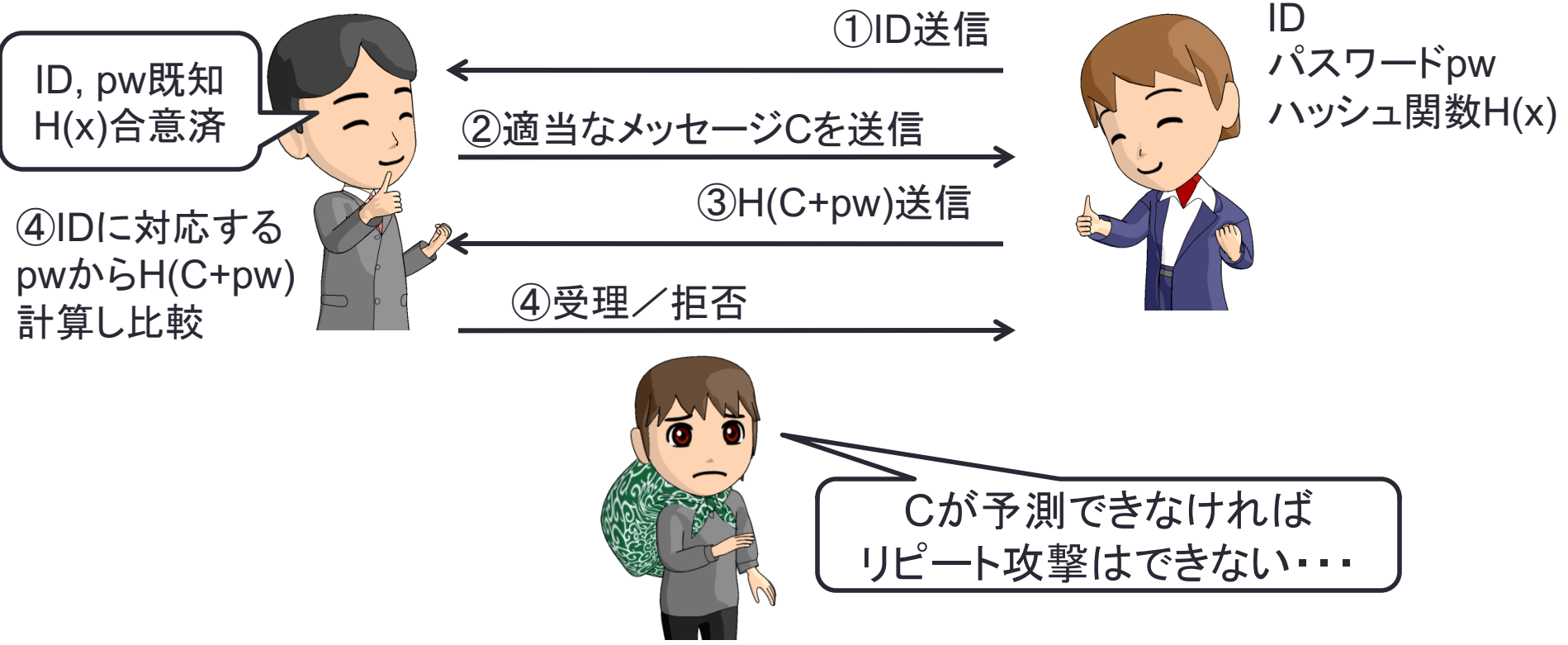
リピート：repeat



チャレンジ-アンド-レスポンス認証

Challenge and Handshake Authentication Protocol (CHAP)等

- 送信者と受信者がお互いにやり取りができる前提で行う認証



ところで利用サービスは パスワードをどう管理している？

サービス： service

- パスワードを忘れたときにパスワードを教えてくれるかどうか
  - 「登録したメールアドレスにパスワードを送ります」
    「秘密の質問に答えて下さいパスワードを表示します」
    →確実に認証DBにそのままパスワードを保存
  - 「登録したメールアドレスに送るURLにアクセスを」
    「登録したメールアドレスに新しいパスワードを送ります」
    →ハッシュして保管している（管理者は知らない）可能性がある
    - ただしメールは盗聴できる・・・

それではパスワードリマインダーのあり方はどうあるべきか？

パスワードリマインダー： password reminder

パスワードは覚えられないので・・・

- 別の「複製しにくいもの」に入れる
  - コンピュータそのものに覚えこませる（一種の機器認証）
    - コンピュータそのものに誰かに触られたり
      ウィルスがかかるとアウト
  - 紙に書いてある
    - インターネットバンキングでの暗号表など
      そのまま送るのは意味がないので工夫をする
  - ICカード／USBキー：「ハードウェアトークン」
    - これらは固有の数字が入っている
    - 亡失がわかるのがメリット
- そもそも複製しにくい固有の情報を使う
  - 代表例がバイオメトリックス：生体情報
    - 指紋・静脈・声紋・網膜・虹彩などなど
    - 亡失がまずないのが最大のメリット

アウト：out

インターネットバンキング：Internet banking

ネットバンキングで
各銀行の対応が
それぞれ違うのが
興味深い

ハードウェアトークン：hardware token

バイオメトリックス：biometrics

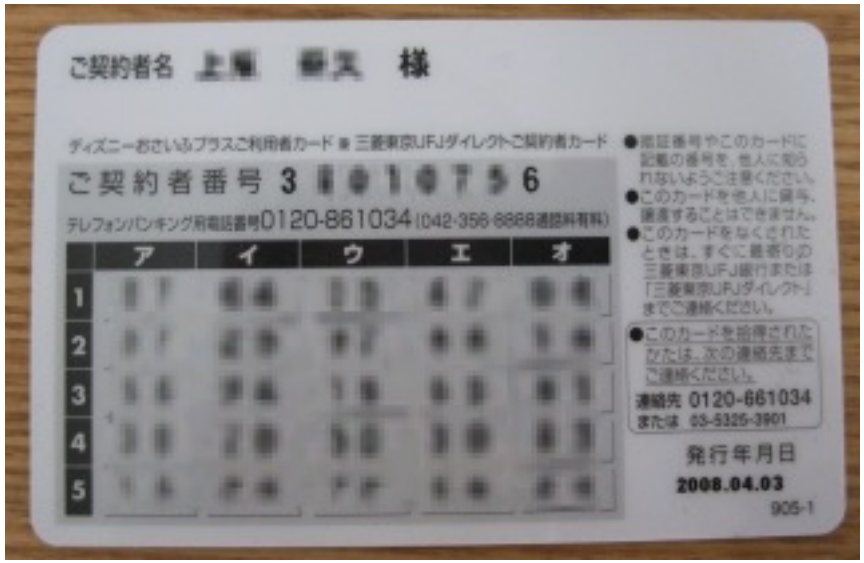
銀行ATMでは
指静脈陣営と
手のひら静脈陣営に
分かれている

銀行はネットバンキングのために多くの認証システムを導入

- 多くはパスワードと併用:「多要素認証」

RSAワンタイムパスワード

1分ごとにどんどんパスワードが生成される機械



その都度「アの1にある数字を入力して」などといわれる

どういうしくみなんだろう？

ワンタイムパスワード：one time password

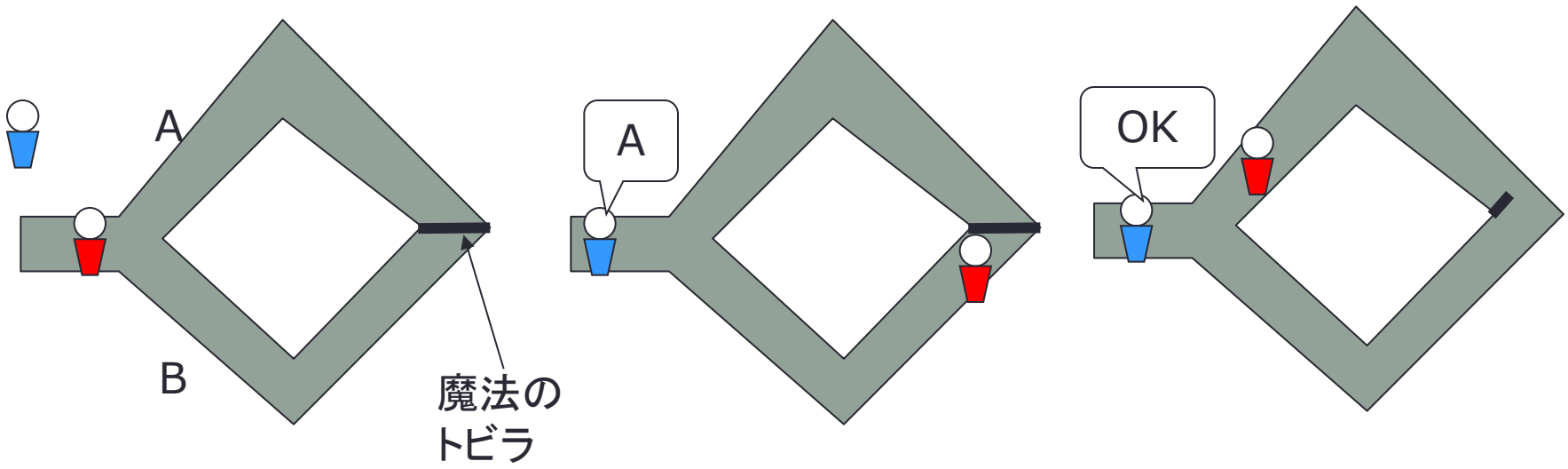
ゼロ知識対話認証

ゼロ知識：zero-knowledge

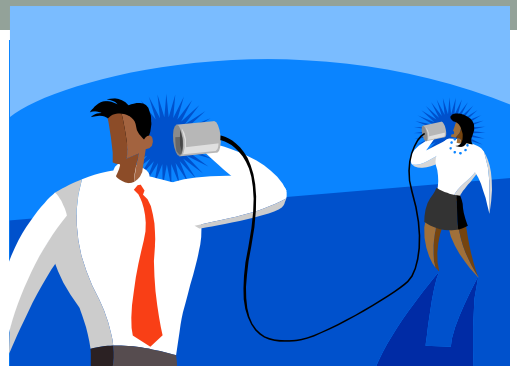
- 「秘密」を送信者から受信者に
  送らなくても「秘密を持っている」ことを
  証明できる認証
  - 送信者は秘密情報sを持っているが
    それを受信者にも教えたくない！
  - でもsの持つ性質は言えるのでそれを用いて
    sを知っていることを公開する
  - 受信者は送信者にいくつか質問をして
    本当にその知識を知っているか確認してゆく
- つまりサーバは送信者の秘密を管理しなくてよい

ゼロ知識証明の簡単な例

- 輪になった洞窟の奥に「魔法の扉」がある
- アリスは洞窟の奥の「魔法の扉」を開ける呪文を知っていると主張
  でも呪文は秘密
- ボブはアリスを洞窟の奥まで行かせてから洞窟の入り口に立ち、A/B
  どちらかから出るように叫ぶ
  - アリスが最初どちらに入ったかはわからない
  - アリスが呪文を本当に知っていれば言ったほうから出られるハズ
- これを繰り返して認証する



実際のゼロ知識証明の例



- 送信者Pと受信者Vはmod Nを使う合意
- Pはsを秘密として$v = s^2 \bmod N$を公開
  - vからsを求めるのは困難
- Pは乱数rを選んで$x = r^2 \bmod N$をVに送信
- VはPに0か1かを送信
- Pは0なら$y=r$を、1なら$y=rs \bmod N$をVに送信
- Vはy^2を計算　0の時は$r^2 \bmod N$と、
  1の時は$r^2v \bmod N$と比較
- これを繰り返してVはPを信頼していく
  - n回連続で嘘をつける確率は2^{-n}なので7回で1%以下、10回で0.1%以下、20回で百万分の一以下

実際にやってみる

- P, V : mod Nを使おう！Nは19にしよう
- P: v= 5 = s^2 mod N だよ。s=256は秘密
- P(1): $r^2=136^2$
- V(1): 0
- P(1): y=136 (0: y=r)
- V: 136^2 mod 19 = 9 = 136^2 mod 19 →OK
- P(2): $r^2=136^2$
- V(2): 1
- P(2): y=8 (1: y=rs mod N → 136×256 mod 19)
- V: 8^2 mod 19 = 7 = $136^2 \times 5$ mod 19 →OK
- これをつづける. . .

s=256は秘密でもVはPがこの秘密を知っていることがわかる

ウソが成功する確率は毎回1/2

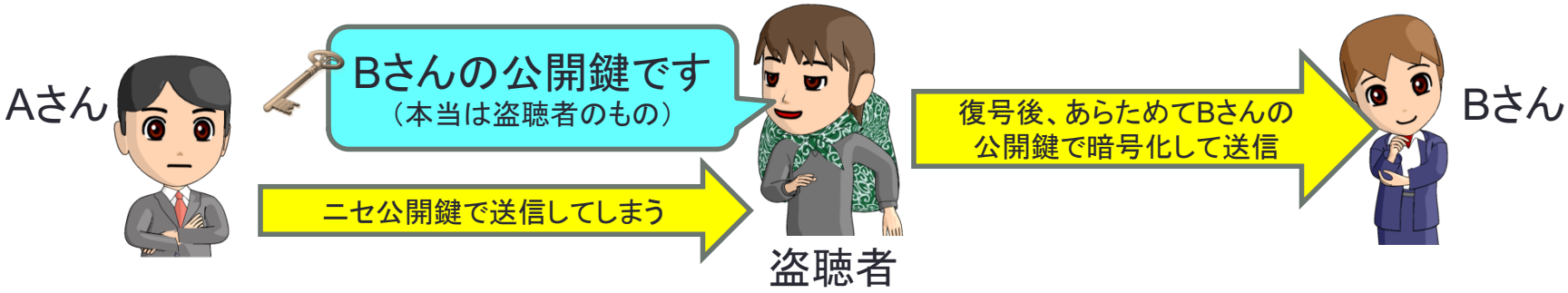
- Pが偽者だった場合、まずVが0というか1というか
  あらかじめ予想する
  - 0をいうと予想した時：v=5
    乱数rを作ってx＝r^2 mod NをVに送信
    質問されたらy=rを送信
  - P’: r = 10, r^2=100, P’: y = 10 → V: 100 mod 19 = 5 = 10^2 mod 19

  - 1をいうと予想した時：v=5
    乱数rを作ってx＝r^2/v mod NをVに送信
    質問されたらy=rを送信
  - P’: r = 10, r^2=100/5 = 20, P’: y = 10
    → V: 20×5 mod 19 = 5 = 10^2 mod 19

  - どちらかなら成功するので毎回確率は1/2

公開鍵暗号と認証

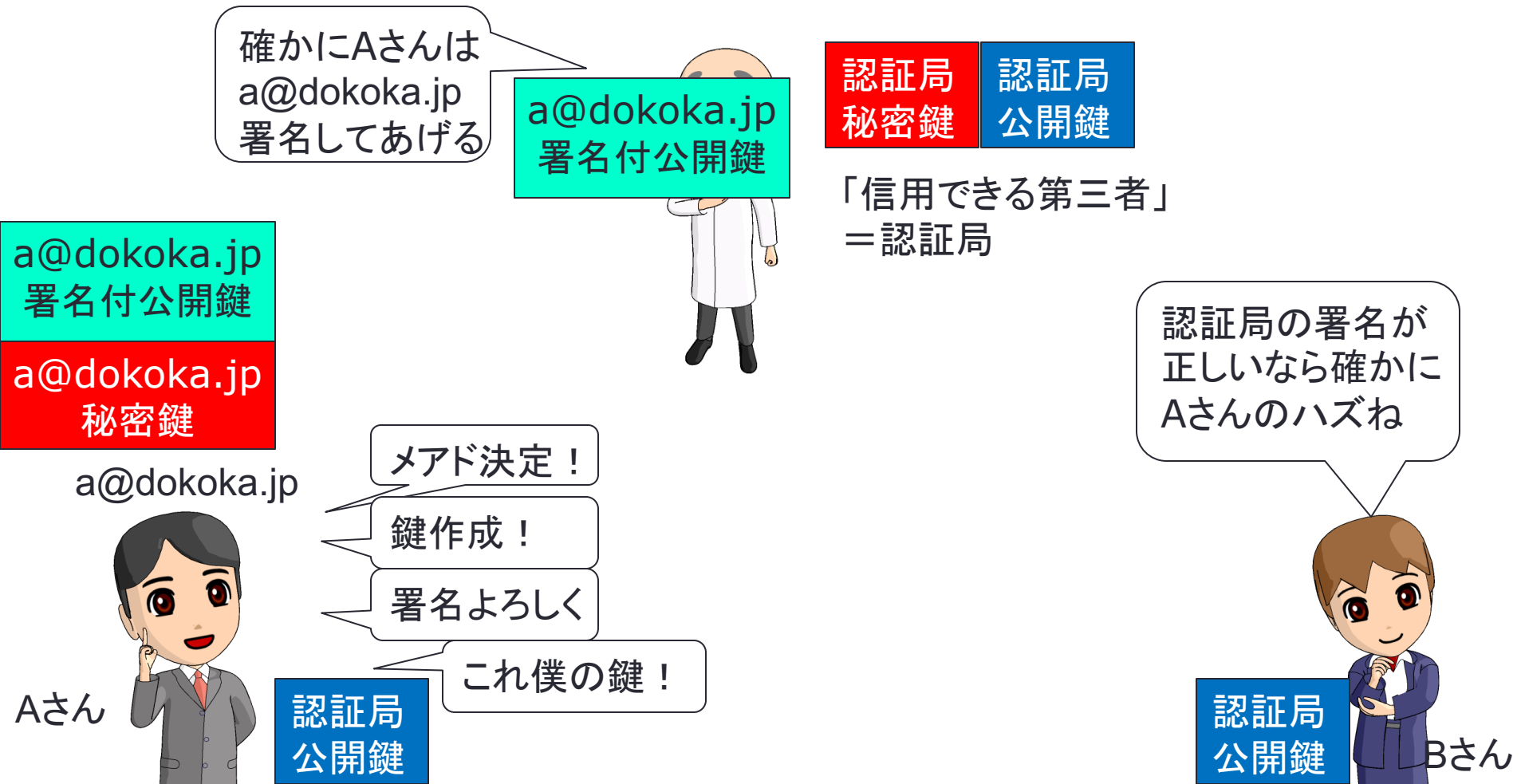
- 公開鍵暗号と電子署名の組み合わせは
  相互の認証を可能にする
- ただし、公開鍵が確実に通信相手のものか
  確認しなくてはならない
  - 他者のものとすり替えられると中間者攻撃が成立



- 公開鍵の確認が必要＝PKIの出番

PKIの基本的考え方

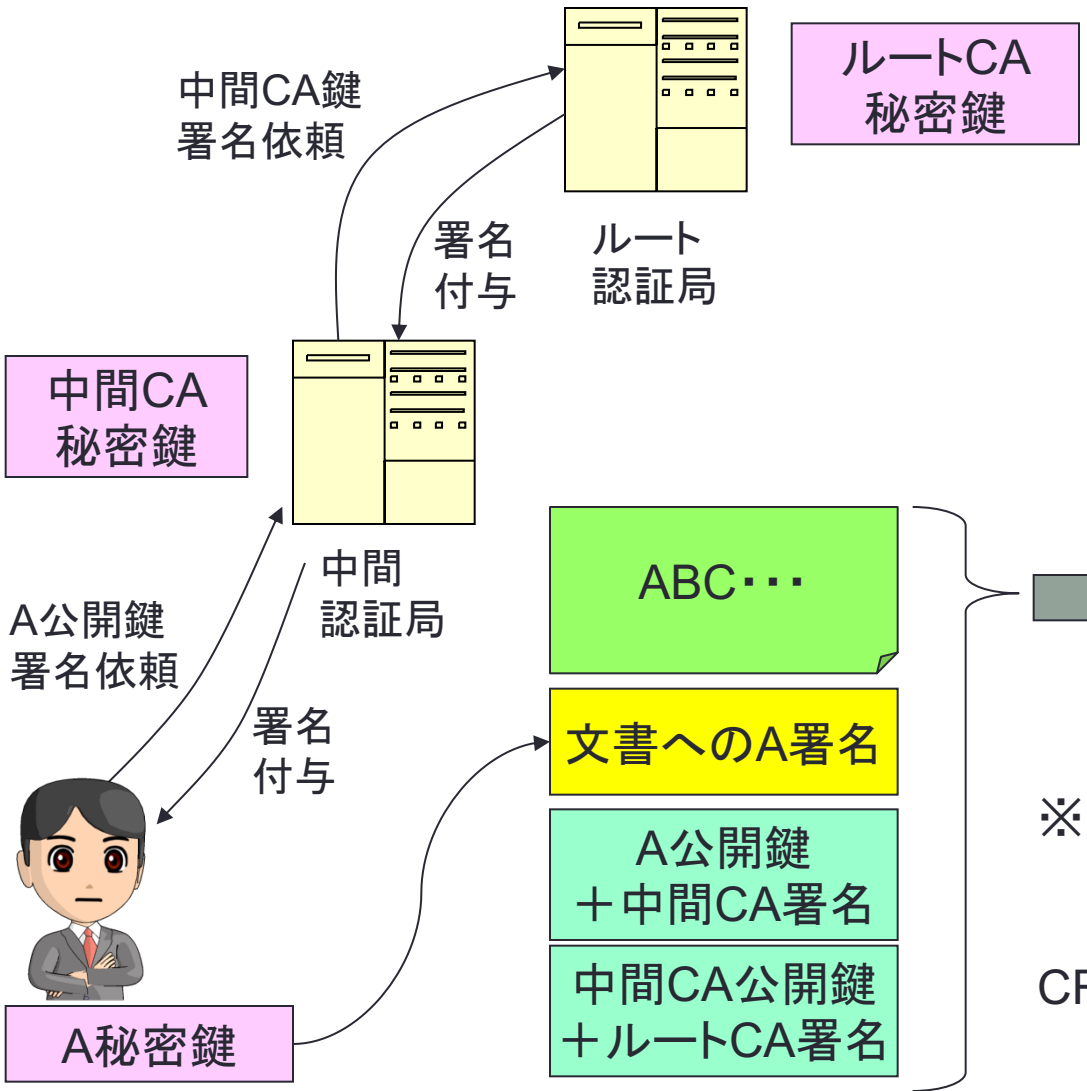
- 公開鍵を「信用できる第三者」に署名してもらう



PKIの構造

CA=Certification Authority
認証局

ルート：root



①ルートCA署名のついた
中間CA公開鍵を確認
②その公開鍵を用いて
Aの公開鍵を確認
③そのAの公開鍵を用いて
文書の署名を確認

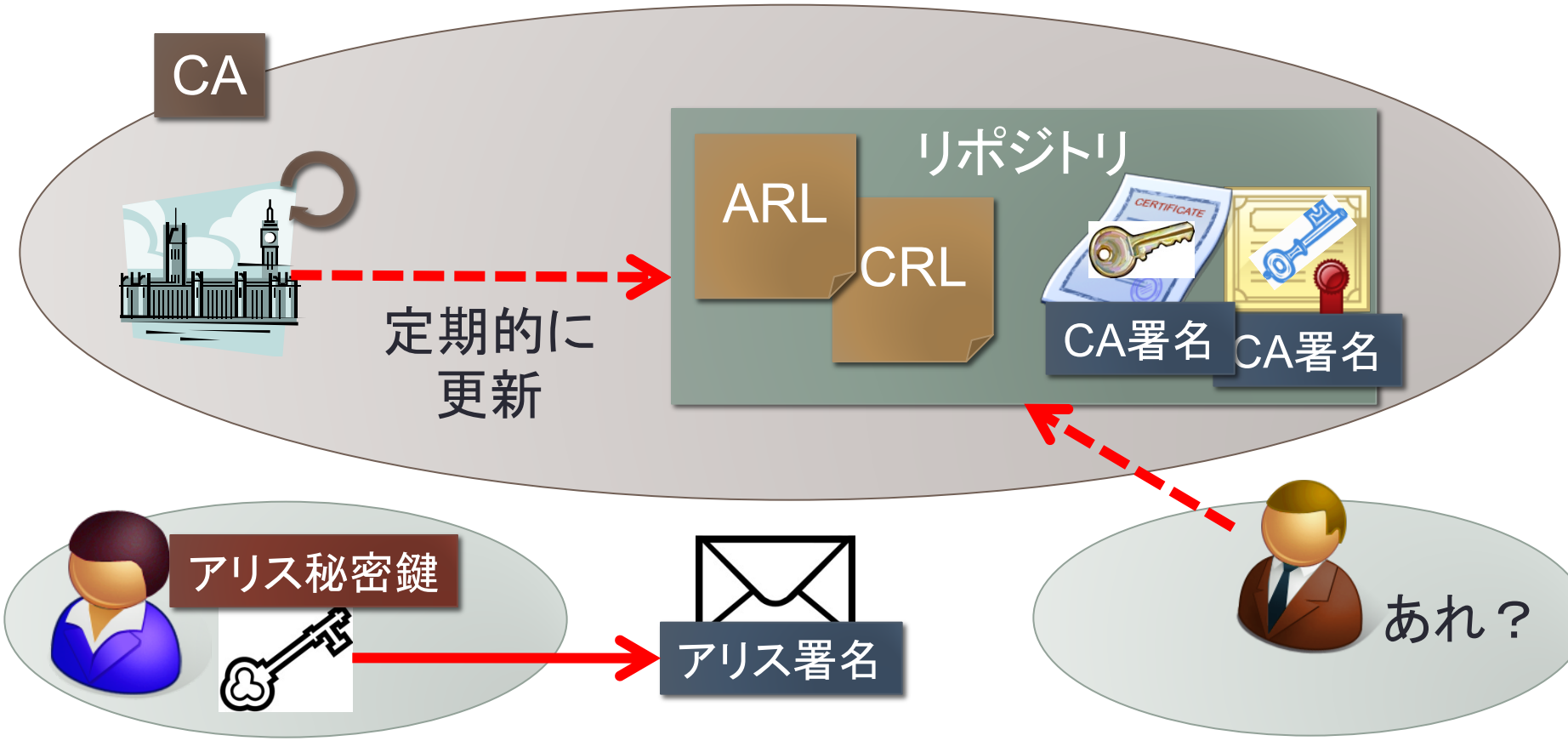
ルートCA
公開鍵

※厳密にはさらにAの公開鍵が
失効していないか中間CAに
問い合わせる必要(CRL確認)
CRL=Certificate Revocation List

証明書の失効

リポジトリ：repository

- 証明書の品質維持のためには有効期限が必須
  - CRL(Certificate Revocation List) ･･･ 利用者の証明書
  - ARL(Authority Revocation List) ･･･ 認証局の証明書



ルートゾーン：root zone

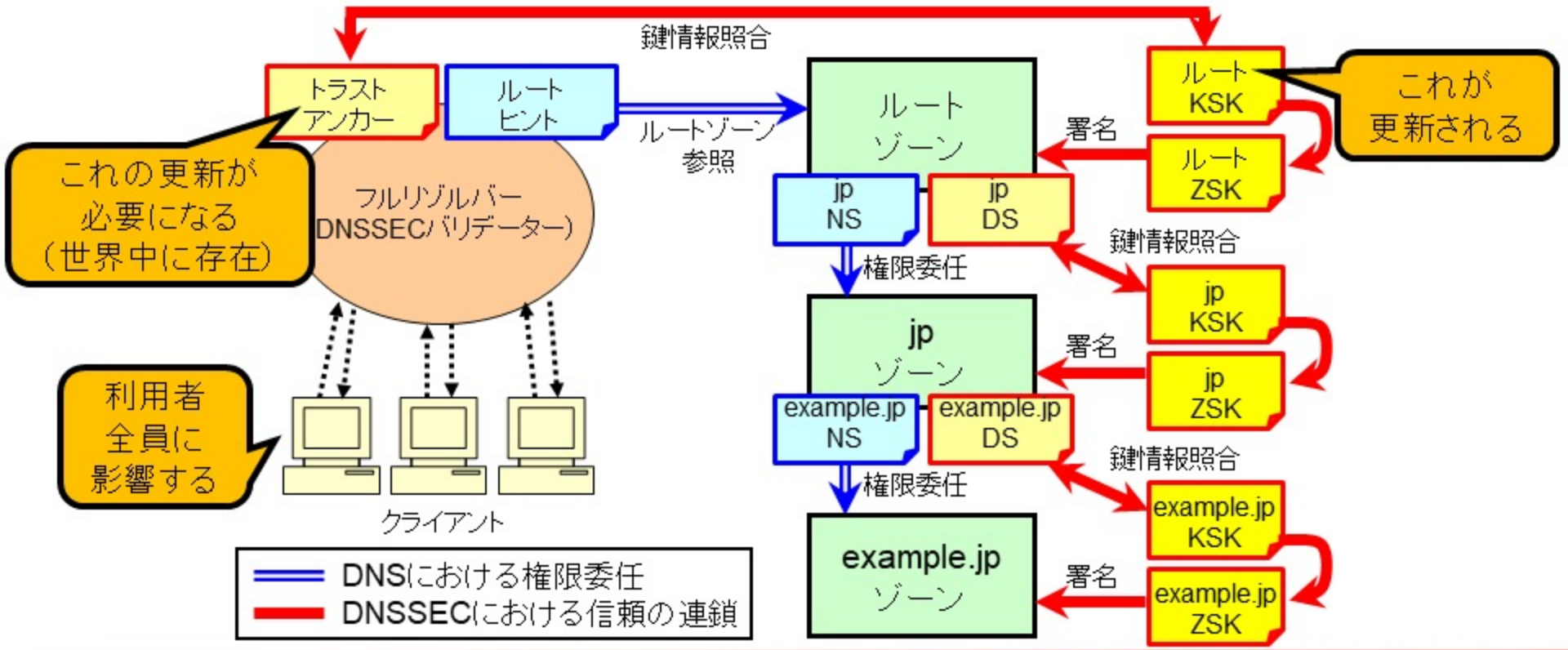
ロールオーバー：rollover

最近の話：ルートゾーンKSKロールオーバー

トラストアンカー：trust anchor

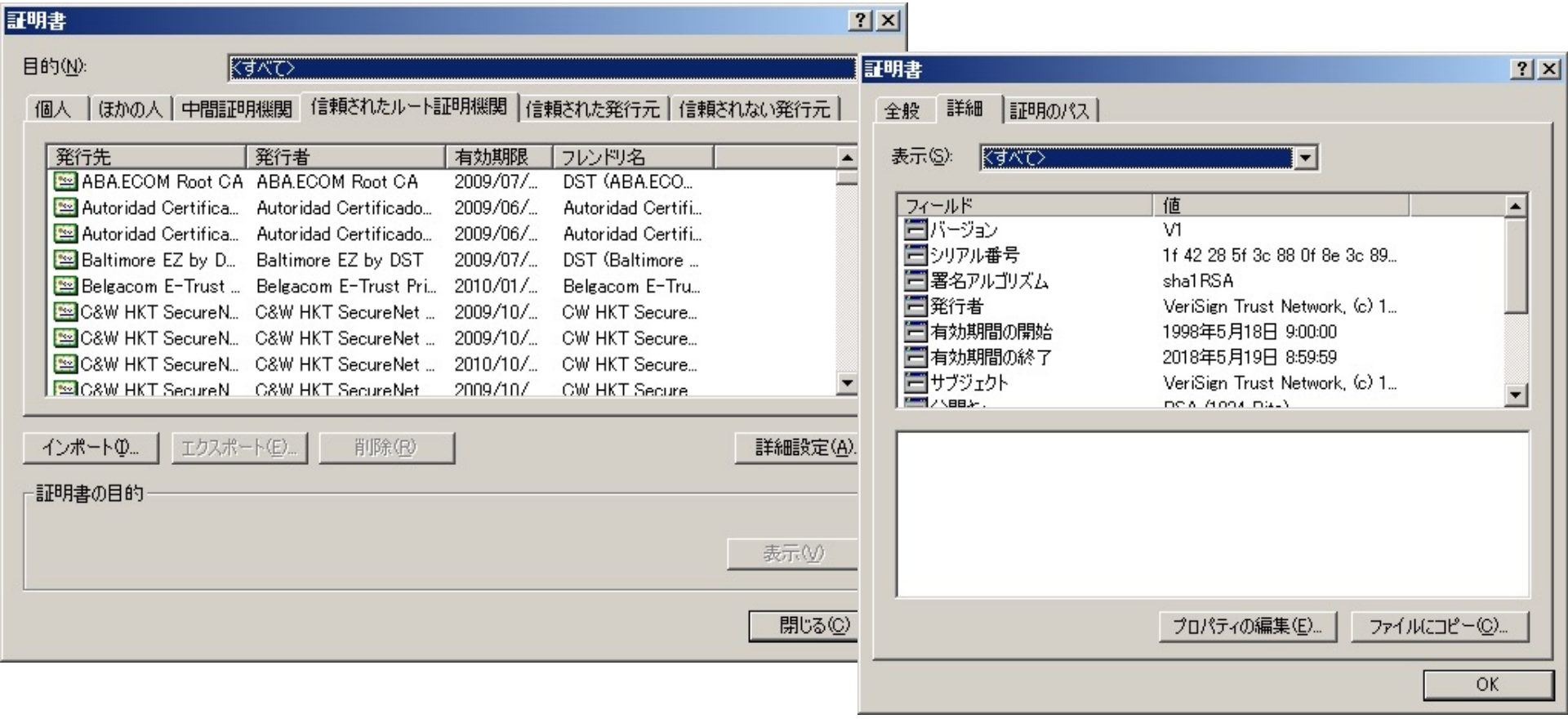
- 2018/10/11：DNSの大元であるルートサーバの鍵の更新
- うまくいかなかったら7.5億人以上が影響を受けた

ルートヒント：root hints



WindowsにはたくさんのルートCAの公開鍵があらかじめ入っている

- インターネットオプション-コンテンツ-証明書



PKIは誰が運営しているか

- ルートCAを運営している会社
  - Verisign(最大手）
  - GlobalSign（安売りもする）
  - Comodo, GoDaddy (価格破壊を先導）
  - セコムトラストシステムズ、日本認証サービス（日本から）
- これらの会社はルートCAを運営すると共に
  その公開鍵をOSやブラウザに導入してもらうよう
  働きかけている
  - 米加の公認会計士の定めたWebTrustという監査に合格し認定を受けたCAの公開鍵を受け入れるのが一般的
  - 最終的にはマイクロソフトやFirefox(Mozilla)の判断

公開鍵の容れ物:「電子証明書」

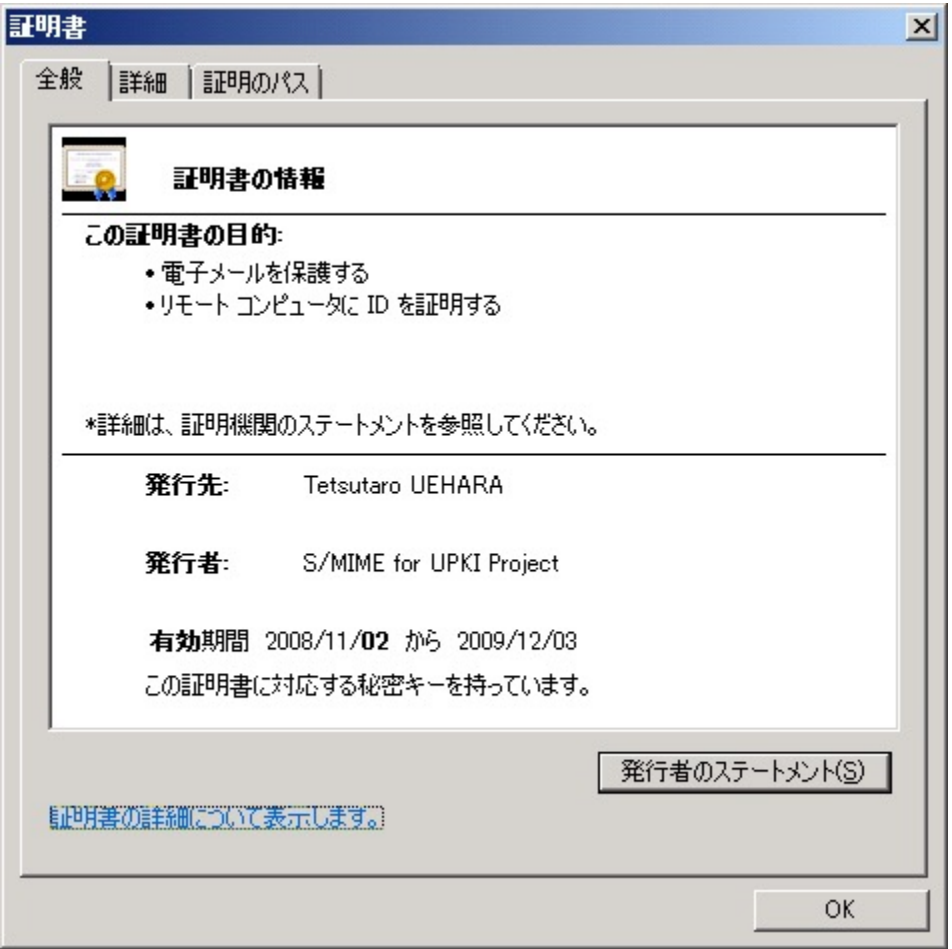
- 証明書には以下のものが入っている
  - 名前やID(メールアドレス、URLのホスト部)など
  - 公開鍵
  - 証明書を署名した認証局(のリスト)
    - ルート認証局まで遡れるまでの全認証局の証明書
  - 有効期限
  - CRLの公開場所
- チェックするべきは・・・・
  - 有効期限は大丈夫か?
  - 認証局(特にルート認証局)は
    自分が知っている(=信用できる)か?
  - アドレスなどはその証明書を送ってきた者と一致するか?
  - CRLで無効化されていないか←チェックしない場合も多い
    Vista+IE7以降はチェックする

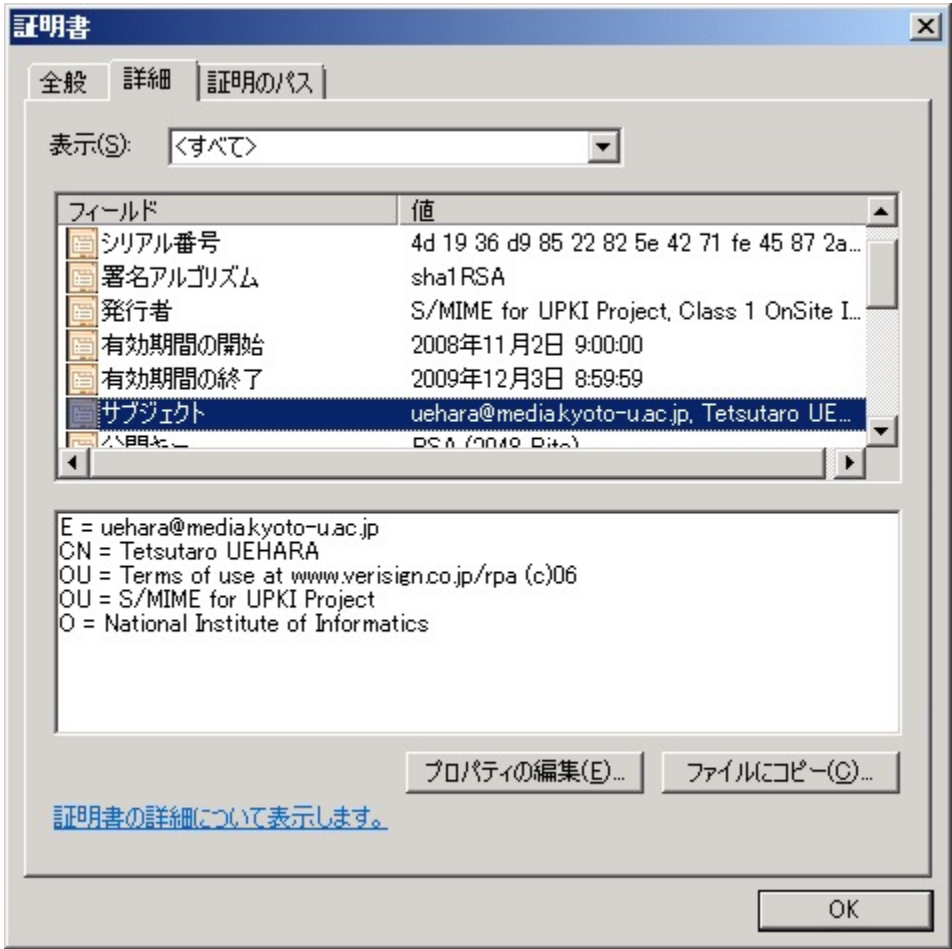
もちろん
署名が
ついている

もちろん
署名も
チェックする

実際の公開鍵・秘密鍵

- インターネットオプション-コンテンツ-証明書



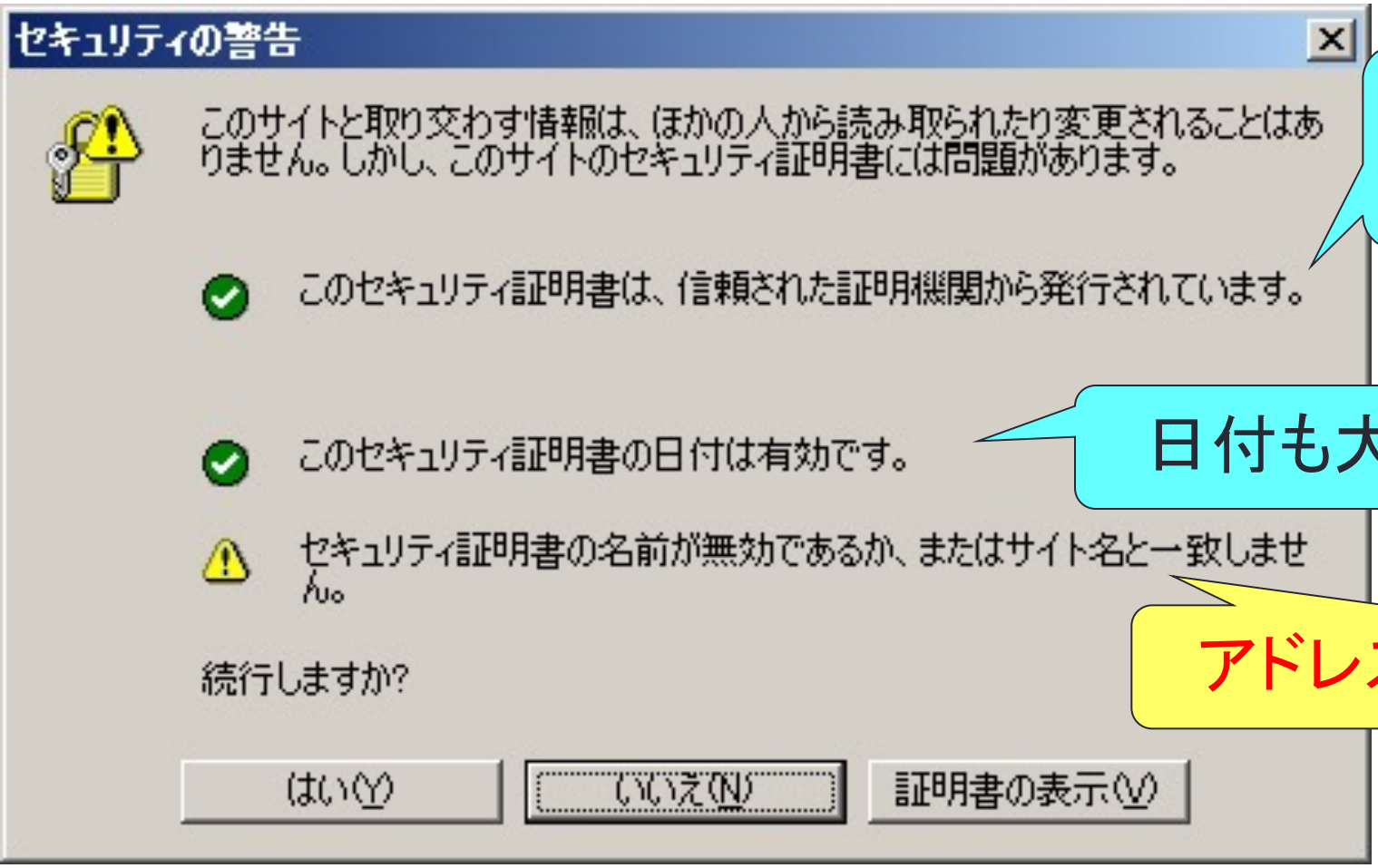


PKIの具体例

- SSL: Secure Socket Layer
  : TCP通信をPKIを使って保護する仕組み
  - 後継＝TLS(Transport Layer Security)
  - httpに適用したものをhttps、
    SMTPに適用したものをsmtpsなどと表記する
- 機能：公開鍵暗号を用いた暗号化と認証
  - 暗号化は常に行われる
  - サーバ認証もしくはクライアント認証
- よくある https://www...というURLを持つ
  ホームページはまさにSSL（TLS）によって保護されている
  - ブラウザの隅に南京錠のマークが出るはず

暗号化ばかりが言われるが認証もとても重要

証明書のチェックに失敗すると

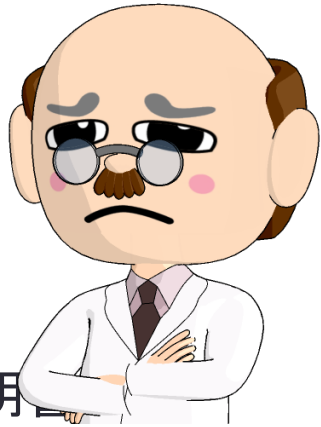


信頼できる CA

日付も大丈夫

アドレスが違う！

信頼できない証明書



- かつて証明書の発行は高額な買い物だった
  - 勝手に独自のCAを立てる人が続出（自己認証局→自己署名証明書）
  - 最近は価格破壊が進み、少ない予算で可能に

- 政府機関や自治体による独自のPKIの構築
  - かつて政府認証基盤GPKI、地方公共団体認証基盤LGPKI
    公的個人認証基盤JPKIのルートCAの証明書は残念ながら
    Windowsに入っていなかった
    - 最近IEの機能により「自動で」GPKIのCA証明書が入るようになった
      （日本政府対応のための特別措置！）
    - LGPKIとGPKIは最近WebTrust for CA認証を取得
      →WindowsのCAアップデートで正式に更新
    - 問題はJPKI・・・なんと都道府県ごとにCAがある

「オレオレ証明書」問題

オレオレ証明書： Self-signed certificate

- 信頼されたCAから発行されていないと
  中間者攻撃の可能性を排除できない！

とはいえ、中間者攻撃は難しいんじゃないの？

サーバ証明書に払うお金はもったいない
自己認証局のサーバ証明書でも暗号化はできる
盗聴のほうが怖いんだから何もしないよりマシ？

Webページに「警告が出ますが心配いりません」
とか書いておけばいいんじゃないかな
あるいはCA証明書をダウンロードしてもらえば。

そうはいかんのです！

- 特に不特定多数が利用する場合はダメ
  「警告が出ますが心配ありません」の
  意味をよく考えずにクリックすることにユーザが慣れる
  - 慣れてしまってから本物の詐欺サイトがあらわれると
    引っかかる可能性が高まる

悪い習慣を広める一役を買ってしまう

- CA証明書の導入は「何が信頼できるか」の根幹に
  関わるが、普通のユーザにそれが理解できる？
  - 慎重にすべきことが一般化するのは怖い

オレオレ証明書問題は根が深い！