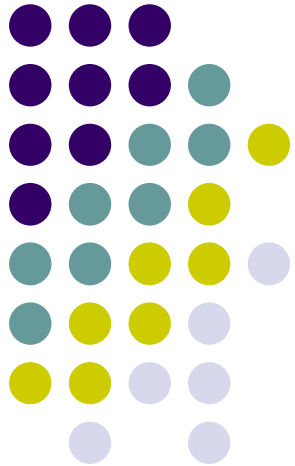
本資料について

- 本資料は下記論文を基にして作成されたものです。文書の内容の正確さは保証できないため、正確な知識を求める方は原文を参照してください。

- 著者：Kazuomi Oishi, Haruyuki Kitawaki
- 論文名：Anonymous IPsec with Plug and Play:
  a prototype of IPsec with IKE using IPv6 temporary addresses and anonymous public-key certificates
- 出展：IC2004
- 発行日：2004年10月

プラグアンドプレイで実現する匿名IPsec

渡辺研究室
040427177 細尾幸宏

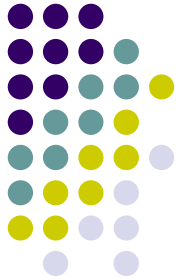


背景

- IPv6デバイスが普及すると通信相手の認証を行う場合，IPsecの自動管理プロトコルIKE(Internet Key Exchange)が事前共有鍵または証明書を使用して秘密鍵を共有する
- 通信相手すべてに事前共有鍵を共有することは不可能
- 通信時にユーザ同士が互いに匿名のまま通信を行いたい場合，通常のIKEが使用する証明書には証明書保持者の情報が含まれているため，望ましくない

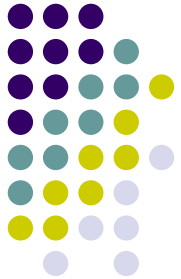
- 中間者攻撃を防止しつつ，互いに匿名のまま認証できる仕組みが必要

拡張IPv6ステートレスアドレス自動設定



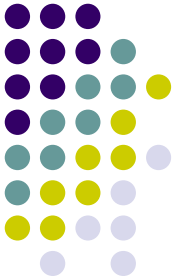
- IPv6のアドレスを自動的に設定する方法の1つ
- インタフェースIDが常に同じである問題
  - 修正EUI－64形式 公共アドレス
- プライバシ拡張(RFC3041)によって解決
  - ランダムに生成　　一時アドレス

IPsec



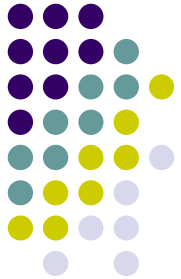
- IP層で秘密性と保全を提供するセキュリティアーキテクチャ
- 2つのプロトコル
  - AH (Authenticating Header)
  - ESP (Encapsulating Security Payload)
- 2つのモード
  - Transport mode
  - Tunnel mode
- SA (Security Associations)による特定
  - SPI (Security Parameter Index)
  - 宛先IPアドレス
  - セキュリティプロトコル(AH, ESP)のID
- 暗号鍵を共有している必要がある

IKE (Internet Key Exchange)



- IPsecにおけるデフォルトの自動管理プロトコル
- 秘密鍵の共有はDiffie-Hellman鍵交換を行う
- 4つの認証方式
  - 事前共有鍵
  - ディジタル署名
  - 2種類の公開鍵暗号による認証
- 事前共有鍵は通信相手すべてと共有することは不可能
- 他の3つは公開鍵の認証が必要
- X.509v3形式の公開鍵証明書により認証

匿名性

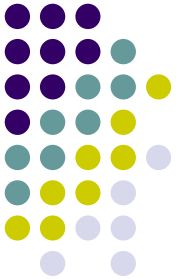


- 匿名性
  - プロトコル実行後にIDを特定されない
- Unlinkability
  - プロトコルを複数回実行後に第三者がプロトコルの動作にあるエンティティが参加しているかどうか決定不可
  - プロトコルが別に実行された場合に同じエンティティが参加したか決定不可
- Untraceability
  - 匿名性とunlinkabilityの両方を満たす

コンピュータに対する仮定

- コンピュータの暗号に関する仮定
- 離散対数問題(DLP)
  - 有限体G, Gのジェネレータgが与えられたとき
  - v∈Gの離散対数は $\log_g v$
  - Gのオーダが十分に大きければ計算は不可能

IPsecとIKEの仕様の問題



- 一時アドレスを使用することで得られる匿名性やunlinkabilityの特性は偽装ノードや代理鍵を使って悪用される可能性
- IPsecは秘密鍵を互いに保持していることが前提であり, IKEが鍵交換を行う
- 一時アドレスを使用して証明書を作成する場合, CA(Certification Authority)による一時アドレスの割り当ての保証が問題になる

排他的なアドレスの保証

- CAが一時アドレスの有効期限が来るまで同じアドレスを含む別の証明書を発行しないことで保証

匿名性の問題

- 公開鍵のunlinkability
  - 公開鍵は証明書が要求されたときに作られる必要
  - 鍵を含む匿名証明書の使用はただ1度だけであるべき
- CAの能力を最小にする
  - CAが証明書ユーザの秘密鍵を計算できてはならない

解決策

- CAはsubjectAltNameとして一時アドレスを含むX.509v3匿名公開鍵証明書を発行
- エンティティは証明書署名要求（CSR; Certificate signing request）を送ってCAに証明書を要求
- CSRのIPパケット転送中は送信元アドレスは一時アドレス
- CAは一時アドレスのためのX.509v3匿名公開鍵証明書を作成し, 送信
- 証明書をIKEにより認証

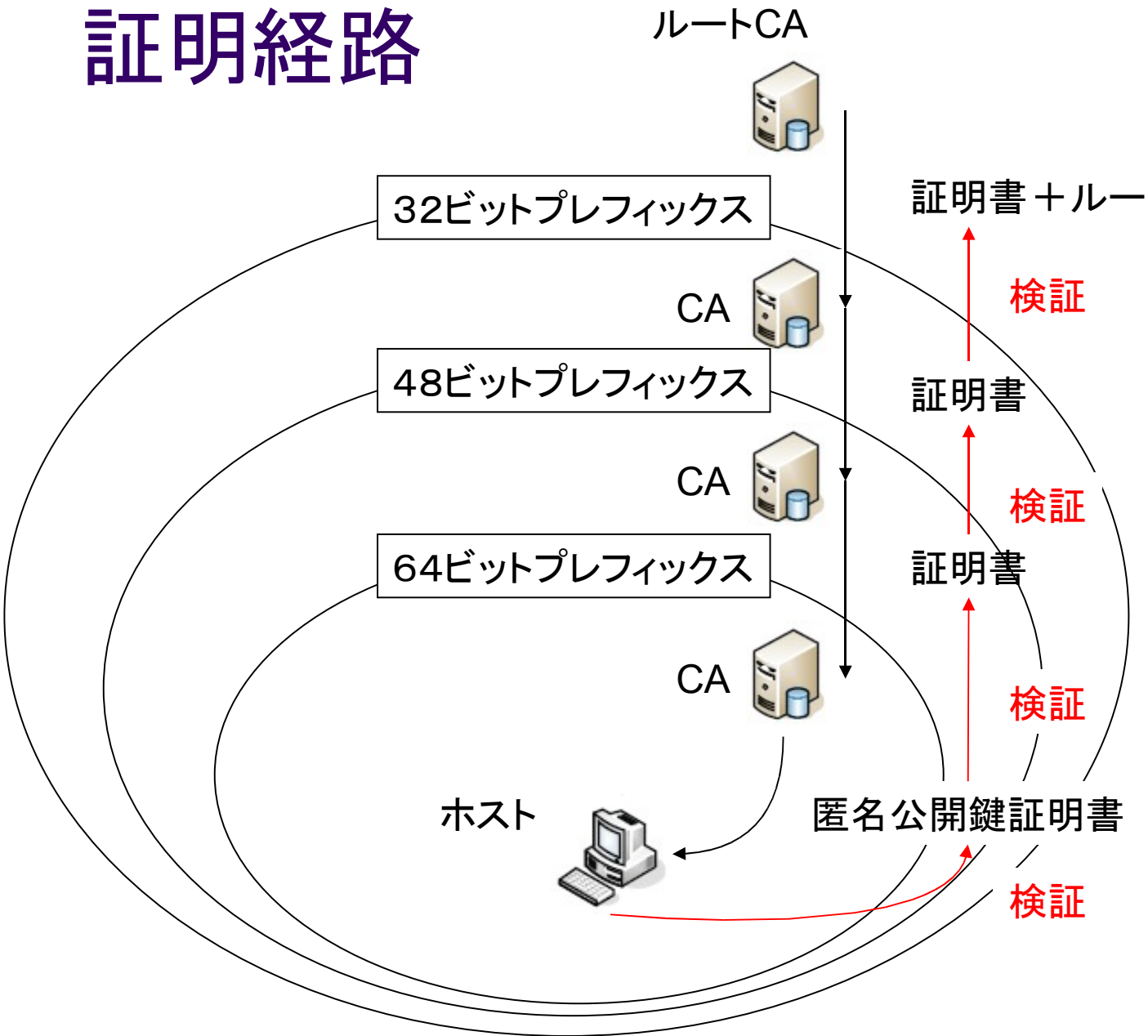
CAの展開

- IPv6アドレスはグローバルルーティングプレフィックスの割り当てによりアドレスが階層構造になる
- 同様にCAも階層構造で管理可能
- 上位のCAから下位のCAへ証明書を発行することでアドレス重複等の問題を検証できる証明経路が形成される

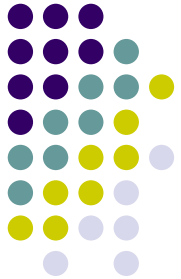
IPv6のグローバルユニキャストアドレスフォーマット

n bits	64-n bits	64 bits
global routing prefix	subnet ID	interface ID

証明経路


ルートCA
32ビットプレフィックス
CA
48ビットプレフィックス
CA
64ビットプレフィックス
CA
ホスト
証明書＋ルートCAの自己署名証明書
検証
証明書
検証
証明書
検証
匿名公開鍵証明書
検証

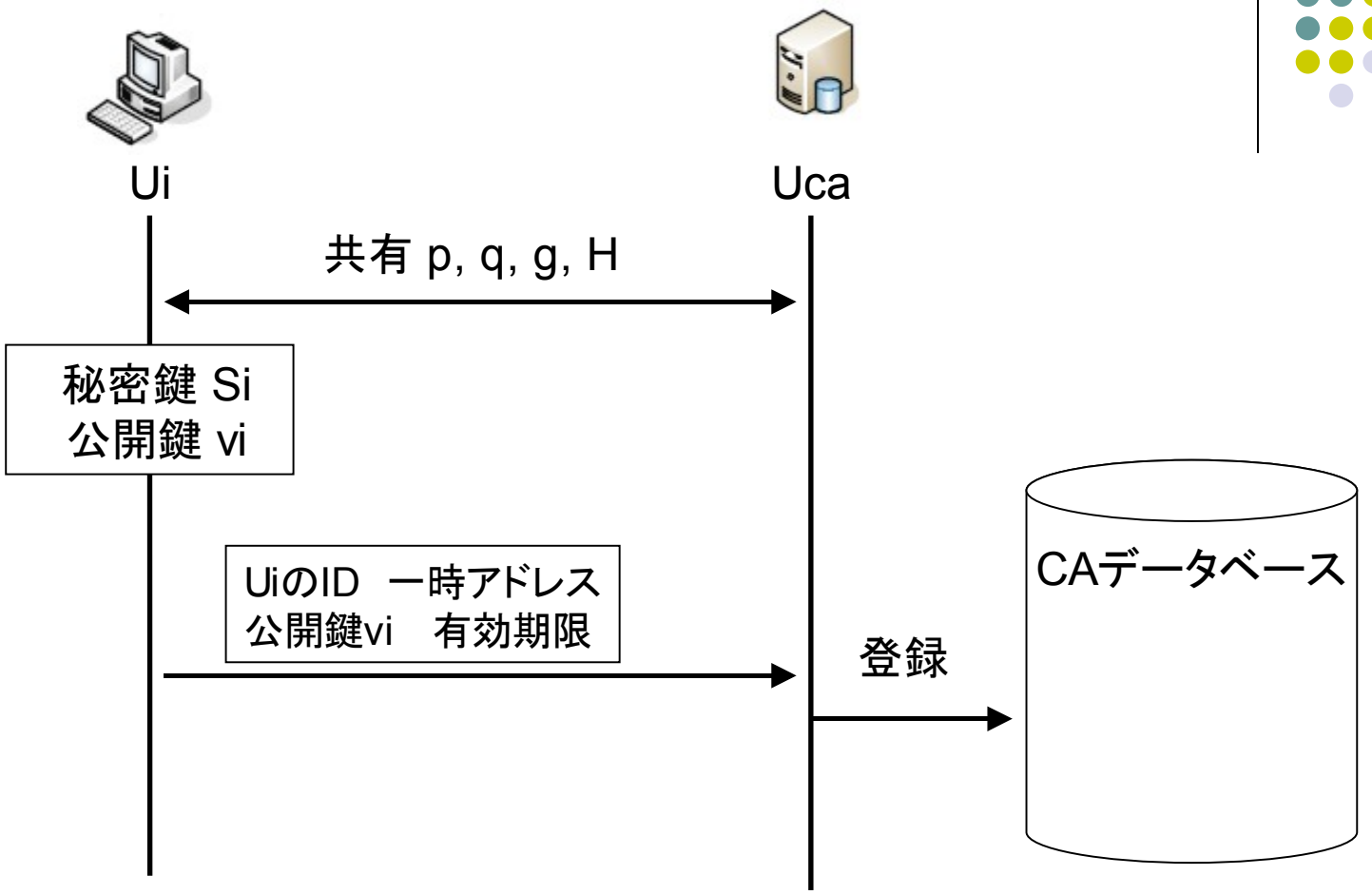
DSAによる匿名公開鍵証明書



- DSA (Digital Signature Algorithm)によって匿名公開鍵証明書の実装

- pを生成, qをq|p-1を満たす素数, gをqオーダのgenerator, Hをハッシュ関数とする
- p, q, g, HはDSAによってユーザとCAが共有

匿名公開鍵証明書


Ui
Uca
共有 p, q, g, H
秘密鍵 Si
公開鍵 vi
UiのID　一時アドレス
公開鍵vi　有効期限
登録
CAデータベース

匿名公開鍵証明書

Ui

Uca

共有データg, viからg', vi'を生成

UcaのID　一時アドレス
有効期限　p, q, g', vi'

X

署名sigca

検証

Ucaの公開鍵 ＋

匿名公開鍵証明書 (X, sigca)

メッセージ m

Uiの署名sig1, sig2を生成

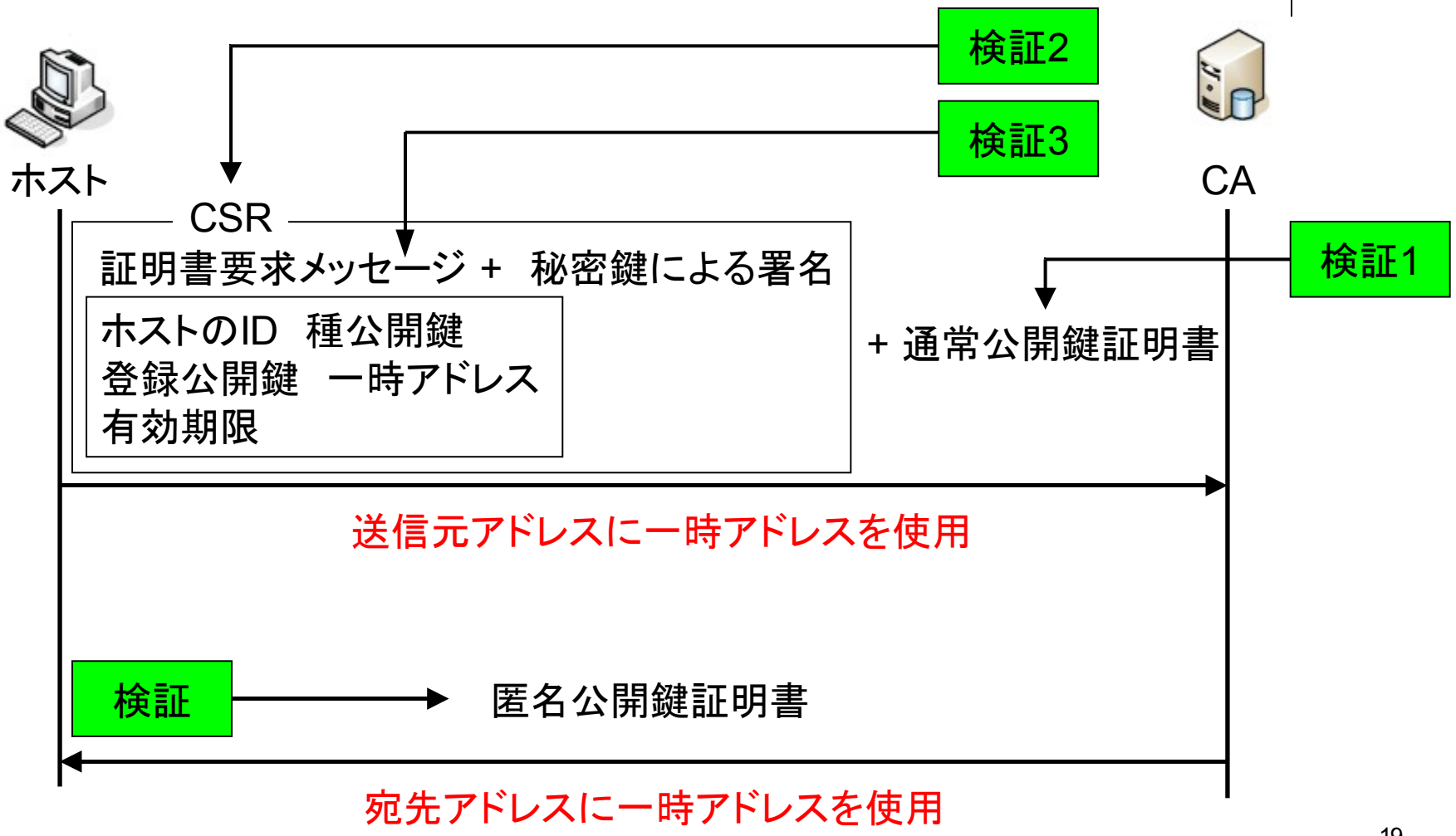
匿名署名 (X, sigca, m, sig1, sig2)

プロトコル動作1

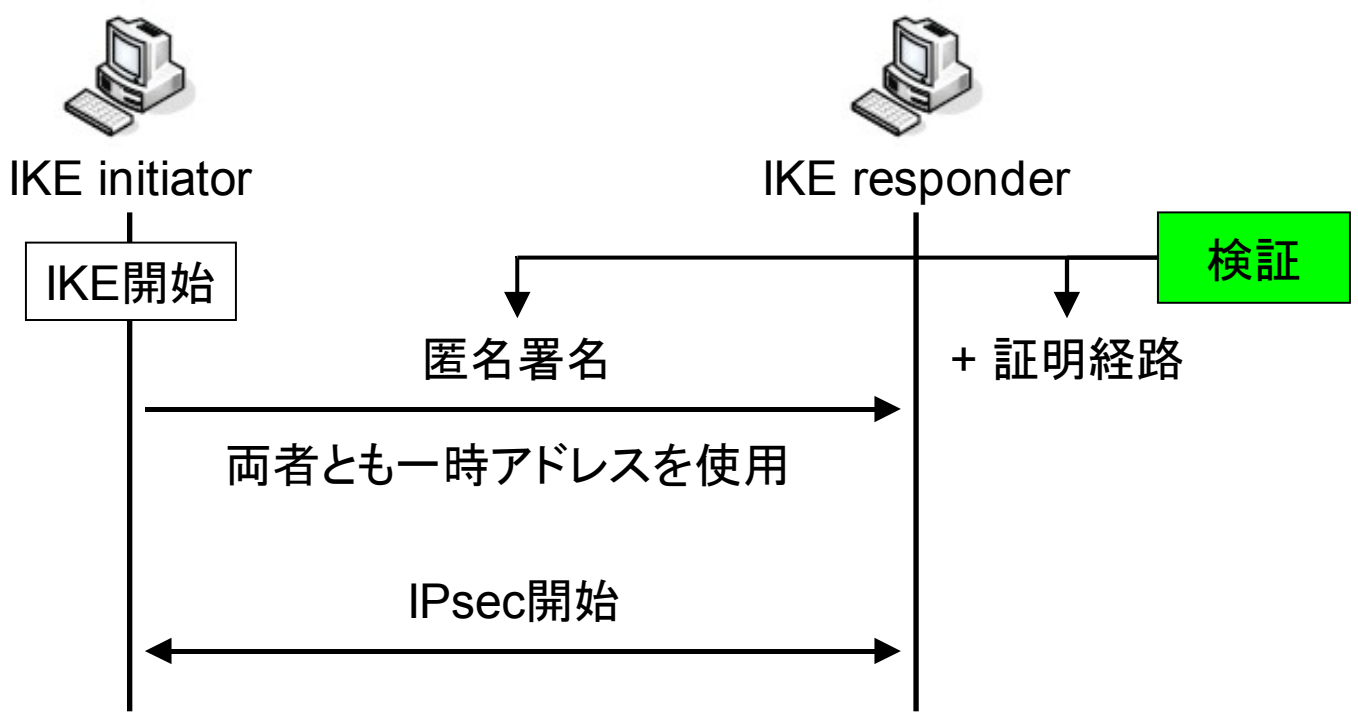
- 工場
  - ホストが秘密鍵と公開鍵を作成し，メーカにIDと公開鍵を提出
  - メーカのデータベースにIDと公開鍵を登録
    - 登録公開鍵
  - メーカから通常公開鍵証明書を発行

- ホスト設置時
  - 最初のみ，種公開鍵を作成
  - 一時アドレスを作成し，インタフェースに割り当てる

プロトコル動作2


検証2
検証3
ホスト
CA
CSR
証明書要求メッセージ＋　秘密鍵による署名
ホストのID　種公開鍵
登録公開鍵　一時アドレス
有効期限
検証1
＋通常公開鍵証明書
送信元アドレスに一時アドレスを使用
検証
匿名公開鍵証明書
宛先アドレスに一時アドレスを使用

プロトコル動作3

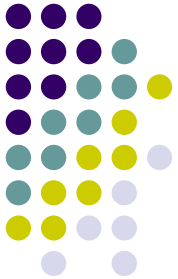


- IKEを開始するホストが一時アドレスでIKEを開始
- 匿名署名はDSAによって作成, IKEの応答側に送信
- IKE応答側は匿名署名内の匿名公開鍵証明書を受け取り, 匿名署名と証明経路を検証

プロトコル動作4

- 一時アドレスの取り消し期限になったら
  - ホストは新しい一時アドレスを作成し，プロトコル動作2，3を再度行う
  - CAはデータベースから一時アドレスを消去

プロトコルのセキュリティ

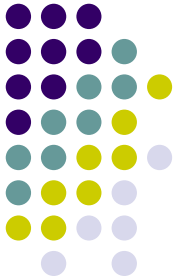


- 要求フェーズ
  - ディジタル署名が安全である限り偽造は不可能
  - CSRが暗号化されていなければ匿名性が失われる
- 発行フェーズ
  - DLPが安全である限り安全である
- IKEフェーズ
  - DLPが安全ならばIKEで使用されるDSAも安全
  - IKEの認証は中間者攻撃は防止できる

プロトタイプ

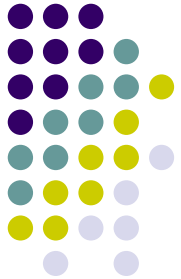
- ハードウェア　：IBM互換型PC
- OS　：FreeBSD 4.7-RELEASE
- IPv6&IPsec　：KAME SNAP
- IKE　：racoon
- 暗号化ライブラリ　：OpenSSL 0.9.6g
- 証明書フォーマット　：X.509v3
- 署名方式　：DSA

プロトタイプの仕様



- 仮定
  - IPv6デバイスは自身の秘密鍵と公開鍵，メーカが発行した公開鍵証明書を持っている
- 簡単化
  - IPv6デバイスのIDはメーカの公開鍵証明書によって保証される
- 拡張
  - ルータが自動的にホストにCAのアドレスを通知するようにRA,RSを拡張
  - プラグアンドプレイを実現するための拡張

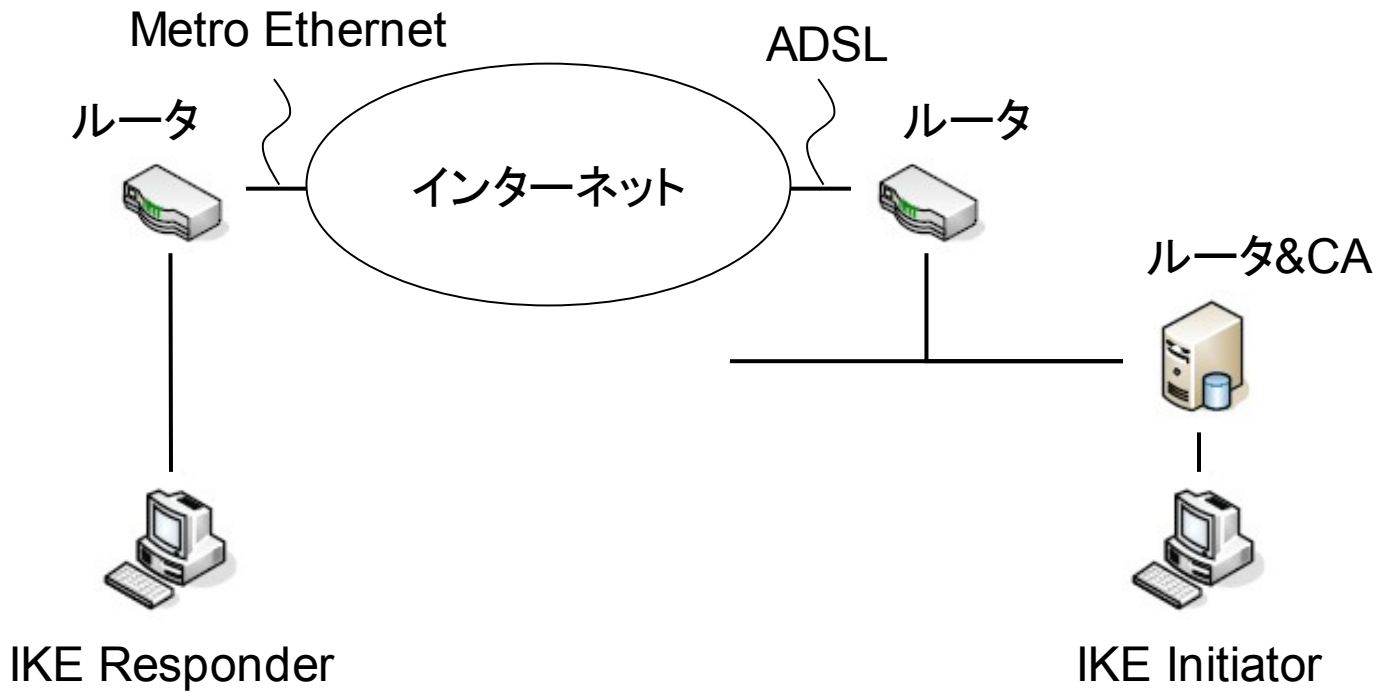
実装の設計問題



- racoonは複数のCAをサポートしないため，通常公開鍵証明書と匿名公開鍵証明書を発行するCAは1つ
- 登録公開鍵を種公開鍵として使用
- 転送プロトコルはTCPを使用
- CAの公開鍵によるCSRの暗号化は省略
- OpenSSL 0.9.6gがsubjectAltNameにIPv6アドレスを埋め込めるように修正

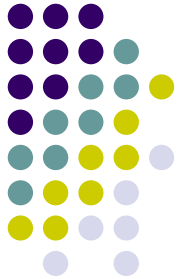
実験環境

- インターネットを挟んだ2つのサイト間でIKEとIPsecの通信



結果

- プロトタイプの動作を確認
- ホスト設置時に匿名公開鍵証明書の要求と受信を行い，IPv6一時アドレスと匿名公開鍵証明書を使用した通信の準備を自動的に行うことが可能
- プラグアンドプレイの特性に必要な特徴を備えている
- 上位層のping6, ssh, ネットワークカメラのキャノンオリジナルプロトコルを含めることを確認

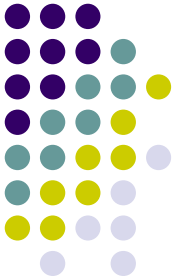


終わり

補足

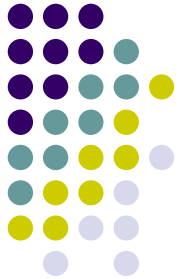
- IPv6ステートレスアドレス自動設定
  - グローバルアドレスを発生した後, DADによってネットワーク上で重複がないか確認する
  - 重複する場合はアドレスを破棄し, ステートフルアドレス自動設定を行う

Diffie-Hellman鍵交換方式



- generator g , prime p を両者が受け取る
  - 第三者に知られてもよい
- hA : 乱数 x を生成, 配送鍵 A=g^x mod p を生成し, 配送
  - 第三者に知られてもよい
- hB : 乱数 y を生成, 配送鍵 B=g^y mod p を生成し, 配送
  - 第三者に知られてもよい
- hA : B^x mod p を計算
- hB : A^y mod p を計算
- 計算結果が同じになり, 秘密鍵を共有可能
- 安全性
  - 共有鍵 K = B^x mod p = A^y mod p = g^xy mod p
  - 第三者が知っているのはA, B, g, p, 未知数はx, y
  - 値が等しくなるx, yの組み合わせを見つけるのは困難
  - p の値が大きいほど計算は困難になる(x, yの組み合わせが増加)

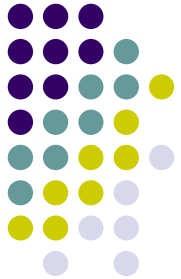
匿名公開鍵証明書の検証



- Ucaの公開鍵によってsigcaを検証

- 検証結果が正当だと判断されれば，メッセージの匿名署名がUcaによって認証された匿名エンティティが作成したものであることが実証される

プログラム



- 4つのプログラムを実装
  - rtsold, rtadvd
    - ルータからホストへCAのアドレスを伝えるため, KAME SNAPのオリジナルデーモンtsold, tadvdを拡張
  - apcreqd
    - CAに匿名公開鍵証明書を要求するためにホスト側で実行
  - apcresd
    - ホストに匿名公開鍵証明書を発行するためにCA側で実行