CAを応用したIPv6鍵管理プロトコル強化の提案

6F－7

島 成佳　　櫻井 三子　　石井 秀治

NEC ネットワーキング技術研究所

1. はじめに

現在IETFのIPsecワーキンググループでは，安全な通信を行うためにIPレベルでの認証，暗号化の枠組みを規定している．IPsecでは，データの正当性を保証するAH（認証ヘッダ）とデータを暗号化するESP（暗号ペイロード）によって安全な通信を保証している．AHやESPを使用するためには，ホストがお互いに鍵，暗号アルゴリズム，認証方法などについて合意しなければらない．そのためIPv6鍵管理プロトコル内で，それらの合意を自動的に行なう鍵交換プロトコルが提案された．

現在鍵交換プロトコルは，Diffie-Hellman 鍵交換プロトコルを基本としたISAKMP/Oakleyが必須となっている．しかし，ISAKMP/Oakleyでは，Diffie-Hellman 公開鍵とその所有者であるホストの対応が正しいことを証明する仕組みがない．また，Diffie-Hellman 公開鍵は鍵交換専用で，RSAのように署名や暗号化に利用することはできない．そのためISAKMP/Oakleyで相手を認証しようとする場合には，Diffie-Hellmanと別の公開鍵暗号とを併用しなければならずプロトコルが複雑になる．そこで本論文では，以下の特徴を持つ簡単な鍵交換プロトコルを提案する．

- Diffie-Hellman 以外の一般的な公開鍵暗号を利用
- 公開鍵とその所有者であるホストが正しいことをCA(公開鍵証明書発行局)が保証
- 相手の公開鍵を証明書形式でsecure DNSから入手

2. 公開鍵の保証

2.1 保証されていない公開鍵の危険性

ホストとホストが生成した公開鍵の関係が証明されていない場合，以下のような問題が発生すると予想される．

- 鍵の偽造
  悪意を持った者が公開鍵暗号の鍵のペアーを生成し，その公開鍵をホストAのものとして配布することで，中間一致攻撃に利用される恐れもある．
- 署名の否認
  ホストA(またはユーザ)がデータ1に署名したのち，ホストAが署名に使用されている秘密鍵は自分のものではないと主張することで，ホストAはデータ1への署名を否定できる．

以上から，ホストと公開鍵の関係を示す証明書が必要になると考えられる．

2.2 公開鍵証明書

公開鍵証明書の内容は，公開鍵，暗号アルゴリズム，鍵の生成者，有効期限，証明書の発行機関などと，それらを発行機関が秘密鍵により署名したものからなる[1]．公開鍵証明書の正当性は，発行機関の公開鍵により，証明書の署名を検証することで確認できる．

2.3 CAとsecureDNS

CAは，ホストとホストの公開鍵との対応が正しいことを証明するために，証明書の発行，配布，検証，廃棄を行なう．CA間の構成は信頼できる最上位CAを頂点としてツリー状になっており，上位CAが下位CAを保証することですべてのCAが保証されている．このとき上位CAは下位CAが決められたセキュリティ基準をみたさなければ登録を行なわない．またCAは証明書を発行する際のホストのチェックをオフラインで厳しく行なう．

secure DNSはホスト名やIPアドレスを受け取り，公開鍵または公開鍵証明書，CRL(廃棄証明書リスト)を返す機能を持つ．secure DNSの構成は，secure DNSサーバー間が互に信頼することを前提として構成されている，誰かが各secure DNSを安全であると保証しているわけではない．これを利用するとインターネット上にある任意のホストの証明書のみを入手することが可能である．

3. 提案方式

3.1 全体の通信の流れ

ここではまず公開鍵の入手からIPsecの通信が始まるまでの流れを図1に沿って説明する．

1) secure DNSによって公開鍵を入手する(図1の1)．

Design of Enhanced IPv6 Key Management Protocol with Certification Authority

Shigeyoshi Shima, Mine Sakurai, Shuji Ishii
NEC Corporation

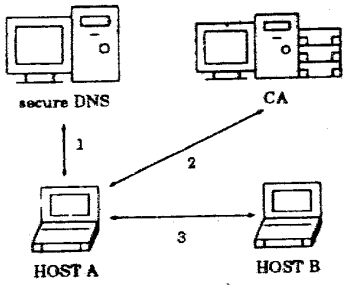

図1：提案方式の全体図

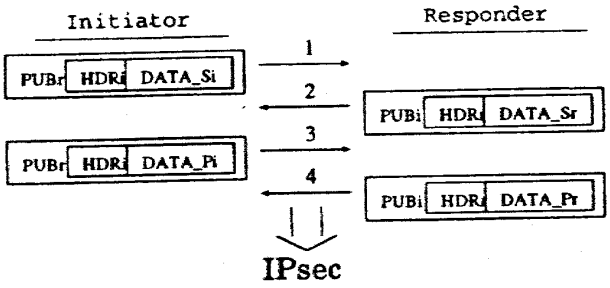

図2：ホスト間の鍵交換プロトコル

2) 証明書を検証したい場合は，ホストがその証明書を発行したCAに対して検証を依頼し，検証結果を得る(図1の2).
3) 提案する鍵交換プロトコルにより，ホスト間でAHやESPのセキュリティ・アソシエーションについて合意する(図1の3).

次に本提案方式の特徴について以下で述べる.

1)では，secure DNSを証明書配布機能として利用することで，通信相手の証明書を探すためのコストが小さくなる．また，secure DNSとホストの通信は[2]で定義されたものをそのまま利用できる.

2)では，証明書を発行したCAに証明書の検証を直接依頼する．それはホストの証明書とCAの証明書が偽造されてsecure DNSに置かれるのを防ぐためである．ホストが各CAの証明書を正しく入手できる場合は，その証明書でホストの証明書を検証してもよい(この場合2)は省略できる).

3)では，Diffie-Hellman鍵交換方式とは違い，通信開始時から2)で保証された公開鍵を使った暗号通信を行って，セキュリティ・アソシエーションの合意を簡単に行うことができ，また送信されてきたデータの検証についても結果が保証される.

3.2 鍵交換プロトコル

提案する鍵交換プロトコルは，図2に示すように4回のデータ通信からなる．前半2回では，後半2回の通信で利用する暗号アルゴリズム，ハッシュ関数，認証方式などの合意を行い，後半2回でIPsecで使用されるデータの交換を行う.

次に図2のプロトコルについて説明を行なう.

1) はじめに始動者は，メッセージIDと長さを示すHDR(ヘッダ)とDATA_S(暗号通信を行なうための合意データとそれらを始動者の秘密鍵で署名したもの)をPUBr(応答者の公開鍵)によって暗号化し，応答者に送る(図2の1).
2) 応答者は送られてきたデータを自分の秘密鍵で復号し，DATA_S内の署名内容を検証する．(始動者の公開鍵がない場合，secure DNSから入手する)次に応答者は始動者と同様にHDRとDATA_SをPUBi(始動者の公開鍵)によって暗号化し，応答者に送る(図2の2).
3) 始動者は応答者からデータを復号し，署名の内容を検証する．次に始動者はHDRとDATA_P(IPsecで使用するためのデータ)を署名したものにPUBrで暗号化し，応答者に送信する(図2の3).
4) 応答者も同様にDATA_Pを始動者に送信する(図2の4).
5) 始動者と応答者で合意ができた後にIPsecでの通信を行なう.

なお，具体的なデータフォーマットについては，ISAKMP[3]の枠組みを応用し，ISAKMPで定義されたヘッダとデータ構造を運ぶためのいくつかのペイロードを組み合わせて構成する．図2のHDRはISAKMPヘッダ，DATA_Sはセキュリティ・アソシエーションペイロード，DATA_Pはプロポーザルペイロードに対応する．提案方式でISAKMPを採用した理由としてネットワーク上のサービス妨害攻撃や再送攻撃を防止する仕組みがあるからである.

4. おわりに

本論文ではCAとsecure DNSを利用し，鍵交換プロトコルで問題となっていた公開鍵や公開鍵証明書の入手と証明書の検証について提案した．また，Diffie-Hellman以外の公開鍵暗号を利用した簡単な鍵交換プロトコルを提案した．今後はさらに詳細設計を進め実装を行う予定である.

参考文献

[1] CCITT: "The Directory – Authentication Framework", CCITT Recommendation X.509, Nov. 1988.
[2] D.Eastlake, 3rd, C. Kaufman "Domain Name System Security Extensions", Jaunary. 1997. RFC2065
[3] Mark Schneider, Jeff Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", Jul. 1997. draft-ietf-ipsec-isakmp-08