IPsecにおける証明書管理方式

5H−02

乗松 淳一† 若山 公威† 村瀬 晋二‡ 鈴木 春洋‡ 岩田 彰†
†名古屋工業大学 電気情報工学科
‡株式会社シーティーアイ 先端IT開発部

1. はじめに

近年、インターネットは凄まじい勢いで普及し、様々な目的で使われるようになってきた。それに伴い、個人データなどを通信する際には、セキュアな通信路の必要性が問われている。現在、最も良く知られ、使われている暗号技術はSSL (Secure Sockets Layer) [1] だろう。しかし、セッション層での実装は、必ずしも良いとは言い切れない。そこで、IPv6にて実装必須となっている、IPsec (IP Security) [2] と呼ばれる暗号技術がある。これは、ネットワーク層での実装を実現しており、IPv6の普及と共に、一般的に使われるようになることが期待される。そのIPsecの機能として、認証、暗号化による通信を行うことができるが、その際使用する鍵は、IPsecの安全性を高めるために、頻繁に交換することが望まれている。しかし、手動で交換することの困難性から、IKE (Internet Key Exchange) [3] といわれる自動鍵交換が使われている。今回はこのIKEに焦点を合わせて、その有効性を検証していく。

2. IKE

IKEは、IPsecにおいて、セキュアな通信路を確立するための、SA (Security Association) のパラメータを自動交換することができる。その際、2つのフェーズを必要とし、フェーズ1においてセキュアな通信路を確立し、フェーズ2において、IPsecに必要なパラメータの交換を行う。これによりフェーズ2で交換されたパラメータの機密性を保っている。この一連の動作において、フェーズ1の認証は重要なものといえる。その認証方式として、現在4つのものが定義されている。

1. 事前共有秘密鍵認証方式
2. デジタル署名認証方式
3. 公開鍵暗号化認証方式
4. 改良型公開鍵暗号化認証方式

様々な認証方式がある中、比較の目安として、どのような事前処理が必要かが検討されるだろう。今回使用するデジタル署名認証方式では、相手に特化した事前処理を必要とせず、自身の秘密鍵と公開鍵証明書を保持していればよい。

ここで、デジタル署名認証方式についてだが、フェーズ1での各IKEノードの通信手順は、図1のように示される。

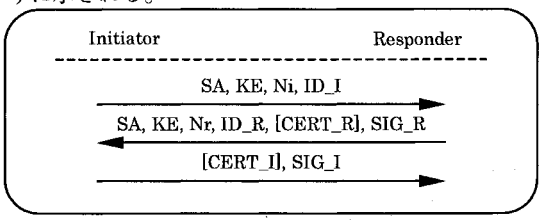

図1 デジタル署名認証方式
(アグレッシブモード)

3. 問題点と解決策

前節で述べた、デジタル署名認証方式において考える。この方式の利点とも思われる証明書を使用することだが、証明書の特性より、いくつかの問題点がある。以下に証明書を扱う場合の問題点をあげる。

- ユーザが証明書を正しく扱うには、証明書に関する知識が必要である
- 証明書更新などの際、各IKEノードに証明書を配布するのは手間がかかる

Implementation of Certificate Management Mechanism for IPsec
Junichi Norimatsu†, Kimitake Wakayama†,
Shinji Murase‡, Shunyo Suzuki‡, Akira Iwata†
†Nagoya Institute of Technology
‡CTI Co., Ltd.

個々のユーザが証明書を管理するのは困難である。これは、証明書には有効期限が設定され、期限切れなどをユーザが管理しなければならないことが原因である。有効期限が切れるたびに証明書を更新し、インポートするのは手間がかかる上、証明書の知識のない人には困難な作業だろう。

そこで解決策として、証明書管理サーバを設置し、ユーザに代わって証明書を通信相手に送信するシステムを提案し、その開発を行った。各 IKE ノードの証明書管理コストを削減するこのシステムは、証明書を普段扱うことのないユーザには便利なものだと言える。それは、自身の証明書が有効なものかを、証明書管理サーバに任せてあるからだ。このシステムによって、ユーザは自身の証明書を扱う煩わしさから解放される。

4. システム概要

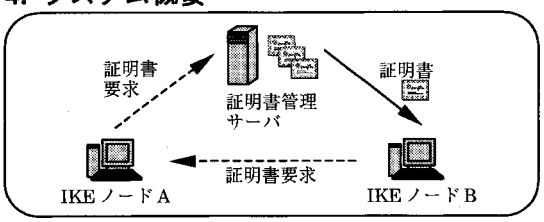

図 2 システム構成

今回提案したシステムの構成を図 2 に示す。これより、IKE の手順に合わせて説明を行う。まず、IPsec 通信相手となる IKE ノード B から、IKE ノード A へ証明書要求が送られる。しかし、IKE ノード A では自身の証明書を保持していないので、証明書管理サーバに証明書要求を送信する。要求を受けた証明書管理サーバは、IKE ノード A に該当する証明書を IKE ノード B に送信する。これにより IKE ノード B は IKE ノード A の証明書を手に入れ、その後の処理が問題なく続行される。なお、IKE ノード A の秘密鍵については、自身が保持している。

もう一つ問題となっている証明書更新などの際の配布処理についてだが、CA (Certification Authority) によって発行された証明書は、通常は各 IKE ノードに配布しなけらばならないため、IKE ノードの数だけ配布処理をしなければならない。しかし本システムでは、各 IKE ノードの証明書は証明書管理サーバが集中管理しているので、サーバに配布するだけでよい。

5. 実装

証明書管理サーバの開発には、OpenSSL0.9.6a [4] を使用した。動作確認に用いた IKE ノードには、同じく OpenSSL0.9.6a [4]と、FreeBSD4.4 で IPsec を実装している kame-20011224-freebsd44-snap [5]、更に IKE デーモンとして racoon-20011215a [5]を使用した。今回の動作確認のため、IKE デーモンプログラムに一部改良を行い、証明書管理サーバを経由するオプションを追加した。また、証明書管理サーバでは、その要求を満たすデーモンプログラムを実装している。本システムで、問題なく IKE の動作が行われることを確認した。

6. おわりに

今回提案したシステムにより、IKE ノードでの証明書管理負担を削減した。その結果、IKE に証明書が必要なときも、証明書管理サーバが自動的に要求に答えるので、ユーザは自身の証明書の有効期限を気にすることなく、IPsec 通信を行うことができる。また、証明書管理サーバの管理者は、CA によって発行された証明書を一括管理すればいいので、各 IKE ノードに証明書を配布する手間を省くことができる。

今後は、相手の証明書の有効期限、失効情報の確認や、証明書管理サーバによる証明書キャッシュなどについて検討したい。

参考文献

[1] Internet-Draft "The SSL Protocol Version 3.0"
Alan O. Freier, Philip Karlton, Paul C. Kocher, November 1996.
[2] RFC2401 "Security Architecture for the Internet Protocol"
S.Kent, R. Atkinson, November 1998.
[3] RFC2409 "The Internet Key Exchange"
D. Harkins, D. Carrel, November 1998.
[4] http://www.openssl.org
[5] http://www.kame.net