暗号化フィルタによるセキュア通信の実現と実装*

5N－4

横田 智文　安井 浩之　松山 実†
武蔵工業大学‡

1 はじめに

現在のインターネットはセキュリティ機能が備わっていないため，利用者自身が通信データを盗聴や改竄から守るために暗号化などを行う必要がある．アプリケーションごとに暗号化するソフトウェアは存在するが，汎用的に利用可能なソフトウェアは存在しない．そこで，インターネットアプリケーションに依存しない汎用的な暗号化フィルタを開発し，より安全な通信を簡便に行うことを試みた[1)]．ここでは，暗号化フィルタの実装方法について報告する．

2 セキュアコミュニケーションプロトコル

セキュアコミュニケーションプロトコル(Secure Communication Protocol．以下 SCP)は，ここで開発した暗号化フィルタシステムの名称で，ソフトウェアにより開発した．ここで暗号化の対象にしたサービスは，FTP，TELNET，HTTP，POP の４種とした．SMTP を対象に入れなかった理由は，上記４種はデータがサーバから直接送られてくるタイプのプロトコルに対し，SMTP は，データがいくつかのサーバに中継されて送られるタイプのプロトコルである為である．

通常のサービスを利用する場合(図１)，サーバとクライアントは直接通信するが，SCP を利用した場合(図２)は，サーバ上の SCP Server とクライアント上の SCP Client が通信する．現在のインターネットアプリケーションは，ほとんどセキュリティ機能が備わっていないので，SCP Server と SCP Client 間でセキュリティを確保することで，セキュリティ機能を持っていないインターネットアプリケーションでも，安全な通信が行えるようになる．

さらに，SCP を実装していないコンピュータとでも通信を行えるようにするため，ハンドシェイク機能を組み込んでいる．

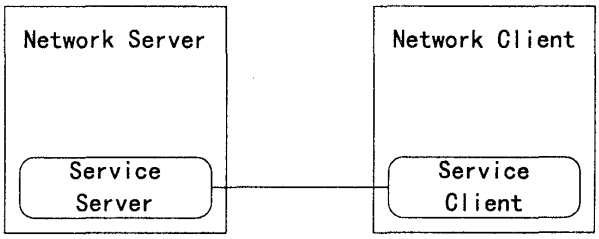

図 1. 一般の通信イメージ

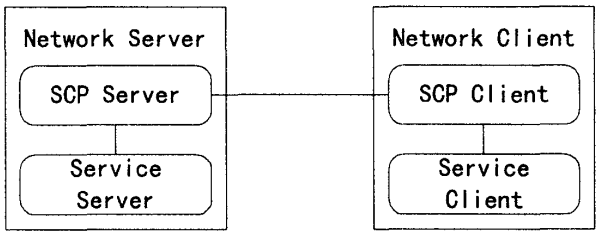

図 2. SCP の通信イメージ

3 実装

ここでは Sun Solaris 2.6，SONY NEWS-OS 4.2RD，Linux 2.0.35(Slackware3.5)で開発および実装を行い，前述の４つのサービスについて，暗号化通信が行えることを実験的に確認した．

3.1 通信遅延の測定実験

SCP はソフトウェアであるため，ハードウェアによる暗号化よりも通信遅延が大きいことが予測される．そこで SCP を利用した場合と利用しない場合の通信遅延を実験的に測定し，速度比較を行った．実験環境は，サーバ／クライアントともに UNIX(サーバ CPU：MIPS R4000 100MHz，クライアント CPU：Intel Pentium 120MHz)，通信速度は 64Kbps(ISDN)と 10Mbps(Ethernet)の２種，転送ファイルサイズは，4.7MB(10Mbps)と 0.6MB(64Kbps)，使用したプロトコルは FTP(ダ

---

* Equipment of Secure Communication with Encryption Filter

† Tomofumi Yokota(E-mail: yokota@lb.ipc.musashi-tech.ac.jp)，Hiroyuki Yasui，Minoru Matsuyama
‡ Musashi Institute of Technology

ウンロード)である.

10Mbps のネットワーク環境は，インターネットから切り離された LAN，64Kbps のネットワーク環境は，既存プロバイダの回線(インターネット網)を使用し測定した．各測定は項目ごとに 15 回行い，そのうちの上位 5 回の平均値を表 1 に示す.

表 1. 通信遅延の測定

通信速度	SCP	転送時間	通信レート
10Mbps	あり	約 22.74 秒	206KB/秒
10Mbps	なし	約 4.34 秒	1079KB/秒
64Kbps	あり	約 85.76 秒	6.86KB/秒
64Kbps	なし	約 84.88 秒	6.93KB/秒

3.2 考察

通信速度が 10Mbps の場合は，SCP を利用しない場合と比べて 19%の速度比となったが，64Kbps の場合は，SCP を利用しない場合に比べても 99%の速度比が得られた．高速なネットワークを使用する場合は相対的に遅延が大きくなるが，これは暗号アルゴリズムの変更などによって改善されることが期待できる.

4 SCP の利用について

4.1 SCP サーバと SCP クライアントの設定

SCP は UNIX のデーモンとして実装しており，サーバモードかクライアントモードを指定してバックグラウンドプロセスとして起動する．さらに以下に示すような引数を指定する必要がある.

```
#scpd <Mode> <Host> <Port> <SPort> <Protocol>
Mode : -s(サーバ)または-c(クライアント)
Host : 接続するサーバ名
Port : 接続するサーバのポート番号
SPort : 待ち受け SCP ポート番号
Protocol : プロトコル名<ftp, telnet, http, pop>
```

図 2 のように SCP サーバは，自身のサービスサーバのみに接続するため(接続先が 1 つ)，1 回の設定で済む．またサービスクライアントも，自身の SCP クライアントのみに接続するため(接続先が 1 つ)，1 回の設定で済む．一方，SCP クライアントは，複数の SCP サーバに接続する可能性がある.

4.2 SCP 設定の手間の軽減

SCP サーバが 1 台しか存在しない場合は，SCP クライアントも 1 回の設定で済む．しかし SCP サーバが複数存在する場合は，SCP クライアントは接続するたびに設定を変えなくてはならない．そこで，SCP クライアントを Proxy として動作させることを検討した.

Proxy として最も一般的である HTTP について実装を行った．WWW ブラウザから送られてくるメッセージから，接続先サーバの情報を SCP クライアントが理解することにより，この機能を実現した．動作確認は，WWW ブラウザの Proxy に SCP クライアントを登録し，Netscape Navigator，Netscape Communicator，Internet Explorer から複数の SCP サーバに接続できた.

また前述したハンドシェイク機能を改良することにより，Proxy に SCP クライアントが設定されている WWW ブラウザからでも，SCP サーバ以外の HTTP サーバに対して，通常の接続・通信ができることを確認した.

5 おわりに

アプリケーションに依存しない暗号化フィルタを実装することで，使用中のインターネットアプリケーションに変更を加えることなく安全な通信が実現できることが確認できた．また SCP クライアントを Proxy として動作させることにより，クライアントの設定変更の手間を軽減することができた.

現在，SCP クライアントを Windows 系 OS に移植中である.

---

[1] 横田智文 他：暗号フィルタによるセキュア通信の実現，情報処理学会，第 57 回全国大会，講演論文集(3),pp. 512-513,1998