暗号化フィルタによるセキュア通信の実現*

6G−3

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，何らかの方法で利用者側が通信データを盗聴や改竄から守る必要がある．

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．

2 データ暗号化システム

2.1 ハードウェアによる暗号化

ハードウェアを利用して暗号化する場合の長所は，導入が比較的簡単である，導入による通信遅延が少ない，ことなどが挙げられる．一方，短所は，高価である，一般に同種のハードウェア間での通信に限定されるため，利用可能な場所に制限を受ける，ことなどが挙げられる．企業の本社－支社間，本社－関連会社など，限られた範囲内での利用価値は高いが，一般利用者や公衆回線からの利用などでは非現実的である．

2.2 ソフトウェアによる暗号化

ソフトウェアを利用して暗号化する場合の長所・短所は，ハードウェアを利用する場合とほぼ逆になる．さらに短所として，アプリケーションごとの暗号化システムは存在するが，１つの暗号化システムで全てのアプリケーションに対応するのが難しいといったことが挙げられる．

3 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)とは，ここで開発したシステムの名称である．SCP は，ハードウェアによる暗号化の短所である利用可能な場所に制限を受けるという点と，ソフトウェアによる暗号化の短所である１つの暗号化システムで全てのアプリケーションに対応するのが難しいという点を解決するため，ソフトウェアにより開発した．

3.1 SCP の開発条件

SCP の開発条件は，

(1)稼動しているサーバ及びクライアントに変更を加えることなく利用できる

(2)必要最小限の設定で済むようにする

(3)多くのサービスに対応できるようにする

とした．この条件を満たす方法として OS レベルで実装する方法が挙げられるが，SCP の簡便な導入という観点からOSには変更を加えないことにした．

ここで対象にしたサービスは，利用頻度が高いという理由から，FTP，TELNET，HTTP，POP の 4 種にした．ここで SMTP を対象に入れなかったのは，プロトコルの性質が他の 4 種と異なるためである．また今回対象としたサーバ／クライアントは，ともに UNIX ワークステーションである．

3.2 通信イメージ

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．そこで 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，セキュア通信が行えるようにした．

また，SCP を実装しているコンピュータと，そ

---

* Secure Communication with Encryption Filter

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

‡ Musashi Institute of Technology

うでないコンピュータが通信を行えるようにするため，ハンドシェイクプロトコルを組み込んだ．

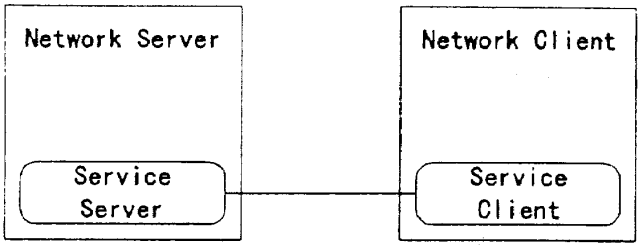

図1. 一般の通信イメージ

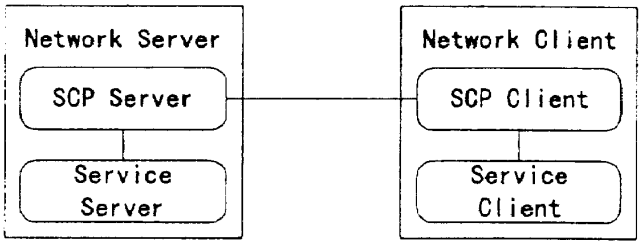

図2. SCP の通信イメージ

3.3 ハンドシェイクプロトコルの組み込み

SCP で使用する暗号アルゴリズムは，対称鍵暗号と非対称鍵暗号[1]の２つである．これはそれぞれの暗号方式の特徴を生かし，暗号化による通信遅延を極力減少させるためである．対称鍵暗号の暗号化は一般に非対称鍵暗号の約 100 倍の速度[2]で行えるため，対称鍵暗号をデータ本体の暗号化に用い，その鍵を非対称鍵暗号で暗号化する．またここで使用する対称鍵暗号の鍵と非対称鍵暗号の鍵は，１セッションの間でのみ有効な鍵をハンドシェイクの中で自動生成する（図 3）．今回 SCP では，対称鍵暗号として DES，非対称鍵暗号として RSA を使用した．

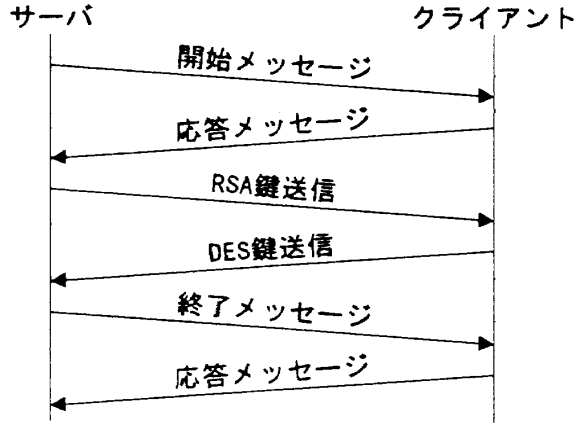

図3. ハンドシェイクプロトコル

4 実装

ここでは SONY NEWS-OS 4.2RD および Linux 2.0.30（Slackware 3.4）で開発および実装を行った．開発条件の中に記述した４つのサービスに対して，暗号化通信が行えることを実験的に確認した．

4.1 通信遅延

SCP はソフトウェアであるため，暗号化の際に通信遅延が起きる．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測定した．通信速度は 10Mbps（Ethernet）と 28.8Kbps（Modem）の２種，使用したプロトコルは HTTP，転送に使用したファイルサイズは 1MB である．

表1. 通信遅延

通信速度	SCP	転送時間	通信レート
10Mbps	あり	約 100 秒	10KB/秒
10Mbps	なし	約 3.5 秒	280KB/秒
28.8Kbps	あり	約 330 秒	3KB/秒
28.8Kbps	なし	約 250 秒	4KB/秒

4.2 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，わずか 4%の通信レートに落ちたが，28.8Kbps の場合は，SCP を利用しない場合に比べても，75%の通信レートを保った．

通信速度が高速な場合は，通信遅延が大きくなるが，これは暗号アルゴリズムの見直しによって，大幅に改善されることが期待できる．

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．今後は，SCP を Windows 系 OS への移植を検討する予定である．

---

[1] 稲村 雄：Open Design No.14 最新の暗号技術によるセキュリティの実現，ＣＱ出版社，（1996）

[2] すずきひろのぶ：Software Design 11 月号，技術評論社，pp.29-35（1996）