モーバイル環境に適した圧縮／暗号化通信方式（その1）　パケット圧縮／暗号化通信方式

3W－5

高橋　泰弘*　松井　進*　中田幸男*　近藤　毅*　大津　豊**
*（株）日立製作所システム開発研究所　**（株）日立製作所ソフトウェア開発本部

1．はじめに

モーバイル環境、特に無線では、有線に比べ、通信速度が遅く、盗聴の危険もあるため、通信データに対して、圧縮および暗号化を行い、効率的かつ、セキュアな通信を実現する必要がある。

そこで、TCP/IPベースの既存のアプリケーション（以下、AP）をそのまま使い、WinSockのAPIにおいて、通信データに対するEnd to Endの圧縮、暗号化を行う通信方式を検討した。

2．通信路データ圧縮暗号化の問題点

通信路データの圧縮暗号化においては、1）APレベルで暗号化し、モデムにて圧縮する方法、2）通信制御部分に圧縮暗号化機能を組み込む方法、があげられる。しかしながら、1）では暗号化によって、データがランダム化されてしまい、その後の圧縮では、圧縮性能が上がらない。また、暗号化処理をAPに組み込まなくてならない。2）では、TCP/IP自身を改造しなくてはならず、前提条件を満たさない。また、従来からあるPPPにおける圧縮暗号化では、アクセスサーバまでの区間のみが対象となり、End to endの条件を満たさない。

3．WinSock・API取り込みによる圧縮暗号化

TCP/IPベースのPC用通信APにおいては、ソケットインタフェースとして標準であるWinSock・APIが用いられている。そこで、このAPIに対する命令を取り込み、独自の処理を加えることにより、圧縮暗号化機能組み込みを実現した。すなわち、APとWinSockとの間のリンクを実行時に変更し、APからのsend命令を、今回開発したセキュア通信付加機構に一度取り込み、圧縮／暗号化処理を加えてから、実際のTCP/IPに渡す。また、receive命令では、TCP/IPからの受信データをセキュア通信付加機構にて復号／伸長してから、APに返すようにした。これにより、APおよびTCP/IPソフトを改造することなく、圧縮／暗号化を組み込むことができた。

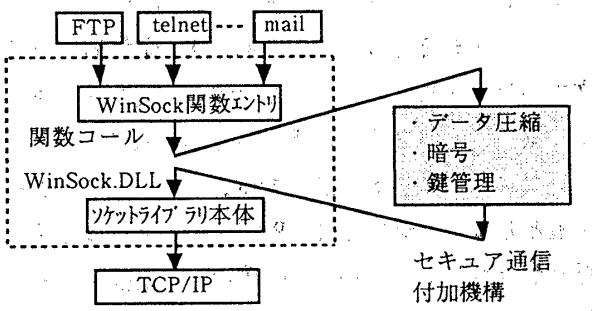

図1　WinSock API取り込み

4．ネゴシエーション機能の組み込み

圧縮暗号化機能をもつPCと、これらの機能を持たないPCが共存するためにはネゴシエーション機能が必要である。しかしながら、TCP/IPプロトコルを変更することはできないため、コネクション設定処理の中で、通信相手の機能判別、圧縮方式、暗号化方式の選択、暗号鍵の確認等を含めることにした。すなわち、APからのconnect命令をセキュア通信付加機構にて取り込み、TCP/IPに対してconnect命令を出し直す際、あらかじめ決めておいたセキュアポート番号に置き換えて送信する。これに対する

Communication Method with Data Compression and Encryption for Mobile Computing Environment (1)

Yasuhiro Takahashi*, Susumu Matsui*, Yukio Nakata*, Takeshi Kondo*, Yutaka Ootsu**
*Systems Development Laboratory, Hitachi, Ltd.　**Software Development Center, Hitachi, Ltd.

リターンの有無でセキュア通信可能か否か判断する。可能な場合、相手のacceptに対するリターンをそのまま返すのではなく、引き続き、通信相手のセキュア通信付加機構とで、圧縮暗号化のネゴシエーションを行い、これが完了したところで、APに対してリターン値を返す。接続された側でも、APからのaccept命令をセキュア通信付加機構にて取り込み、TCP/IPに対してaccept命令を出し直すが、そのリターンはネゴシエーション完了後とする。

これらのネゴシエーションは、TCP/IPにとっては、コネクション設定完了後の通常のデータ送受信としてみえるため、TCP/IPプロトコルを変更することなく、独自プロトコルを組み込み、実現できた。

5．ストリームデータのパケット化

本方式では、APからのデータをTCP/IPに渡す前でデータ圧縮を行っているため、APが依頼した平文でのデータ長と、実際にTCP/IPに渡し送信される圧縮後のデータ長とは異なった値になる。さらに、ノンブロックモードの送信では、send命令のリターン値として、実際にTCP/IPがバッファに取り込んだ分が返されるのだが、この値は、圧縮後のデータ長に基づいた値であるため、APに対して返す場合は、これを平文に換算し直して渡さねばならない。しかしながら、圧縮済みのデータの一部分が、圧縮前のデータのどれくらいにあたるかを示すことは難しい。さらに、通信中動的に圧縮辞書を更新していく圧縮方式では、圧縮したデータがすべて相手に届かないと、送信側と受信側とで辞書の更新に不一致が生じてしまう。送信側の辞書を部分的に戻す処理が必要になる。

そこで、本方式ではAPからのストリームデータを、セキュア通信付加機構にて所定の大きさ以下になるように分割し、圧縮暗号化処理を加えパケット化してから、TCP/IPに渡す方式を採用した。セキュア通信付加機構の責任において、パケット単位でデータを送り切るようにするため、APに対する平文換算の送信完了データ長の報告は、送信完了したパケット数から容易に計算できる。また、圧縮暗号化は分割単位で行われ、必ず送信されるため、圧縮辞書を戻す処理は必要なくなる。このようにAPとTCP/IPとの間で通信データを分割し、圧縮暗号化を行うことにより、APは、下位の通信制御内で行われている圧縮暗号化処理を意識しなくてすむ。

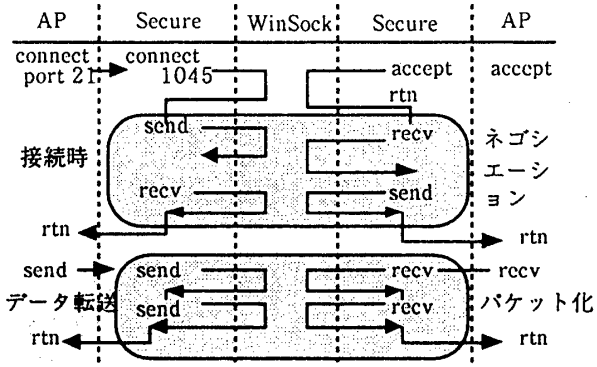

図2　ネゴシエーションとパケット化転送

6．圧縮／暗号化方式のライブラリ化

圧縮方式や暗号化方式は、取り扱うデータの種類やユーザのセキュリティ運用の方針によって、選択はさまざまである。セキュア通信付加機構では、圧縮／暗号化処理を外部のライブラリとして、選択使用できるようにし、圧縮と暗号化の順序制御を行い、かつ、暗号化方式により異なる暗号鍵の取扱いも共通インタフェース化した。新たに開発した圧縮暗号化同時処理アルゴリズムも選択可能である[1]。

7．おわりに

WinSock APIでの命令のやりとりを取り込み、圧縮暗号化処理とネゴシエーションを行うパケット圧縮暗号化通信方式を提案した。これにより、既存のAPやTCP/IPソフトをそのままで、圧縮および暗号化機能の組み込むが可能になる。

参考文献

1) 吉浦 他、「モーバイルに適した圧縮／暗号化通信方式（その２）　圧縮／暗号同時実行アルゴリズム」［情報処理第52回全国大会 3W-6］