(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3783142号
(P3783142)
(45)発行日　平成18年6月7日(2006.6.7)　(24)登録日　平成18年3月24日(2006.3.24)

(51)Int.Cl.　　F I
H04L 12/22 (2006.01)　　H04L 12/22
H04L 29/06 (2006.01)　　H04L 13/00　３０５Ｚ

請求項の数 22　(全 46 頁)

(21)出願番号　特願2005-512962(P2005-512962)
(86)(22)出願日　平成16年7月30日(2004.7.30)
(86)国際出願番号　PCT/JP2004/011304
(87)国際公開番号　WO2005/015827
(87)国際公開日　平成17年2月17日(2005.2.17)
審査請求日　平成17年11月29日(2005.11.29)
(31)優先権主張番号　特願2003-290822(P2003-290822)
(32)優先日　平成15年8月8日(2003.8.8)
(33)優先権主張国　日本国(JP)

早期審査対象出願

(73)特許権者　503287764
ティー・ティー・ティー株式会社
大阪府大阪市北区東天満１丁目１番１９号
アーバンエース東天満ビル
(74)代理人　100122884
弁理士　角田　芳末
(74)代理人　100133824
弁理士　伊藤　仁恭
(72)発明者　尾崎　博嗣
大阪府大阪市北区東天満１丁目１番１９号
アーバンエース東天満ビル　ティー・ティー・ティー株式会社内

最終頁に続く

(54)【発明の名称】通信システム、通信装置、通信方法、及びそれを実現するための通信プログラム

(57)【特許請求の範囲】

【請求項１】

トランスポート層に位置するＴＣＰ又はＵＤＰプロトコルに暗号化機能を追加して通信を行う通信システムであって、

通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行うための接続シーケンス手段と、

通信路の両端で対応する暗号化及び復号化ロジックを取り決める取決め手段と、

送受信する情報単位としてのパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰプロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って暗号化して送信するプロトコル暗号化手段と、

受信した前記暗号化されたプロトコルのペイロードを前記取決め手段により取り決めた復号化ロジックに従って復号化するプロトコル復号化手段と、を備え、

前記取決め手段は、前記接続シーケンス手段が正当な権限を有すると判断して接続した通信相手とのみ前記トランスポート層の前記ＴＣＰ又はＵＤＰプロトコルを用いて前記暗号化及び復号化ロジックに基づいた通信を行うことを特徴とする通信システム。

【請求項２】

トランスポート層に位置するＴＣＰ又はＵＤＰプロトコルに暗号化機能を追加して通信を行う通信システムに用いられる暗号化及び復号化ロジックを取り決める取決め手段と、通信相手が正当な権限を有する通信相手であるかどうかを判断した後に該通信相手との接続を行うための接続シーケンス手段とを備える第１及び第２の通信装置と、前記暗号化及

び復号化ロジックを取り決める取決め手段を備えない第３の通信装置とがそれぞれネットワークに接続されてなる通信システムであって、
前記第１及び第２の通信装置は、送受信する情報単位のパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰプロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って暗号化して送信するプロトコル暗号化手段と、受信した前記暗号化されたプロトコルのペイロードを前記取決め手段により取り決めた復号化ロジックに従って復号化するプロトコル復号化手段からなる暗号化プロトコル処理手段と、前記暗号化及び復号化ロジックを伴わない通常のプロトコル処理手段の両方を備え、
前記第３の通信装置は、前記ＴＣＰ又はＵＤＰプロトコルの暗号化及び復号化ロジックを取り決めるための取決め手段を備えない通常のプロトコル処理手段のみを備え、
前記第１の通信装置が、前記第２の通信装置と通信するときは、前記接続シーケンス手段の判断情報に基づいて、前記暗号化及び復号化ロジックの取決め手段により、
前記暗号化プロトコル手段を選択し、前記暗号化プロトコル手段により通信するとともに、
前記第１の通信装置が前記第３の通信装置と通信するときは、前記接続シーケンス手段の判断情報に基づいて、前記暗号化及び復号化ロジックの取決め手段により、前記暗号化及び復号化を伴わない前記通常プロトコル処理手段により通信するか、前記第３の通信装置との通信を行わないことを選択可能とすることを特徴とする通信システム。
【請求項３】
前記暗号化及び復号化ロジックの取決め手段による取決め候補となりうる暗号化及び復号化ロジックをメモリに記憶ないし回路に実装し、該記憶ないし実装した取決め候補となりうる暗号化及び復号化ロジック定期的に変更するロジック変更手段をさらに備えたことを特徴とする請求項１または２に記載の通信システム。
【請求項４】
前記暗号化及び復号化ロジックの取決め手段が、前記暗号化及び復号化ロジックに関連して、暗号化をしないで平文を取り扱う旨を取り決めることができるようにした請求項１～３のいずれか１項に記載の通信システム。
【請求項５】
トランスポート層に位置するＴＣＰ又はＵＤＰプロトコルに認証機能を追加して通信を行う通信システムであって、
通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行うための接続シーケンス手段と、
通信路の両端で対応する完全性認証ロジックを取り決める完全性認証取決め手段と、
送受信する情報単位であるパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰに該当するプロトコルのペイロードを前記完全性認証取決め手段により取り決めた完全性認証ロジックに従って完全性認証情報を付加して送信するプロトコル完全性認証情報付加手段と、
受信した該完全性認証情報付加されたプロトコルを前記完全性認証取決め手段により取り決めた完全性認証ロジックに従って完全性認証するプロトコル完全性認証手段と、を備え、
前記接続シーケンス手段が正当な権限があると判断した通信相手とのみ、前記トランスポート層にあるＴＣＰ又はＵＤＰプロトコルを用いて前記完全性認証ロジックに基づいた通信を行うことを特徴とする通信システム。
【請求項６】
トランスポート層に位置するＴＣＰ又はＵＤＰを用いて通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行うための接続シーケンス手段と、前記ＴＣＰ又はＵＤＰを用いて完全性認証の取決めを行う完全性認証取決め手段を備える第１及び第２の通信装置と、
前記完全性認証取決め手段を備えない第３の通信装置とがそれぞれネットワークに接続されてなる通信システムであって、
前記第１及び第２の通信装置は、前記完全性認証情報を付加してＴＣＰ又はＵＤＰを処

理する完全性認証プロトコル処理手段と、前記完全性認証情報の付加を行わない通常のＴＣＰ又はＵＤＰを処理する通常プロトコル処理手段の両方を備え、
前記第３の通信装置は、前記完全性認証を伴わない通常のプロトコル処理手段のみを備え、
前記第１の通信装置が、前記第２の通信装置と通信するときは、前記接続シーケンス手段において前記第２の通信装置である通信相手が正当な相手であることを確認してから該第２の通信装置との接続を行って前記完全性認証情報を付加した完全性認証プロトコル手段により通信するとともに、
前記第１の通信装置が前記第３の通信装置と通信するときは、前記接続シーケンス手段における判断情報に基づいて、前記完全性認証情報を付加しないことを決定し、前記通常プロトコル処理手段により第３の通信装置との通信を行うか、又は前記完全性認証取決め手段により前記第３の通信装置である通信相手が正当な権限を有する通信相手でないことを確認して通信の接続を行わないかのいずれかを選択可能とすることを特徴とする通信システム。
【請求項７】
前記完全性認証取決め手段による取決め候補となりうる完全性認証ロジックをメモリに記憶ないし回路に実装しており、該記憶ないし実装した完全性認証ロジックを定期的に変更する完全性認証ロジック変更手段をさらに備えた、請求項５または６に記載の通信システム。
【請求項８】
前記完全性認証取決め手段による前記取決めは、送信データに前記完全性認証情報を付加するか、又は前記完全性認証情報を付加しない旨を取り決めるものである請求項５～７のいずれか１項に記載の通信システム。
【請求項９】
トランスポート層に位置するＴＣＰ又はＵＤＰプロトコルに暗号化機能を追加して通信を行う通信装置であって、
前記ＴＣＰ又はＵＤＰプロトコルを用いて通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行うための接続シーケンス手段と、
通信のための暗号化及び復号化ロジックを取り決める取決め手段と、送信する情報単位としてのパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰプロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って暗号化して送信するプロトコル暗号化手段と、受信した前記暗号化されたプロトコルのペイロードを前記取決め手段により取り決めた復号化ロジックに従って復号化するプロトコル復号化手段とからなる暗号化プロトコル処理手段と、
前記暗号化及び復号化を伴わない通常のプロトコルを処理する通常プロトコル処理手段の両方を備え、
前記暗号化及び復号化ロジックの取決め手段は、前記接続シーケンス手段が、通信相手が正当な権限を有する通信相手であると確認した場合には、前記暗号化プロトコル処理手段を用いて通信を行い、前記通信相手が正当な権限を有していない通信相手であることを確認した場合には、前記通常プロトコル処理手段を用いて通信するか、あるいは通信を行わないかを選択可能としたことを特徴とする通信装置。
【請求項１０】
トランスポート層に位置するＴＣＰ又はＵＤＰプロトコルに認証機能を追加して通信を行う通信装置であって、
前記ＴＣＰ又はＵＤＰプロトコルを用いて通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行うための接続シーケンス手段と、
通信のための完全性認証ロジックを取り決める完全性認証取決め手段と、
送受信する情報単位であるパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰに該当するプロトコルのペイロードを完全性認証取決め手段により取り決めた完全性認証ロジックに従って完全性認証情報を付加して送信するプロトコル完全性認証情報付加手段と、受信

した該完全性認証情報付加されたプロトコルを前記完全性認証取決め手段により取り決めた完全性認証ロジックに従って完全性認証するプロトコル完全性認証手段と、を備え、
前記接続シーケンス手段の判断情報に基づいて、トランスポート層にあるＴＣＰ又はＵＤＰプロトコルを用いて前記完全性認証ロジックに基づいた通信を行うことを特徴とする通信装置。
【請求項１１】
トランスポート層のＴＣＰ又はＵＤＰに該当するプロトコルに暗号化機能を追加して通信する通信方法であって、
前記ＴＣＰ又はＵＤＰプロトコルを用いて通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行う接続ステップと、
通信路の両端で対応する暗号化・復号化ロジックを事前にもしくは動的に取り決める取決めステップと、
送受信する情報単位となるパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰのペイロードに該当するプロトコルを前記取決めステップにより取り決めた暗号化ロジックに従って暗号化して送信するプロトコル暗号化ステップと、
受信した暗号化されたプロトコルを前記取決めステップにより取り決めた復号化ロジックに従って復号化するプロトコル復号化ステップと、を含み、
前記接続ステップにおいて通信相手が正当な権限を有すると判断した場合には、前記トランスポート層のＴＣＰ又はＵＤＰに該当するプロトコルに暗号化処理を施して通信することを特徴とする通信方法。
【請求項１２】
トランスポート層のＴＣＰ又はＵＤＰに該当するプロトコルに暗号化機能を追加して通信する通信方法に用いられる暗号化及び復号化ロジックを取り決める取決め手段と通信相手が正当な権限を有する通信相手であるかどうかを判断した後に該通信相手との接続を行うための接続シーケンス手段とを備える第１及び第２の通信装置と、前記暗号化及び復号化ロジックを取り決める取決め手段を備えない第３の通信装置とがそれぞれネットワークに接続されてなる通信方法であって、
前記第１の通信装置から前記第２の通信装置へ通信するときには、前記接続シーケンス手段の判断情報に基づいて、前記ＴＣＰ又はＵＤＰに該当するプロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って暗号化して通信するとともに、
前記第１の通信装置が前記第３の通信装置と通信するときは、前記接続シーケンス手段の判断情報に基づいて、前記ＴＣＰ又はＵＤＰプロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って暗号化して送信しないことを決定し、前記暗号化ロジックを伴わない通常のＴＣＰ又はＵＤＰプロトコルで通信するか、前記第３の通信装置との通信を行わないかのいずれかを選択して通信することを特徴とする通信方法。
【請求項１３】
前記取決めステップにおいて取決め候補となりうる暗号化及び復号化ロジックをメモリないし回路に記憶しておき、該記憶する暗号化及び復号化ロジックの内容を定期的に変更することを特徴とする請求項１１または１２に記載の通信方法。
【請求項１４】
前記取決めステップにおいて、暗号化及び復号化ロジックについて暗号化をしないで平文を取り扱う旨を取り決めることができる、請求項１１～１３のいずれか１項に記載の通信方法。
【請求項１５】
前記取決めステップより前に、通信相手を認証するステップをさらに含む請求項１１～１４のいずれか１項に記載の通信方法。
【請求項１６】
トランスポート層にあるＴＣＰ又はＵＤＰに該当するプロトコルに認証機能を追加して通信する通信方法であって、
前記ＴＣＰ又はＵＤＰプロトコルを用いて通信相手が正当な権限を有する通信相手であ

るかどうかを判断した後に通信相手との接続を行う接続ステップと、
通信路の両端で対応する完全性認証ロジックを事前に取り決める完全性認証取決めステップと、
送受信する情報単位のパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰのペイロードに該当するプロトコルを前記完全性認証取決めステップにより取り決めた完全性認証ロジックに従って完全性認証情報を付加して送信するプロトコル完全性認証情報付加ステップと、
受信した該完全性認証情報付加されたプロトコルを前記完全性認証取決めステップにより取り決めた完全性認証ロジックに従って完全性認証するプロトコル完全性認証ステップと、を含み、
前記接続ステップにおいて通信相手が正当な権限を有すると判断した場合には、前記トランスポート層にある前記ＴＣＰ又はＵＤＰプロトコルに前記完全性認証情報を付加して通信することを特徴とする通信方法。

【請求項１７】
トランスポート層のＴＣＰ又はＵＤＰを用いて通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行うための接続シーケンス手段と、前記ＴＣＰ又はＵＤＰを用いて完全性認証の取決めを行う完全性認証取決め手段を備える第１及び第２の通信装置との間、若しくは前記完全性認証取決め手段を備える第１又は第２の通信装置と前記完全性認証取決め手段を備えない第３の通信装置との間でネットワークを介して通信する通信方法であって、
完全性認証プロトコルを搭載した前記第１の通信装置が、同じく完全性認証プロトコルを搭載した前記第２の通信装置と通信するときは、前記接続シーケンス手段の判断情報に基づいて前記完全性認証取決め手段により、前記完全性認証情報を付加したＴＣＰ又はＵＤＰを処理する完全性認証プロトコル処理を行って送信し、
前記完全性認証プロトコルを搭載した前記第１又は第２の通信装置が、前記完全性認証プロトコルを搭載しない前記第３の通信装置と通信するときは、前記接続シーケンス手段の判断情報に基づいて前記完全性認証取決め手段により、前記完全性認証情報を付加しないことを決定し、通常のＴＣＰ又はＵＤＰを処理する通常プロトコル処理を行って前記第３の装置と通信を行うか、前記完全性認証取決めをすることなく前記第３の装置と通信を行わないかのいずれかを選択することを特徴とする通信方法。

【請求項１８】
前記完全性認証取決めステップにおいて取決めの候補となりうる完全性認証情報を付加するための完全性認証ロジックをメモリに記憶ないし回路に実装するステップと、該記憶ないし実装した内容を定期的に変更する完全性認証ロジック変更ステップをさらに備えた請求項１６または１７に記載の通信方法。

【請求項１９】
前記完全性認証取決めステップにおいて、完全性認証情報を付加するための完全性認証ロジックにより完全性認証情報付加をしない旨を取り決めることができる、請求項１６～１８のいずれか１項に記載の通信方法。

【請求項２０】
前記完全性認証取決めステップより前に、通信相手を認証するステップをさらに含む請求項１６～１９のいずれか１項に記載の通信方法。

【請求項２１】
トランスポート層に位置するＴＣＰ又はＵＤＰプロトコルに暗号化機能を追加して通信を行う通信システムを実現するプログラムであって、
前記ＴＣＰ又はＵＤＰプロトコルを用いて通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行う機能と、
通信路の両端で対応する暗号化及び復号化ロジックを取り決める取決める機能と、
送受信する情報単位のパケットのうち、少なくとも前記プロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って送信するプロトコルを暗号化する機

能と、
受信した前記暗号化されたプロトコルのペイロードを前記取決めにより取り決めた復号化ロジックに従って復号化するプロトコル復号化機能を具備し、
前記通信相手が正当な権限を有すると判断したときには、前記ＴＣＰ又はＵＤＰプロトコルを用いて前記暗号化及び復号化ロジックを施した通信機能を実現する通信プログラム。
【請求項２２】
トランスポート層にあるＴＣＰ又はＵＤＰに該当するプロトコルに認証機能を追加して通信する通信システムを実現するコンピュータプログラムであって、
前記ＴＣＰ又はＵＤＰプロトコルを用いて通信相手が正当な権限を有する通信相手であるかどうかを判断した後に通信相手との接続を行う機能と、
通信路の両端で対応する完全性認証ロジックを取り決める完全性認証取決め機能と、
送受信する情報単位であるパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰに該当するプロトコルのペイロードを前記完全性認証取決めにより取り決めた完全性認証ロジックに従って完全性認証情報を付加して送信するプロトコル完全性認証情報付加機能と、
受信した該完全性認証情報が付加されたプロトコルを前記完全性認証取決めにより取り決めた完全性認証ロジックに従って完全性認証するプロトコル完全性認証機能を具備し、
前記通信相手が正当な権限を有すると判断したときには、前記ＴＣＰ又はＵＤＰプロトコルを用いて前記完全性認証ロジックを施して通信する機能を実現するための通信プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、通信におけるセキュリティシステム、特に、インターネット上でのデータの「漏洩」及び「改竄」さらには「なりすまし」、「進入」ないし「攻撃」を防ぐための通信システム、特に、通信システムを実現するプロトコルスタックと、通信装置、通信方法、及びそれを実現するためのコンピュータプログラムに関する。
【背景技術】
【０００２】
近年、インターネットを利用した通信は、Ｗｉｎｄｏｗｓ（登録商標）パソコンさえあれば、それをネットワークに接続するだけで、誰でもネットワーク上のコンピュータにアクセスできるため、社会の中で急速に普及拡大している。一方、このインターネット通信の普及拡大に伴って、ハッカー（Hacker）やクラッカー（Cracker）が他人のコンピュータシステムに侵入して、ソフトウエアやデータを盗み見たり、改竄や破壊を行ったりするという社会問題も大きくなっている。
【０００３】
具体的な不正妨害のケースとしては、まず第１に、中心的なシステムが使えなくなるように、ネットワークから大量のメッセージを送りつけコンピュータシステムの運用を妨害するシステム妨害がある。この妨害によってホストが過負荷になるとシステムダウンに陥ってしまうことも起こりうる。
【０００４】
また、ホストのパスワードを入手し、機密情報を盗んだり、情報の改竄や破壊を行ったりする「不正アクセスとなりすまし」の不正妨害がある。この妨害にはコンピュータが保有する情報を勝手に書き換え、人を陥れる卑劣のものもある。また、特定のパソコンに忍び込み、メールアドレスやパスワードなど個人の機密データを搾取するスパイウエアといわれる不正行為も発生している。このようにネットワークに接続したコンピュータが持つデータベースの内容を不正に盗み見る、いわゆる盗聴行為も頻繁に行われている可能性も否定できない。
【０００５】
また、サイト若しくはサーバの運営元で、意図的に個人情報を盗むといった行為や、社

内に潜むスパイなどによるサイバーテロ（Cyber terrorism）といった危機も全くないとはいえない状況である。

【０００６】

さらに、他人のコンピュータにコンピュータ障害をもたらすプログラムである「ウイルス」を送り込むという不正妨害が最近多くなっている。この送り込まれたウイルスに、メールなどを自宅で使用したパソコンが感染し、それを社内に接続した瞬間に社内のパソコン全体が感染したり、ウイルスがコンピュータの中のファイルを破壊させたり、更には、ネットワーク全体をダウンさせたりするといった問題も生じている。

【０００７】

このため、従来のＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）やＵＤＰ（User Datagram Protocol）を利用したインターネット上での通信において、データの「漏洩」「改竄」等を防ぐ機能として、ＩＰｓｅｃ（アイピーセック：Security Architecture for Internet Protocol）やＳＳＬ（Secure Socket Layer）といわれる暗号化通信が利用されている。一般に、暗号化通信には、共通鍵（秘密鍵ともいう）暗号方式と公開鍵暗号方式があるが、ＩＰｓｅｃは共通鍵暗号方式を用いたものが多い。共通鍵暗号方式の方が公開鍵暗号方式より暗号化・復号化の速度が速いという特徴がある。このＩＰｓｅｃに用いられる共通鍵暗号方式は、暗号化と復号化を同じ鍵で行う方式であり、鍵の生成は送信側又は受信側のいずれで生成してもよいが、受信側と送信側とで共通鍵を使うために、鍵の交換時に内容が外部に漏れないよう細心の注意を払わなければならない。

【０００８】

共通鍵暗号方式に用いられるアルゴリズムはＤＥＳ（Data Encryption Standard：米国ＩＢＭ社が開発した共通鍵（秘密鍵）暗号化アルゴリズム）が代表的である。ＩＰｓｅｃもこのＤＥＳを暗号アルゴリズムの１つに採用している。ＩＰｓｅｃは、ＩＥＴＦ（Internet Engineer Task Force）が標準化を進めたものであり、この特徴は、単に特定のアプリケーションだけを暗号化するのではなく、ホストから送信されるあらゆる通信をＩＰレベルで暗号化する点にある。これによりユーザはアプリケーションを意識することなく安全な通信を可能とすることができる。また、ＩＰｓｅｃは、将来にわたって使えるように、それ自体の仕組みを変えることなく使用する暗号アルゴリズムを変更することを可能としている。

【０００９】

ＩＰｓｅｃで用いられる共通暗号鍵としては、ＳＰＩ（Security Pointer Index）と呼ばれる３２ビット符合が用いられ、鍵交換プロトコルとしては、ＩＫＥ（Internet Key Exchange）が用いられる。さらに、ＩＰｓｅｃには、完全性認証のためのプロトコルＡＨ（Authentication Header）が用意されている。

【００１０】

また、ＳＳＬは、米ネットスケープ社（現在ＡＯＬに吸収合併）の開発したセキュリティ機能付ＨＴＴＰプロトコルであり、これを利用することによりクライアントとサーバがネットワーク上でお互いを認証できるようになり、クレジットカード情報などの機密性の高い情報を暗号化してやり取りすることが可能となる。これにより、データの盗聴、再送攻撃（ネットワーク上に流れたデータを盗聴して何度も繰り返して送ってくる攻撃）、なりすまし（本人の振りをして通信する）、データの改竄などを防止することができる。

【００１１】

図２５は従来のＩＰｓｅｃを用いた暗号化通信を行う場合のプロトコルスタックの例を示し、図２６は従来のＳＳＬを用いた暗号化通信を行う場合のプロトコルスタックの例を示している。

【００１２】

ＯＳＩ参照モデルは、最下層（第１層）が物理層、第２層がデータリンク層、第３層がネットワーク層、第４層がトランスポート層、第５層がセッション層、第６層がプレゼンテーション層、最上層（第７層）がアプリケーション層になっている。このＯＳＩ参照モ

デルにおける７階層は、通信機能を７段階に分けたものであり、その階層毎に標準的な機能モジュールを定めている。図２５では、第５層のセッション層までが示されている。

【００１３】

プロトコルスタックとは、ネットワークの各階層における機能を実現するためのプロトコルを選び、階層状に積み上げたソフトウエア群である。

まず、ＯＳＩ参照モデルについて概略を説明すると、第１層の物理層は、信号線の物理的な電気特性や符号の変調方法などを規定した層である。ただ、この層だけが単独で定義・実装されることは少なく、通常は、第２層のデータリンク層と共に、たとえばイーサネット（登録商標）の規格、などとして定義される。

【００１４】

第２層のデータリンク層は、データのパケット化や物理的なノードアドレス、パケットの送受信方法などを規定する層である。この層は、物理的な通信媒体を通して、２つのノード間でパケットをやり取りするためのプロトコルを規定するものであり、各ノードに対して、何らかのアドレスを付け、そのアドレスに基づいてパケットの送信先を特定し、パケットを通信媒体上に送信する。通信媒体としては、銅配線や無線、光ファイバなど、多様なものがある。また、接続形態（トポロジー）も、１対１の対向接続だけでなく、バス型やスター型、リング型など多くの種類がある。通信媒体上に送信されたパケットは、受信側ノードに到着した時点でそのノードに取り込まれ、さらに上位のプロトコル層へと渡される。

【００１５】

物理層とデータリンク層に渡って配置されるＮＩＣ（Network Interface Card） Driverは、パソコンやプリンタなどを構内ネットワーク（ＬＡＮ）につなぐための拡張ボードである。単にネットワークカードという場合はイーサネット（登録商標）につなぐ場合が多い。

【００１６】

このＮＩＣ Driverにより、データを送信したいノード（機器）がケーブルの空き状況を監視して、ケーブルが空くと送信を開始するようにしている。このとき、もし複数のノードが同時に送信を開始するとケーブル内でデータが衝突して破壊されるので、両者は送信を中止し、ランダムな時間を待って送信を再開するのである。これによって一本のケーブルを複数のノードが共有して互いに通信することができる。

【００１７】

第３層のネットワーク層は、任意の２つのノード間での通信方法を規定する層である。ＴＣＰ／ＩＰでいえばＩＰ層に相当する。データリンク層では、同一ネットワーク媒体上のノード間での通信を行うことができるが、その機能を使って、ネットワーク上に存在する任意の２つのノード間で、ルーティング（Routing）を行いながら通信するのがこのネットワーク層の役目である。ここで、ルーティングとはＴＣＰ／ＩＰネットワークにおいて目的のホストまでパケットを送信するときに、最適な経路を選択して送信することをいう。例えば、イーサネット（登録商標）では、同一セグメント上のノード同士でしかお互いに通信できないが、ネットワーク層では、２つのイーサネット（登録商標）セグメント間でパケットをルーティングすることによって通信を行う。また、電話回線を通じてコンピュータをネットワーク（イーサネット（登録商標））に接続するダイヤルアップＰＰＰ（Point to Point Protocol）回線へのルーティング、また、光ファイバを使った専用線へのルーティングなど、物理的なネットワーク媒体によらずにパケットをルーティングすることができる。この目的のため、通常は、物理媒体に依存しないアドレス（ＴＣＰ／ＩＰならば、ＩＰアドレス）を各ノードに割り当て、これに基づいてルーティングを行っている。ＩＰｓｅｃは、このネットワーク層で、つまりＩＰレベルでホストから送信されるあらゆる通信を暗号化するので、ユーザはアプリケーションを意識することなく安全な通信を可能とすることができるのである。

【００１８】

第４層のトランスポート層は、各ノード上で実行されている２つのプロセス間で、エラ

ーのない、仮想的な通信路を実現するためのプロトコル層である。ＴＣＰ／ＩＰでいえばＴＣＰ層に相当する。ネットワーク層では、２つのノード間での通信を行う機能を提供しているが、これを使って、２つのプロセス（アプリケーション）間で、エラーのない、仮想的な通信路を提供するのがこの層の役目である。すなわち、ネットワーク層ではデータを送ることはできるが、そのデータが確実に相手に届くという保証はない。また、送信した順に正しくデータが届くという保証もない。そこで、アプリケーションにとって使いやすくするために、エラーのない通信路を提供するのがこの層である。必要ならばデータの再送・回復処理などを行う。

【００１９】

このトランスポート層にはＴＣＰの他にＵＤＰも配置されるが、このＵＤＰとＴＣＰの違いは、ＴＣＰがデータの補償がされているプロトコルである分、低速であるのに対して、ＵＤＰはデータ補償がない分、高速になっている点である。コンピュータ間の通信のように主としてデータを送る場合にはＴＣＰが用いられ、ＩＰ電話のように音声や画像を送る場合にはＵＤＰが多く用いられる。この第３層のトランスポート層に暗号化処理を施した例は、今まで存在していない。

【００２０】

第５層のセッション層は、セッション（通信の開始から終了まで）の手順を規定する層であり、アプリケーション間でコネクションを開設して通信ができる状態にする層である。この層に配置されるソケット（Socket）は、コンピュータが持つネットワーク内の住所に当たるＩＰアドレスと、ＩＰアドレスのサブアドレスであるポート番号を組み合わせたネットワークアドレスを意味している。コンピュータ同士を接続する場合は、必ずソケット（ＩＰアドレスとポート番号の組）を指定して行う。図２６に示すように、従来の代表的な暗号化通信技術であるＳＳＬは、このセッション層で暗号化通信を実現している。

【００２１】

第６層のプレゼンテーション層は、セッション（通信の開始から終了まで）でやり取りするデータの表現方法や符号化、暗号化などを規定する層である。ＴＣＰ／ＩＰプロトコルでは、この層に相当する部分はなく、通常はアプリケーション自身でストリームデータの処理をハンドリングしている。

【００２２】

また、第７層のアプリケーション層は、アプリケーション間でのデータのやり取りを規定するための層であり、ＴＣＰ／ＩＰプロトコルではこの層に相当する部分はない。例えば、電子メールのフォーマットや、文書の内部構造など、アプリケーション間で相互にデータをやり取りする場合に必要な、共通のデータ構造などを規定する層である。

【００２３】

図２５は、ＩＰｓｅｃを搭載した標準プロトコルスタックであるが、まず、物理層（第１層）とデータリンク層（第２層）に、ＮＩＣ（Network Interface Card） Driverが設けられている。このドライバは、コンピュータなどのハードウエアをネットワークに接続するためのインターフェースカードのドライバであり、その内容はデータ送受信制御ソフトウエアである。例えばイーサネット（登録商標）に接続するためのＬＡＮボードまたはＬＡＮカードがこれに相当する。第３層のネットワーク層は、一部がトランスポート層（第４層）まで伸びたＩＰエミュレータ（emulator）が存在している。このトランスポート層まで延びた部分には、トランスポートとしての機能は実装していない。セッション層に、ネットワーク層の機能を提供しているだけある。このＩＰエミュレータは、ＩＰｓｅｃによる暗号化通信を行うプロトコルと、暗号化通信を行わないプロトコルであるＩＰを用途に応じて切り換えて使う働きをする。また、第３層のネットワーク層にはＡＲＰ（Address Resolution Protocol）が配置されている。このＡＲＰは、ＩＰアドレスからイーサネット（登録商標）の物理アドレスであるＭＡＣ（Media Access Control）アドレスを求めるのに使われるプロトコルである。ＭＡＣは、媒体アクセス制御と呼ばれる、ＬＡＮなどで利用される伝送制御技術であり、データの送受信単位であるフレームの送受信方法やフレームの形式、誤り訂正などを規定する技術として利用されている。

【００２４】
また、このネットワーク層には、ＩＰのエラーメッセージや制御メッセージを転送するプロトコルであるＩＣＭＰ（Internet Control Message Protocol）と、同一のデータを複数のホストに効率よく配送するための又は配送を受けるために構成されるホストのグループを制御するためのプロトコルであるＩＧＭＰ（Internet Group Management Protocol）が設けられている。そして、ネットワーク層の上位層のトランスポート層には、ＴＣＰとＵＤＰが、そしてその上位層であるセッション層にはソケット（Socket）インターフェースが配列されている。
図２６は、暗号化処理プロトコルとしてＳＳＬを具備した標準プロトコルの例であり、ネットワーク層にＩＰｓｅｃを搭載しない代わりにソケット（セッション層）にＳＳＬが搭載されている。この他のプロトコルは図２５に示したものと同じである。
【００２５】
従来の代表的な暗号化通信技術の中で、ＩＰｓｅｃは、ＩＰのパケットを暗号化して送受信するものであり、したがって、ＴＣＰやＵＤＰなどの上位のプロトコルを利用するアプリケーションソフトはＩＰｓｅｃが使われていることを意識する必要がない。
一方、ＳＳＬでは、互いの認証レベルではＲＳＡ（Rivest Shamir Adleman：公開鍵暗号方式を開発者３人の頭文字）公開鍵暗号技術を用いたデジタル証明書が用いられ、データの暗号化ではＤＥＳなどの共通鍵暗号技術が用いられている。このＳＳＬは第５層のセッション層にあるため、特定のアプリケーションに依存することになる。
【００２６】
ＩＰｓｅｃは、ＯＳＩにおける第４層（トランスポート層）より下位の第３層（ネットワーク層）におけるデータの「漏洩」や「改竄」を防ぐ機能として実現したものである（例えば、R. Atkinson、１９９５年８月、「Security Architecture for the Internet Protocol」、ＲＦＣ１８２５を参照。）。これに対して、ＳＳＬは、第５層のセッション層における暗号化技術であり、現在インターネットで広く使われているＷＷＷ（World Wide Web）やＦＴＰ（File Transfer Protocol）などのデータを暗号化して、プライバシーに係る情報や企業秘密情報などを安全に送受信するためのものである。
【００２７】
表１は、ＩＰｓｅｃとＳＳＬの機能を比較して記載したものである。この表から見る限り、ＩＰｓｅｃとＳＳＬは互いに相反する利点と欠点があるように思える。
例えば、クライアント－クライアント間の通信では、ＳＳＬの場合、そのコマンド体系と通信内容が主従の関係、つまりクライアント／サーバとなってしまうことから、サーバを介することなくクライアント－クライアント間の通信はできなかった。すなわち、端末Ａから端末Ｂに秘密のデータをＳＳＬにより暗号化して送る場合には、必ず間にサーバを介在する必要があった。これに対して、ＩＰｓｅｃではこのような制約がないので直接通信が可能となる。
【００２８】
また、ＰＰＰ（Point to Point Protocol）モバイル環境あるいはＡＤＳＬ（Asymmetric Digital Subscriber Line）環境においては、ＩＰｓｅｃは、データの暗号化通信を開始する前に、暗号化方式の決定、鍵の交換、相互認証に使用するプロトコルであるＩＫＥ（Internet Key Exchange）プロトコルを使用した通信の中で、接続先相手の認証を行っている。したがって、ＰＰＰモバイル環境（リモートクライアント）又は、ＡＤＳＬ環境の場合、ＩＰアドレスが固定できないため、ＩＰｓｅｃのゲートウェイ間で、最も使用しているＩＫＥのMainモード、つまり認証の際に通信相手のＩＰアドレス情報を使用するモードを使用することができない。なお、解決策としては、Aggressiveモードを使用することで、ＩＤ情報にＩＰアドレスを使用しなくてもよく、ＩＤ情報に例えばユーザ情報を使用し、既知共有鍵にユーザのパスワードを使用することで相手を特定することができる。但し、Aggressiveモードでは鍵交換情報と同じメッセージの中で接続先相手のＩＤを送信するため、ＩＤは暗号化されずに平文のまま送信することになる。また、ＸＡＵＴＨ（Extended Authentication within IKE）を利用することにより、認証の問題を解決すること

ができるが、ファイアウォールの設定で、リモートクライアントからのアクセスは、ＩＰアドレスが不明であるため、ＩＫＥ、ＩＰｓｅｃを全て許可にする必要があり、セキュリティ上問題が残る。ＳＳＬは、この環境下であっても、通信することが可能である。

【００２９】

また、ＩＰｓｅｃは、ＮＡＴ（Network Address Translation）やＩＰマスカレードに対応することができないという問題がある。これらに対応するためには、例えばＵＤＰのペイロードに載せるといった他の機能と併用しなければならない。ＮＡＴは、インターネットに接続された企業などが１つのグローバルなＩＰアドレスを複数のコンピュータで共有する技術であり、組織内でのみ通用するＩＰアドレス（ローカルアドレス）とインターネット上のアドレス（グローバルアドレス）を相互変換する技術である。ＮＡＴに対応できないのは、ＩＰヘッダがＡＨ（Authentication Header）の認証範囲に入っているため、このローカルアドレスからグローバルアドレスの相互変換ができなくなり、サブネットの異なるローカルアドレス同士の通信ができなくなるからである。

【００３０】

また、ＩＰマスカレードとは、ＬＡＮ内のプライベートアドレスを持つ複数のクライアントからインターネットにアクセスすることを可能とするような仕組みであり、これを利用すると、外部（インターネット）からはＩＰマスカレードが動作している端末しか見えないのでセキュリティ上から見て望ましいといえるものである。ＩＰｓｅｃがＩＰマスカレードに対応できない理由は、ＩＰｓｅｃのＥＳＰ（Encapsulating Security Payload：暗号ペイロード）ヘッダがＩＰヘッダのすぐ後にあるためである。ＩＰマスカレードを実装している通常のルータは、ＩＰヘッダのすぐ後ろには、ＴＣＰ／ＵＤＰのポート番号があると判断している。したがって、ＩＰマスカレードを実装しているルータを経由すると、このポート番号を変更してしまうため、ＩＰｓｅｃは、改竄されたと判断して、ホストの認証ができないという問題が生じる。この問題は、ＵＤＰのペイロードに乗せるためのＮＡＴ－Ｔ（ＮＡＴ－Traversal）をサポートする製品を利用することで回避することができる。但し、ＮＡＴ－Ｔのドラフトバージョンが異なると、ＮＡＴ－Ｔ対応製品同士でも接続することができない。ＳＳＬは、この環境下であっても、通信することが可能である。

【００３１】

これに対し、ハッカーやクラッカーといわれるネットワークの不正侵入者によるＴＣＰ／ＩＰへのさまざまな攻撃、いわゆるＤｏＳ攻撃（Denial of Service：サービスを停止させる攻撃）に対しては、ＳＳＬは無力である。ＴＣＰ／ＩＰプロトコルスタックへのＤｏＳ攻撃、例えば、ＴＣＰ切断攻撃が行われると、ＴＣＰセッションが切れてしまいＳＳＬのサービスは停止してしまうのである。ＩＰｓｅｃは第３層（ＩＰ層）に実装しているため、ＩＰ層にセキュリティ機能を持つので、ＴＣＰ／ＩＰ（第４層、第３層）へのＤｏＳ攻撃を防ぐことができる。しかし、ＳＳＬは、ＴＣＰ／ＩＰ（第４層、第３層）より上の層（第５層）に実装されている暗号化プロトコルであるため、ＴＣＰ／ＩＰへのＤｏＳ攻撃を防ぐことができない。

【００３２】

さらに、物理ノイズが多く通信エラーが多発するような劣悪な通信環境化における通信に対しては、ＳＳＬの方がＩＰｓｅｃに比べて効果的である。すなわち、ＩＰｓｅｃは、エラーの検出をした場合、再送動作を上位のＴＣＰに任せることになる。ＴＣＰは、再送データをＩＰｓｅｃに送るが、ＩＰｓｅｃはこの再送データであることが認識できず、再暗号を行ってしまうのである。ＳＳＬはＴＣＰでエラー回復処理を行うので同じデータを再暗号化することはない。

【００３３】

また、ＩＰｓｅｃでは異なったＬＡＮ間の通信ができない。すなわち、ＬＡＮ内のサブネットアドレスの配布管理は、ＬＡＮ内にあるＤＨＣＰ（Dynamic Host Configuration Protocol）サーバが管理しているため、ＬＡＮ内では、同一のサブネットアドレスが割り振られることはないが、異なったＬＡＮ間の通信の場合、お互いのＬＡＮ内にあるＤＨＣ

Ｐサーバが個別にサブネットアドレスを割り振っているため、同一アドレスが割り振られる可能性がある。このように同一アドレスが割り振られた場合には、ＩＰｓｅｃでは通信することができない。但し、別にＩＰｓｅｃ－ＤＨＣＰサーバを立てて、同一アドレスにならないように管理すれば、通信することができる。ＳＳＬは上述したようにＯＳＩ参照モデルの第５層（セッション層）に位置しているため、下位層のＴＣＰでエラー回復処理を行うことができ、上記のような劣悪な環境下でも通信することが可能となる。

【００３４】

また、異なったネットワーク環境下における通信に対しては、ＩＰｓｅｃは、経由するノード全てを管理し、ＩＰｓｅｃが通過できるように設定変更しなければならないため、管理が大変であるが、ＳＳＬは、この環境下であっても、経由するノードの環境を意識せず通信することが可能である。

さらに、ＩＰｓｅｃは複数のキャリア経由の接続ができないという問題がある。つまり、ＩＰｓｅｃは、経由するノード全てを管理し、ＩＰｓｅｃが通過できるように設定変更しなければならならないため、複数のキャリア経由の接続ができない。例えば、東京と大阪で、別々のキャリアと契約している場合に、接続できないため、別途、高額な専用線を引いているケースもある。ＳＳＬは、この環境下であっても、通信することが可能となる。

【００３５】

また、ＳＳＬは、ＵＤＰの通信をサポートしていないため、ＵＤＰを暗号通信することができない。ＴＣＰも特定のポートしかサポートしていないため、ＴＣＰ全てのポートを暗号通信することができない。これに対して、ＩＰｓｅｃは、ＵＤＰでもＴＣＰでも暗号通信することができる。

【００３６】

さらに、ＳＳＬはアプリケーションに対する互換性を持たない点において問題がある。アプリケーションは、インターネット通信を行う際にソケット（第５層）をプログラムインターフェースとして使用する。このため、アプリケーションがＳＳＬ（第５層）を使用する場合には、このソケットインターフェースをＳＳＬインターフェースに変更しなければならない。従って、ＳＳＬにアプリケーションの互換性はない。これに対し、ＩＰｓｅｃは、ソケット（第５層）の下に位置しているため、アプリケーションは、ソケット（第５層）をプログラムインターフェースとしてそのまま使用することができるのでアプリケーションの互換性がある。

【００３７】

また、ＩＰｓｅｃは、ＩＰアドレス単位で制御することができるのに対し、ＳＳＬは、ソース単位（ＵＲＬ単位、フォルダー単位）で制御することになる。

さらに、ＩＰｓｅｃは最大セグメントサイズが小さくなるという問題がある。すなわち、ＩＰｓｅｃでは、ＥＳＰヘッダ、ＥＳＰトレーラを使用するため、ペイロードが小さくなるため、フラグメント（パケットの分割）が発生し、スループットが低下する。また、ＴＣＰパケットでは、フラグメントが禁止であるため、エンドエンドで、ＩＰｓｅｃの通過する環境を把握し、フラグメントが発生しない最大セグメントサイズを設定する必要がある。これに対し、ＳＳＬは、通過する環境を把握する必要がないため、最大セグメントサイズを設定する必要はない。

【００３８】

以上、表１にもとづいてＩＰｓｅｃとＳＳＬの機能比較について説明したが、後述する本発明のプロトコルであるＴＣＰ２（登録商標出願中）は、これらＩＰｓｅｃとＳＳＬのすべての長所を含み、さらに他にも多くのメリットを有する画期的な暗号通信プロトコルである。

【発明の開示】

【発明が解決しようとする課題】

【００３９】

本発明は、上述のような問題に鑑みてなされたものであり、コンピュータ端末への不正

侵入を防ぐための「暗号化の機能」をアプリケーション・プログラムそれぞれに実装する必要がなく、したがって、アプリケーション・プログラム自体を再作成する必要もなく、かつ上記暗号化機能に対応していない通信相手とも従来の平文による通信でき、さらにはＩＰｓｅｃが利用できない環境（あるいは利用したくない状況）でも暗号化や認証の恩恵を受けることができる通信システム、特にプロトコルスタック、並びに関連する通信装置、通信方法、及びこれらを実現するための通信プログラムを提供することを目的とする。

【課題を解決するための手段】

【００４０】

上記課題を解決し、本発明の目的を達成するため、本発明の通信システムは、トランスポート層に位置するＴＣＰ又はＵＤＰに該当するプロトコルを取り扱う通信システムであって、通信路の両端で対応する暗号化及び復号化ロジックを取り決める取決め手段と、送受信する情報単位としてのパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰに該当するプロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って暗号化して送信するプロトコル暗号化手段と、受信した該暗号化されたプロトコルを前記取決め手段により取り決めた復号化ロジックに従って復号化するプロトコル復号化手段と、を備え、トランスポート層のプロトコルに暗号化及び復号化ロジックを適用して暗号化通信を行うものである。

【００４１】

これにより、従来は存在しなかったＴＣＰ又はＵＤＰレベル独自の暗号化が可能となり、ＩＰより上の層におけるデータ漏洩や改竄の可能性が激減することになる。つまり、ＩＰレベルの暗号化であるＩＰｓｅｃが解除された後のデータも、ＴＣＰ又はＵＤＰレベルの独自の暗号化がなされていることとなり、二重の暗号化という意味で暗号の強度が増すとともに、ＩＰが正当に復号化された直後のデータに対する傍受などインターフェースを狙ったデータ漏洩に対する有効な防御となる。

【００４２】

また、ＩＰが暗号化されない場合にもＴＣＰやＵＤＰだけを暗号化することによりセキュリティを独自に強化することができる。

さらに、ＵＤＰのブロードキャスト機能をパフォーマンス等の観点からＩＰｓｅｃと切り離して独自に働かせる場合があるが、この場合も本発明のＴＣＰ又はＵＤＰレベルの暗号化が有効である。

【００４３】

なお、暗号化及び復号化ロジックの取決めは、通信路の両端で対応する暗号化及び復号化ロジックの事前に取り決めることが好ましい。ここで、通信路とは有線、無線を問わない。衛星を介して通信する方法も含むことは言うまでもない。また、本発明の暗号化及び復号化ロジックの取決めには、フロッピー（登録商標）ディスク、ＣＤ（Compact Disk）、ＵＳＢメモリあるいはＩＣチップ等のリムーバブルメディア（媒体）に暗号化及び復号化ロジックを記憶させて、これらの媒体を送信側と受信側で交換して暗号化及び復号化ロジックの取決めを行うことも含むものである。

【００４４】

また、本発明では、より上位の層、典型的にはＨＴＴＰ等のアプリケーション層への「進入」や「攻撃」等の不正通信パターンの認識を、より下位の層（トランスポート層）で行うことができる。例えば、本発明の通信システムで用いられるプロトコル暗号化手段若しくはプロトコル復号化手段と、従来のクラッキング・プロテクタのような機能モジュール（一般的なクラッキングパターンの検出、破棄ないし通過制限手段）との組み合わせが、上位層であるアプリケーション層より下位層であるトランスポート層のＴＣＰ、ＵＤＰ、さらにその下の層であるネットワーク層に該当するＩＰ、ＡＲＰ、ＩＣＭＰ、ＩＧＭＰ等のいずれかで実現する。これらのプロトコルスタックは、単一のプロトコルスタックとして「ソフトウエアないしハードウエアモジュール」によって実現することができる。

【００４５】

これにより、上述の効果の他に、データの「漏洩」「改竄」さらには「なりすまし」や

「進入」「攻撃」を防ぐ機能についてプロトコルスタック間に重複や隙間がなく費用対効果の大きい通信システムを実現することができる。

【００４６】

また、本発明の通信システムでは、暗号化及び復号化ロジックを取り決める取決め手段を備える第１及び第２の通信装置と、暗号化及び復号化ロジックの取決め手段を備えない第３の通信装置を含み、取決め手段を備える通信装置（第１の通信装置と第２の通信装置）は、ＴＣＰ又はＵＤＰの暗号化及び復号化プロトコル処理手段のほかに、暗号化及び復号化を伴わない通常のＴＣＰ又はＵＤＰを処理する通常プロトコル処理手段をも備え、これら暗号化及び復号化ロジック取決め手段を有する通信装置同士で通信するときには、暗号化及び復号化プロトコル処理手段を用いて通信を行い、取決め手段を備える通信装置（第１及び第２の通信装置）と暗号化及び復号化ロジックの取決め手段を備えない第３の通信装置と通信する場合には、暗号化及び復号化取決め手段により、この通信において暗号化及び復号化を行わないことを決定し、通常のＴＣＰ又はＵＤＰプロトコル処理手段により、通信を行うことができるようにしている。

これにより、本発明による暗号通信に対応していない通信装置との間でも従来どおりの通信を確保することができることになる。

【００４７】

さらに、本発明の通信システムにおいては、暗号化及び復号化ロジックを取り決める取決め手段を備える通信装置（第１又は第２の通信装置）から暗号化及び復号化ロジックを取り決める取決め手段を備えない通信装置（第３の通信装置）に通信する場合に、第１及び第２の通信装置が、暗号化及び復号化ロジック取決め手段により、第３の通信装置との通信を行わないことを決定し、該第３の通信装置との通信を行わないようにすることもできる。

これにより、通信相手の制限及び各セキュリティレベルについて徹底したセキュリティポリシーを採用することができる。

【００４８】

本発明ではまた、暗号化及び復号化ロジック取決め手段による取決めの候補となりうる暗号化及び復号化ロジックをメモリないし回路に記憶しており、該記憶する内容を定期的に変更するロジック変更手段をさらに備えることができる。

これにより、プロトコルスタック自体を再作成したり入れ替えたりする必要なく、新しい暗号化アルゴリズムに対応したり、暗号鍵を変更することにより解読リスクを削減することができる。

【００４９】

さらに、本発明では、暗号化及び復号化ロジック取決め手段が、暗号化・復号化ロジックについて暗号化をしないで平文を取り扱う旨を取り決めることも可能である。

これにより、通信相手、例えばクライアント側のプロトコルスタック等が本発明による暗号化などに対応していない場合でも、従来どおり通信することができる。

なお、このような場合でも、「なりすまし」や「進入」「攻撃」を防ぐいわゆるクラッキング・プロテクタ（ＣＰ）機能については生かすことができる。

【００５０】

本発明は、また、ＴＣＰ又はＵＤＰに該当するプロトコルを取り扱う通信システムであって、通信路の両端で対応する完全性認証ロジックを取り決める完全性認証取決め手段と、送受信する情報単位としてのパケットのうち、少なくとも前記ＴＣＰ又はＵＤＰに該当するプロトコルのペイロードを前記完全性認証取決め手段により取り決めた完全性認証ロジックに従って完全性認証情報を付加して出力又は送信するプロトコル完全性認証情報付加手段と、受信した該完全性認証情報付加されたプロトコルを前記完全性認証取決め手段により取り決めた完全性認証ロジックに従って完全性認証するプロトコル完全性認証手段と、を備えた通信システムを提供するものである。

【００５１】

また、本発明は、トランスポート層のＴＣＰ又はＵＤＰを用いて完全性認証の取決めを

行う完全性認証取決め手段を備える第１及び第２の通信装置と、完全性認証取決め手段を備えない第３の通信装置とがそれぞれネットワークに接続されており、第１及び第２の通信装置は、完全性認証情報を付加してＴＣＰ又はＵＤＰを処理する完全性認証プロトコル処理手段と、完全性認証情報の付加を行わない通常のＴＣＰ又はＵＤＰを処理する通常プロトコル処理手段の両方を備え、第３の通信装置は、完全性認証を伴わない通常のプロトコル処理手段のみを備えており、完全性認証取決め手段を有する通信装置同士（第１の通信装置と第２の通信装置）で通信するときは、この完全性認証取決め手段により、完全性認証情報を付加した完全性認証プロトコル手段により通信するとともに、完全性認証取決め手段を有する通信装置、例えば第１の通信装置が完全性認証取決め手段を備えない通信装置（第３の通信装置）と通信するときには、前記完全性認証情報を付加しないことを決定し、前記通常プロトコル処理手段により通信を行うようにしている。

【００５２】

また、この場合、完全性認証取決め手段を備えた通信装置（第１又は第２の通信装置）が、完全性認証取決め手段を備えない通信装置（第３の通信装置）と通信するときは、完全性認証取決め手段により通信を行わないことを決定し、通信をしないようにすることもできる。

【００５３】

また、本発明では、完全性認証取決め手段による取決め候補となりうる完全性認証ロジックをメモリに記憶ないし回路に実装しており、該記憶する完全性認証ロジック定期的に変更する完全性認証ロジック変更手段をさらに備えることができる。

【００５４】

さらに、本発明では、完全性認証取決め手段が、完全性認証情報付加及び完全性認証をしない旨を取り決めることも可能である。

なお、かかる場合でも、「なりすまし」や「進入」「攻撃」を防ぐクラッキング・プロテクタ（ＣＰ）機能については生かすことができる。

【００５５】

また、本発明は、トランスポート層のＴＣＰ又はＵＤＰに該当するプロトコルを暗号化して通信する通信方法を提供する。この通信方法は、通信路の両端で対応する暗号化・復号化ロジックを事前に取り決める取決めステップと、送受信する情報単位となるパケットのうち、少なくともＴＣＰ又はＵＤＰのペイロードに該当するプロトコルを取り決めた暗号化ロジックに従って暗号化して送信するプロトコル暗号化ステップと、受信した暗号化されたプロトコルを取り決めた復号化ロジックに従って復号化するプロトコル復号化ステップと、を含み、トランスポート層のＴＣＰ又はＵＤＰに該当するプロトコルに暗号化処理を施して通信するものである。

【００５６】

また、本発明の通信方法では、トランスポート層のＴＣＰ又はＵＤＰに該当するプロトコルを暗号化して通信する通信方法に用いられる暗号化及び復号化ロジックを取り決める取決め手段を備える第１及び第２の通信装置と、暗号化及び復号化ロジックを取り決める取決め手段を備えない第３の通信装置とがそれぞれネットワークに接続されている場合の通信方法を提供する。すなわち、暗号化及び復号化ロジックを取り決める取決め手段を備える通信装置（第１及び第２の通信装置）同士で通信するときには、ＴＣＰ又はＵＤＰに該当するプロトコルのペイロードを前記取決め手段により取り決めた暗号化ロジックに従って暗号化して通信するとともに、暗号化及び復号化ロジック取決め手段を備えた通信装置（第１又は第２の通信装置）が暗号化及び復号化ロジック取決め手段を持たない通信装置（第３の通信装置）と通信するときは、ＴＣＰ又はＵＤＰプロトコルのペイロードを取決め手段により取り決めた暗号化ロジックに従って暗号化して送信しないことを決定し、暗号化ロジックを伴わない通常のＴＣＰ又はＵＤＰプロトコルで通信するようにしている。

【００５７】

また、第１又は第２の通信装置と第３の通信装置との通信では、第１又は第２の通信装

置が、第３の通信装置が暗号化及び復号化取決め手段を備えていないことを理由に通信を行わないことを決定し、前記第３の通信装置との通信を行わないようにすることもできる。

【００５８】

また、上記の暗号化及び復号化ロジックの取決めにおいて取決め候補となりうる暗号化・復号化ロジックをメモリないし回路に記憶しておき、該記憶する暗号化・復号化ロジックの内容を定期的に変更することも可能である。

さらに、この取決めステップにおいて、暗号化・復号化ロジックについて暗号化をしないで平文を取り扱う旨を取り決めることもできる。また、本発明による通信方法ではまた、取決めステップより前に、通信相手を認証するステップをさらに含むことができる。

【００５９】

また、本発明は、トランスポート層にあるＴＣＰ又はＵＤＰに該当するプロトコルを暗号化して通信する通信方法であって、通信路の両端で対応する完全性認証ロジックを事前に取り決める完全性認証取決めステップと、送受信する情報単位のパケットのうち、少なくともＴＣＰ又はＵＤＰのペイロードに該当するプロトコルを完全性認証取決めステップにより取り決めた完全性認証ロジックに従って完全性認証情報を付加して送信するプロトコル完全性認証情報付加ステップと、受信した該完全性認証情報付加されたプロトコルを完全性認証取決めステップにより取り決めた完全性認証ロジックに従って完全性認証するプロトコル完全性認証ステップと、を含み、トランスポート層にある前記ＴＣＰ又はＵＤＰプロトコルに完全性認証情報を付加して通信する通信方法を提供する。

【００６０】

そして、さらに本発明は、トランスポート層のＴＣＰ又はＵＤＰを用いて完全性認証の取決めを行う完全性認証取決め手段を備える通信装置（第１及び第２の通信装置）間、若しくは前記完全性認証取決め手段を備える通信装置と前記完全性認証取決め手段を備えない第３の通信装置との間でネットワークを介して通信する通信方法を提供する。この通信方法は、完全性認証プロトコルを搭載した通信装置（例えば、第１の通信装置）が、同じく完全性認証プロトコルを搭載した通信装置（第２の通信装置）と通信するときは、完全性認証取決め手段により、完全性認証情報を付加したＴＣＰ又はＵＤＰを処理する完全性認証プロトコル処理を行って送信し、完全性認証プロトコルを搭載した第１又は第２の通信装置が、前記完全性認証プロトコルを搭載しない第３の通信装置と通信するときは、完全性認証取決め手段により、完全性認証情報を付加しないことを決定し、通常のＴＣＰ又はＵＤＰを処理する通常プロトコル処理を行って通信を行うようにしている。

【００６１】

なお、第１又は第２の通信装置が、完全性認証取決め手段を持たない第３の通信装置と通信する際に、第１又は第２の通信装置が、前記第３の通信装置が完全性認証取決め手段を持たないことを理由に、通信を行わないようにすることもできる。

【００６２】

また、本発明では、完全性認証取決めステップにおいて取決めの候補となりうる完全性認証情報を付加するための完全性認証ロジックをメモリに記憶ないし回路に実装するステップと、該記憶ないし実装する内容を定期的に変更する完全性認証ロジック変更ステップをさらに備えることができる。また、完全性認証取決めステップより前に、通信相手を認証するステップをさらに含むこともできる。

【発明を実施するための最良の形態】

【００６３】

以下、図１～図２４を参照しながら本発明の実施の形態の例を説明する。

図１は、本発明の暗号化通信システムの一実施の形態に用いられるプロトコルスタックを示すものである。

【００６４】

本発明に用いられるプロトコルスタックは、図１に示すように、ＯＳＩ７階層の物理層（第１層）とデータリンク層（第２層）に相当する階層に、ＮＩＣ（Network Interface

Card）のDriver１１が配列される。このドライバは、既に述べたように、コンピュータなどのハードウエアをネットワークに接続するためのインターフェースカードのドライバであり、その内容はデータ送受信制御ソフトウエアである。例えばイーサネット（登録商標）に接続するためのＬＡＮボードまたはＬＡＮカードがこれに相当するものである。

【００６５】

第３層のネットワーク層には、一部がトランスポート層（第４層）まで伸びたＩＰエミュレータ（emulator）３が存在している。上記延びた部分には、トランスポートとしての機能は実装していない。セッション層に、ネットワーク層の機能を提供しているだけある。このＩＰエミュレータ３は、暗号化通信を行うプロトコルである「ＩＰｓｅｃ　ｏｎ　ＣＰ」１３ｂと、「ＩＰ　ｏｎ　ＣＰ」１３ａを用途に応じて切り換えて使う働きをするものである。ここで、「ｏｎ　ＣＰ」とは、クラッキング・プロテクタ（ＣＰ）による、「進入」「攻撃」の監視、破棄、切断ないし通過制限の対象となっていること、又は、設定によりなりうることを示している。

【００６６】

また、ネットワーク層にはＡＲＰ　ｏｎ　ＣＰ（Address Resolution Protocol on Cracking Protector）が配列されている。このＡＲＰ　ｏｎ　ＣＰは、クラッカー（Cracker）への保護対策を具備したＩＰアドレスからイーサネット（登録商標）の物理アドレスであるＭＡＣ（Media Access Control）アドレスを求めるのに使われるプロトコルである。ＭＡＣは、媒体アクセス制御と呼ばれる、ＬＡＮなどで利用される伝送制御技術であり、データの送受信単位であるフレームの送受信方法やフレームの形式、誤り訂正などを規定する技術として利用されている。

【００６７】

ここで、ＩＰエミュレータ１３は、本発明による各種のセキュリティ機能を、従来のＩＰ周辺のスタックに整合させるためのソフトウエア又はファームウエアである。すなわち、ＩＰのエラーメッセージや制御メッセージを転送するプロトコルであるＩＣＭＰ（Internet Control Message Protocol）１４ａ、同一のデータを複数のホストに効率よく配送するための又は配送を受けるために構成されるホストのグループを制御するためのプロトコルであるＩＧＭＰ（Internet Group Management Protocol）１４ｂ、ＴＣＰ１５、ＵＤＰ１６さらにソケット（Socket）インターフェース１７に整合させるためのソフトウエア、又はファームウエア、ないしはハードウエア（電子回路、電子部品）である。このＩＰエミュレータ１３により、ＩＰｓｅｃの暗号化・復号化及び必要な認証情報付加・認証等の前後の適合処理を行うことができる。

【００６８】

このＩＰエミュレータ１３の上層のトランスポート層（第４層）には、ＴＣＰエミュレータ１５とＵＤＰエミュレータ１６が配置されている。ＴＣＰエミュレータ１５は、暗号化通信を行うプロトコルである「ＴＣＰｓｅｃ　ｏｎ　ＣＰ」１５ｂと、通常の通信プロトコルである「ＴＣＰ　ｏｎ　ＣＰ」１５ａを用途に応じて切り換えて使う働きをする。同様に、ＵＤＰエミュレータ１６は、暗号化通信を行うプロトコルである「ＵＤＰｓｅｃ　ｏｎ　ＣＰ」１６ｂと、通常の通信プロトコルである「ＵＤＰ　ｏｎ　ＣＰ」１６ａを用途に応じて切り換えて使う働きをする。

【００６９】

本発明の最も特徴とするべき点は、このトランスポート層（第４層）に、ＴＣＰｓｅｃ１５ｂと、ＵＤＰｓｅｃ１６ｂの暗号化通信プロトコルを搭載したことである。ＴＣＰｓｅｃ１５ｂとＵＤＰｓｅｃ１６ｂについては後述する。

このトランスポート層（第４層）の上層のセッション層（第５層）には、ＴＣＰ及びＵＤＰ等のプロトコルとデータのやりとりを行うソケット（Socket）インターフェース１７が設けられている。このソケットの意味は、既に述べたようにコンピュータが持つネットワーク内の住所に当たるＩＰアドレスと、ＩＰアドレスのサブアドレスであるポート番号を組み合わせたネットワークアドレスを意味しており、実際には、一連のヘッダの追加ないし削除をまとめて行う、単一のソフトウエアプログラムモジュール（実行プログラム等

）あるいは単一のハードウエアモジュール（電子回路、電子部品等）からなっている。

【００７０】

このソケットインターフェース１７は、さらに上位のアプリケーション（図２で示すＥＣアプリケーション及び図３で示す放送アプリケーション等）からの統一的なアクセス方式を提供するものであり、引数の種類や型など従来と同様のインターフェースを保つようにしている。

ＴＣＰエミュレータ１５は、トランスポート層において、データの漏洩・改竄の防止の機能、すなわち暗号化、完全性認証及び相手認証等の機能を持つＴＣＰｓｅｃ１５ｂと、このような暗号化、完全性認証、及び相手認証等の機能を持たない通常のプロトコルＴＣＰ１５ａのいずれかにパケットを振り分ける働きをもっている。また、ＴＣＰｓｅｃ１５ｂ及びＴＣＰ１５ａのいずれもクラッキング・プロテクタ（ＣＰ）を備えているため、そのいずれを選択した場合でも、クラッカーによる「進入」「攻撃」に対して防御する機能を実現することができる。ＴＣＰエミュレータ１５は上位層であるソケットとのインターフェースの役割も果たしている。

【００７１】

また、既に述べたようにＴＣＰがエラー補償機能を有するのに対して、ＵＤＰはエラー補償機能を持たないが、その分転送速度が速く、かつブロードキャスト機能を備えているという特徴がある。ＵＤＰエミュレータ１６は、ＴＣＰエミュレータ１５と同様に、データの漏洩・改竄の防止の機能、すなわち暗号化、完全性認証及び相手認証等の機能を持つＵＤＰｓｅｃ１６ｂと、このような暗号化、完全性認証、及び相手認証等の機能を持たない通常のプロトコルＵＤＰ１６ａのいずれかにパケットを振り分ける働きを持っている。

【００７２】

図１に示すように、ソケット１７、ＴＣＰエミュレータ１５、ＵＤＰエミュレータ１６、「ＴＣＰｓｅｃ　ｏｎ　ＣＰ」１５ｂ、「ＵＤＰｓｅｃ　ｏｎ　ＣＰ」１６ｂ、「ＴＣＰ　ｏｎ　ＣＰ」１５ａ、「ＵＤＰ　ｏｎ　ＣＰ」１６ａ、「ＩＣＭＰ　ｏｎ　ＣＰ」１４ａ、「ＩＧＭＰ　ｏｎ　ＣＰ」１４ｂ、ＩＰエミュレータ１３、「ＩＰ　ｏｎ　ＣＰ」１３ａ、及び「ＡＲＰｏｎＣＰ」１２からなるプロトコルスタックが本発明の暗号化処理を行うためのプロトコルスタックであり、以下、このプロトコルスタックを総称してＴＣＰ２（登録商標出願中）と呼ぶことにする。なお、ＴＣＰ２には「ＩＰｓｅｃ　ｏｎ　ＣＰ」１３ｂは必須のものとして含まれていないが、「ＩＰｓｅｃ　ｏｎ　ＣＰ」１３ｂを含めてＴＣＰ２とすることもできる。

【００７３】

本発明のＴＣＰ２は、ＴＣＰ、ＵＤＰ、ＩＰ、ＩＰｓｅｃ、ＩＣＭＰ、ＩＧＭＰ、ＡＲＰの標準プロトコルにＣＰ（クラッキング・プロテクト）を実装し、各プロトコルスタックに対する通信からの攻撃、及び、アプリケーション・プログラムからの攻撃（トロイの木馬、プログラムの改竄、正規ユーザの不正使用）を防御することができる。また、ＴＣＰ２では、ＴＣＰエミュレータ１５を実装し、このＴＣＰエミュレータ１５は、セッション層にあるソケット（Socket）１７、及びネットワーク層にあるＩＰエミュレータ１３から見て、互換性を保つため、外向きには標準ＴＣＰと同じに見せることができる。実際は、ＴＣＰ２の機能として、ＴＣＰとＴＣＰｓｅｃを切り替えて実行する。ＴＣＰｓｅｃは、本発明であるトランスポート層での暗号化及び認証機能である。

【００７４】

また、同様に、ＴＣＰ２では、ＵＤＰエミュレータ１６を実装しており、ＵＤＰエミュレータ１６は、セッション層であるソケット（Socket）１７、及び、ネットワーク層であるＩＰエミュレータ１３から見て、互換性を保つため、外部からは標準ＵＤＰとして見せることができる。実際は、ＴＣＰ２の機能として、ＵＤＰ、ＵＤＰｓｅｃを切り替えて実行する。ＵＤＰｓｅｃは、本発明であるトランスポート層での暗号化及び認証機能である。

【００７５】

次に、ＴＣＰ２において、特に重要な機能である「データ漏洩」を防ぐ機能であるＴＣ

Ｐｓｅｃ１５ｂ及びＵＤＰｓｅｃ１６ｂについて説明する。

ＴＣＰｓｅｃ１５ｂ及びＵＤＰｓｅｃ１６ｂのための暗号化・復号化の方法（アルゴリズム、ロジック（論理））としては、公知の秘密鍵（共通鍵）暗号アルゴリズムが用いられる。例えば、１９６０年代にＩＢＭ社によって開発された秘密鍵暗号アルゴリズムであるＤＥＳ（Data Encryption Standard）や、その改良版としての３ＤＥＳが用いられる。また、その他の暗号アルゴリズムとしては、１９９２年にスイス工科大学のJames L.Massey氏とXuejia Lai氏によって発表されたＩＤＥＡ（International Data Encryption Algorithm）も用いられる。この暗号アルゴリズムは、データを６４ビットのブロックに区切って暗号化するもので暗号鍵の長さは１２８ビットである。秘密鍵暗号を効率よく解読する線形解読法や差分解読法に対しても十分な強度を持つように設計されている。

【００７６】

また、本発明に用いられるＴＣＰｓｅｃ１５ｂ及びＵＤＰｓｅｃ１６ｂの暗号方式として、ＦＥＡＬ（Fast data Encipherment Algorithm）、ＭＩＳＴＹ、ＡＥＳ（Advanced Encryption Standard）といった暗号方式も利用されるほか、また、独自に作成した秘密の暗号化・復号化アルゴリズムを利用することもできる。ここで、ＦＥＡＬは、日本電信電話株式会社（当時）が開発した暗号方式で、暗号化と復号化に同じ鍵をもちいる秘密鍵型の暗号方式である。このＦＥＡＬは、ＤＥＳに比べて高速に暗号化と復号化ができるという利点がある。

【００７７】

次に、同じく本発明に利用される暗号方式であるＭＩＳＴＹは、三菱電機株式会社が開発した秘密鍵型の暗号方式であり、ＩＤＥＡと同様にデータを６４ビットのブロックに区切って暗号化する。鍵の長さは１２８ビットである。暗号化と復号化には同じプログラムが使われる点はＤＥＳなどと同じである。この方式も秘密鍵暗号を効率よく解読する線形解読法や差分解読法に対しても十分な強度を持つように設計されている。

【００７８】

また、ＡＥＳは、米国商務省標準技術局によって選定作業が行われている、米国政府の次世代標準暗号化方式であり、現在の標準的な暗号方式であるＤＥＳに代わる次世代の暗号標準として開発が進められたものである。世界に公募して集められて幾つかの暗号方式の中から、２０００年１０月に、ベルギーの暗号開発者Joan Daemen氏とVincent Rijmen氏が開発したＲｉｊｎｄａｅｌという方式が採用された。

【００７９】

このように、本発明のＴＣＰｓｅｃ１５ｂ及びＵＤＰｓｅｃ１６ｂの暗号方式としては、既に知られているさまざまな秘密鍵の暗号アルゴリズムを採用することができるほか、ユーザが独自に開発した秘密鍵（共通鍵）暗号方式も利用することが可能である。

【００８０】

さらに、いわゆる「なりすまし」及び「データの改竄」などを防ぐための「相手認証」及び「完全性認証」の方法として、公開鍵や事前秘密共有（Pre-shared）を利用したアルゴリズム、例えば、ＭＤ５（Message Digest 5）、ＳＨＡ１（Secure Hash Algorithm 1）などの認証アルゴリズムが用いられる。また、このような公知の認証アルゴリズムに代えて独自の一方向関数を利用したアルゴリズムを採用することもできる。

【００８１】

このＭＤ５は、認証やデジタル署名に用いられるハッシュ関数（一方向要約関数）の一つであり、原文を元に固定長のハッシュ値を発生し、これを通信経路の両端で比較することにより、通信途中で原文が改竄されていないかを検出することができるものである。このハッシュ値は擬似的な乱数のような値をとり、これを基にしては原文を再生できないようになっている。また、同じハッシュ値を生成する別のメッセージを作成することも困難になっている。

【００８２】

ＳＨＡ１も、認証やデジタル署名などに使われるハッシュ関数の一つであり、２の６４乗ビット以下の原文から１６０ビットのハッシュ値を生成し、通信経路の両端で比較する

ことにより、通信途上の原文の改竄を検出するものである。この認証アルゴリズムは従来のインターネットの暗号化通信の代表的なものであるＩＰｓｅｃにも採用されている。

【００８３】

なお、これらの認証アルゴリズムについては、ＤＨ（Diffie-Hellman）公開鍵配送法や、ＩＰｓｅｃと同様のＩＫＥ（Internet Key Exchange）プロトコル（ＵＤＰの５００番）などにより安全な鍵交換ができるように設計され、しかも、定期的に暗号化／完全性認証アルゴリズム（論理）自体やそのための鍵の集合／定義域が変更されるように、プロトコルドライバープログラム（ＴＣＰｓｅｃ１５ｂ、ＵＤＰｓｅｃ１６ｂなど）によりスケジュールされている。

【００８４】

次に、図２に基づいて、本発明の第１の実施の形態である暗号化方式ＴＣＰ２（特に、ＴＣＰｓｅｃ）を用いた暗号化通信について説明する。図２は、特に、ＥＣ（Electronic Commerce：電子商取引）アプリケーションに応用した通信に適用する例である。

【００８５】

図２は、ネットワーク２０に接続されたＥＣアプリケーション用のクライアント端末３ａ、３ｂ、３ｃが、いわゆるルータ又はゲートウェイのようなネットワーク制御機器２を介して他のネットワーク３０に接続されたホストコンピュータ（いわゆるサーバとして機能する通信装置）に接続された場合の全体構成を示す図である。

【００８６】

ネットワーク２０に接続されるクライアント端末３ａ、クライアント端末３ｂ及びクライアント端末３ｃのうち、クライアント端末３ｂと３ｃは、本発明のＴＣＰ２を実装していない。つまりクライアント端末３ｂと３ｃには、本発明の暗号化方式のためのプロトコルであるＴＣＰｓｅｃもＵＤＰｓｅｃも実装されていない。ＴＣＰ２をサポートしているクライアント端末は３ａのみである。そして、クライアント端末３ｂについては、不図示のネットワークポリシー設定により通常のプロトコル処理による接続、すなわち、ＴＣＰレベルについては、「データ漏洩」を防ぐ暗号化機能、「データ改竄」を防ぐ完全性認証機能、及び「なりすまし」を防ぐ相手認証機能は伴わない接続を行うようにしている。

【００８７】

何れのクライアント端末３ａ～３ｃにおいても、ソケット（Socket）の上層には、ＥＣ用のアプリケーションソフトウエアが実装されている。また、ネットワーク３０に接続されたホストコンピュータ１は、ＴＣＰ２を搭載しており、そのソケット１７の上層にＥＣアプリケーションソフトウエア１８が実装されている。図２ではＵＤＰｓｅｃなどの不使用のプロトコルを省略しているが、このホストコンピュータ１のプロトコルスタックの構造は図１のプロトコルスタックの構造であるＴＣＰ２を構成するソフトウエアは全て搭載されている。

【００８８】

すなわち、まず第１層（物理層）と第２層（データリンク層）にまたがってＮＩＣドライバ（ＮＩＣ　Driver）１１が配置され、その上層（第３層）のネットワーク層にはＡＲＰ（Address Resolution Protocol）１２とＩＰエミュレータ１３が配置されている。そして、第４層のトランスポート層にＴＣＰエミュレータ１５とＵＤＰ１６が配置される。図２にＵＤＰエミュレータ（ＵＤＰｓｅｃとＵＤＰを含む）の記載がないのは、第１の実施形態のＥＣアプリケーションに対する暗号化通信としては速度よりも誤り補償に重点をおくＴＣＰｓｅｃが使用されるからである。このことは、ホストコンピュータがＵＤＰｓｅｃを搭載していないことを意味するものではない。ＴＣＰ２を搭載することは、既に説明したようにＵＤＰｓｅｃとＴＣＰｓｅｃの両方を搭載していることを意味している。

【００８９】

ネットワーク２０に接続されたクライアント端末３ａ、３ｂ、３ｃとネットワーク３０に接続されるホストコンピュータ１を介在するネットワーク制御機器２のプロトコルスタックは、ＮＩＣドライバ、ＡＲＰ、ＩＰがスタックとして積み上げられたファームウエア（不揮発メモリ付電子回路）により構成されている。

また、クライアント端末３ａは、本発明のＴＣＰ２をサポートする端末であるが、ここではＴＣＰｓｅｃにのみ対応した通信装置を備えた端末としてプロトコルスタックが示されている。クライアント端末３ｂと３ｃは本発明のＴＣＰ２をサポートしていない。

【００９０】

クライアント端末３ａには、ネットワークを通して又はＣＤ－ＲＯＭのような記録媒体を介して事前に配布されるプロトコルドライバソフトウエアが実装されている。また、クライアント端末３ｂ、クライアント端末３ｃに対しても同様にプロトコルドライバソフトウエアが事前に配布され、実装される。

【００９１】

特に、クライアント端末３ｃについては、従来の暗号化方式であるＩＰｓｅｃを実装しているが、ネットワーク制御機器（ルータ）２がＴＣＰポート番号を参照したＩＰマスカレードを行っているためＩＰｓｅｃが有効に使えないようになっている。更にクライアント端末３ｃについては、不図示のネットワークポリシー設定により接続要求を破棄するようにしている。なお、このようなネットワークポリシーの設定ないしプロトコルを実装しているかどうかの確認（受信パケット解析等）自体については当業者に周知の事項であるのため、本明細書では説明を省略する。

【００９２】

ホストコンピュータ１がクライアント端末３ａと通信するときは、本発明のＴＣＰ２、特にＴＣＰｓｅｃに基づいた暗号化及び復号化取決めにより、通信を行うことになるが、ホストコンピュータ１がクライアント端末３ｂ又は３ｃと通信するときは、本発明のＴＣＰ２（特に、ＴＣＰｓｅｃ）による暗号化及び復号化の取決めがされない状態、つまり通常のＴＣＰプロトコルによる通信が行われることになる。もちろん、ホストコンピュータ１がＩＰｓｅｃをサポートしているクライアント端末３ｃと通信する場合には、ＩＰｓｅｃによる暗号化通信をすることができることは当然である。なお、ホストコンピュータ１が、ＴＣＰ２が搭載されていないクライアント端末３ｂ又は３ｃと通信しようとしてもホストコンピュータ１の有するＴＣＰ２の働きで、通信をストップさせることも可能である。

【００９３】

また、この実施の形態では、ネットワークを介してホストコンピュータ１とクライアント端末３ａとが暗号化及び復号化ロジックの交換を行うようにしているが、ＦＤやＣＤ、ＵＤＢメモリ等のリムーバブルメディアを用いて予め通信相手同士で暗号化及び復号化のための取決めロジックを交換しておくこともできることは言うまでもない。

【００９４】

次に、図３に基づいて、本発明の第２の実施形態であるＴＣＰ２の中のＵＤＰｓｅｃの暗号化方式を用いた暗号化通信について説明する。図３は、ネットワーク２０に接続された放送アプリケーション用のクライアント端末４ａ、４ｂ、４ｃが、いわゆるルータ又はゲートウェイのようなネットワーク制御機器２を介して別のネットワーク３０に接続されたホストコンピュータ（いわゆるサーバとして機能する通信装置）１と通信する通信システムの全体構成を示す図である。

【００９５】

図３は、クライアント端末４ａ、４ｂ、４ｃ及びホストコンピュータ１のプロトコルスタックを示しているが、ＴＣＰ２をサポートしているクライアント端末は４ａと４ｂである。つまり端末４ａと４ｂだけがＵＤＰｓｅｃを備えている。各クライアント端末のソケット（Socket）の上層には、放送用のアプリケーションソフトウエアが実装されている。また、ネットワーク３０に接続されたホストコンピュータ１もＴＣＰ２を搭載しており、そのソケット１７の上層に放送アプリケーションソフトウエア１９が実装されている。図３のホストコンピュータ１も図２のホストコンピュータ１と同様に、図１のプロトコルスタックの構造であるＴＣＰ２を構成するソフトウエアは全て搭載している。

【００９６】

ホストコンピュータ１が保有するプロトコルスタックは、図２のホストコンピュータ１

のプロトコルスタックと略同じであるが、図２のホストコンピュータ１のプロトコルスタックと異なる点は、ＴＣＰエミュレータの代わりにＵＤＰエミュレータ１６がある点である。これは放送アプリケーションソフトでは画像等の大量のデータが取り扱われるため、データ伝送のような誤り補償よりも、高速性が重視されるためである。

【００９７】

ネットワーク２０に接続されたクライアント端末４ａ、４ｂ、４ｃとネットワーク３０に接続されるホストコンピュータ１を介在するネットワーク制御機器２のプロトコルスタックは、ＮＩＣドライバ、ＡＲＰ、ＩＰがスタックとして積み上げられたファームウエア（不揮発メモリ付電子回路）により構成されている。

【００９８】

また、クライアント端末４ａは、本発明のＴＣＰ２をサポートする端末であるが、ここではＵＤＰｓｅｃにのみ対応した通信装置を備えた端末であり、クライアント端末４ｂは、本発明のＵＤＰｓｅｃ及び公知のＩＰｓｅｃに対応した通信装置であり、クライアント端末４ｃは公知のＩＰｓｅｃにのみ対応した通信装置である。このクライアント端末４ｃは本発明のＴＣＰ２をサポートしていない。これらのクライアント端末４ａ～４ｃには、図２のクライアント端末３ａ～３ｃと同様に、ネットワークを通して又はＣＤ－ＲＯＭのような記録媒体を介して事前に配布されるプロトコルドライバソフトウエアが実装されている。

【００９９】

また、特に「データ漏洩」防止のための暗号化・復号化ロジック及び「データ改竄」防止のための認証情報付加・認証ロジックについては、ホストコンピュータ１とクライアント端末４ａ、４ｂ、４ｃの間で対応している必要がある。公知のいわゆるＩＰｓｅｃと同様のポリシーで取決めを行うこともできるが、本発明の第２の実施形態においては、事前にプロトコルドライバソフトウエア自体を配布しているので、より簡潔な独自のプロトコルにより秘密鍵等を取り決めたり、より簡易な構造のパケットを使用したりすることもできる。また、公知の暗号化・復号化及び認証アルゴリズムでなく、独自で作成した暗号化・復号化及び認証アルゴリズム（ロジック）自体をプロトコルドライバのソフトウエアモジュール等として組み込むこともできる。

【０１００】

なお、クライアント端末４ｃは、ＴＣＰ２を実装していないが、インターネットで利用される公知のＩＰｓｅｃを実装しているため、これによってある程度セキュアな通信をすることができる。しかし、クライアント４ａと４ｂは、対象とする放送アプリケーションのパフォーマンスないしセキュリティポリシーの都合上、ＩＰｓｅｃではなく本発明によるＴＣＰ２の構成要素であるＵＤＰｓｅｃを実装して使用している。ＩＰｓｅｃではなくＵＤＰｓｅｃを利用する理由は、例えば、ＵＤＰポート番号部分（ＩＰペイロードに属する）をＩＰｓｅｃで暗号化することによるパフォーマンスの低下など、ＩＰｓｅｃ自体に脆弱さがあるからである。

【０１０１】

また、通信相手が正しいものかどうかを判断する相手認証プロトコルを、本発明のＴＣＰ２のＴＣＰ又はＵＤＰプロトコルスタック、つまりＴＣＰｓｅｃ又はＵＤＰｓｅｃに埋め込むことにより、通信相手相互間で上位アプリケーションを意識することなく、通信相手認証機能を実施することができる。この場合、コスト増にならない範囲で実質的に通信パケット数やパケット長等を増やすこともできる。

【０１０２】

また、特に、ネットワーク内で不特定多数の相手に向かってデータを送信するブロードキャスト機能を実施するに際して、本発明による暗号化方式であるＵＤＰｓｅｃを利用する場合には、ブロードキャストを受けるクライアント端末３ａ、３ｂがネゴシエーション（取り決め）を開始し、通信相手認証や通信用秘密鍵を得る。そして、クライアント端末３ａや３ｂは、通信相手の認証を行って通信用の秘密鍵を取得するまでは、ホストコンピュータ１からのＵＤＰｓｅｃによる配信データを復号化することができない。

【０１０３】
次に、図４に基づいて、本発明の第１及び第２の実施形態の通信で送受信されるパケット構造と、その暗号化範囲及び完全性認証の適用範囲について説明する。
図４（ａ）はＴＣＰｓｅｃ／ＩＰｓｅｃのパケット構造と各暗号化範囲と完全性認証の適応範囲を示し、図４（ｂ）（ｃ）はそれぞれＴＣＰｓｅｃ／ＩＰ、ＵＤＰｓｅｃ／ＩＰのパケット構造と各暗号化範囲及び完全性認証の適用範囲を示したものである。
【０１０４】
図４（ａ）に示すように、ＴＣＰｓｅｃ／ＩＰｓｅｃのパケット構造は、ＩＰヘッダ４１のすぐ後に、ＩＰｓｅｃのＥＳＰヘッダ４２が続いている。続いてＴＣＰヘッダ４３とＴＣＰｓｅｃの付加情報４４が設けられ、その後にアプリケーションデータ４５が続く構造になっている。そして、アプリケーションデータ４５の後には、ブロック暗号で発生するデータのブランクやそのブランクの長さ、次のヘッダの番号などの暗号データをサポートする情報であるＴＣＰｓｅｃ付加トレーラ４６が配列され、その後にＴＣＰｓｅｃの付加認証データ４７が配列される。そして、さらにその後にＩＰのためのＥＳＰトレーラ４８とＥＳＰ認証データ４９が配列されるパケット構造になっている。
【０１０５】
このうち番号４１、４２、４８、４９で示される部分はＩＰｓｅｃ用の情報であり、番号４３、４４、４６、４７が本発明によるＴＣＰ２の中心的な役割を果たすＴＣＰｓｅｃに関連する情報である。なお、ここではＴＣＰｓｅｃもＩＰｓｅｃに準じた配列としたが、採用する暗号化や認証のアルゴリズムによっては、ＴＣＰｓｅｃの付加情報４４と付加トレーラ４６を省略したり、ＴＣＰｓｅｃの付加認証データ４７を削減したりしても利用可能である。
【０１０６】
図４（ａ）に示すＴＣＰ２のパケット構造においては、ＴＣＰｓｅｃとＩＰｓｅｃの二つの方式で暗号化が行われる。この場合、まず送信側では、ＴＣＰｓｅｃ側を最初に暗号化して、ＴＣＰｓｅｃ認証データを付加する。次に、ＩＰｓｅｃを暗号化し、ＩＰｓｅｃ認証データを付加している。そして、受信側では、まず、ＩＰｓｅｃを復号化して、ＩＰｓｅｃ認証データにより受信パケットのデータを検証し、続いてＴＣＰｓｅｃ側を復号化して、ＴＣＰｓｅｃ認証データで受信パケットのデータを検証する。
【０１０７】
このように、図４（ａ）に示すようなパケット構造を有するデータでは、ＩＰｓｅｃとＴＣＰｓｅｃの二種類の暗号アルゴリズムを用いて暗号化し、さらに完全性認証を行うので、ＩＰｓｅｃのみと比べて外部からの侵入等にたいして格段に強固な暗号化通信システムを確立することができる。ＴＣＰｓｅｃにより暗号化される範囲は、アプリケーションデータ４５、ＴＣＰｓｅｃ付加トレーラ４６の部分であり、ＴＣＰｓｅｃによる認証範囲としては上記暗号化範囲にさらにＴＣＰｓｅｃ付加情報４４が加えられる。なお、従来のＩＰｓｅｃで暗号化される暗号化範囲は、ＴＣＰヘッダ４３からＥＳＰトレーラ４８までの部分であり、その認証範囲は、ＥＳＰヘッダ４２からＥＳＰトレーラ４８までの範囲となる。
【０１０８】
図４（ｂ）は、ＴＣＰｓｅｃ／ＩＰのパケット構造を示しており、図４（ａ）と異なり、ＩＰヘッダ４１のすぐ後に、ＴＣＰヘッダ４３及びＴＣＰｓｅｃ付加情報４４が続き、更にアプリケーションデータ４５が続く構造になっている。そして、アプリケーションデータ４５の後には、ブロック暗号で発生するデータのブランクやそのブランクの長さ、次のヘッダの番号などの暗号データをサポートする情報であるＴＣＰｓｅｃの付加トレーラ４６とＴＣＰｓｅｃの付加認証データ４７が配列される構造となっている。
【０１０９】
ここで、番号４３、４４、４６、４７がＴＣＰｓｅｃに特徴的な情報となる。ただし、上述したようにこれらの情報は、採用する暗号化／認証アルゴリズムによっては、ＴＣＰｓｅｃ／ＩＰの使用していないヘッダフィールド部分などに分散したり、個々のパケット

からは逆算・推測できない独立した事前取決め（ネゴシエーション）により省略したりできるものである。また、ＩＰ層の上層に当たるＴＣＰ及びＩＰを使用していないプロトコルフィールドを用いて、図４（ｂ）に示すようなＴＣＰｓｅｃ／ＩＰパケットを構成することにより、より下層のＩＰのみに着目したＩＰｓｅｃパケットよりもパケット長を簡単に削減することができるようになる。なお、ここで暗号化範囲は、図示の通り、アプリケーションデータ４５、ＴＣＰｓｅｃ付加トレーラ４６であり、認証範囲は上記暗号化範囲の他に、ＴＣＰｓｅｃの付加情報４４が加えられる。

【０１１０】

図４（ｃ）は、本発明におけるＵＤＰｓｅｃ／ＩＰのパケット構造を示すものであり、ＵＤＰｓｅｃ付加情報４４ａ、ＵＤＰｓｅｃ付加トレーラ４６ａ及びＵＤＰｓｅｃ付加認証データ４７ａがＵＤＰｓｅｃをサポートするために必要な情報となる。この暗号化範囲は、図示の通り、アプリケーションデータ４５ａ、ＵＤＰｓｅｃ付加トレーラ４６ａであり、認証範囲は上記暗号化範囲の他に、ＵＤＰｓｅｃ付加情報４４ａが加えられる。

【０１１１】

次に、本発明の第１の実施の形態であるＴＣＰｓｅｃを用いた暗号化処理システムの動作を図５～図６、図８～図１４に示す流れ図、及び図７に示すシーケンス図に基づいて説明する。

【０１１２】

図５は、ＴＣＰ、並びに、ＴＣＰｓｅｃのパッシブオープン（図７のホストＢに相当する接続待ちのオープンであり、例えば、Ｗｅｂサーバが、この状態でオープンする。）における処理の流れ図であり、上位アプリケーション・プログラムで、接続待ちオープンした場合に、このＴＣＰ／ＴＣＰｓｅｃパッシブオープン処理がスタートする（ステップＳ１）。なお、この部分は、図７でいうとホストＢ側の処理がこれに相当する。

【０１１３】

まず、最初に、オープンするポート番号の解析が行われる（ステップＳ２）。この解析では、例えば、Ｗｅｂサーバの場合は、ＴＣＰポートの８０番を使用して、その定義状態を確認する。そして次にこのポート番号８０が、ＴＣＰｓｅｃのパッシブオープンが許可されているかどうかを判定する（ステップＳ３）。ステップＳ３においてＴＣＰｓｅｃのパッシブオープンが許可されていない場合は、今度はＴＣＰパッシブオープンが許可されているかどうかが判断される（ステップＳ４）。判断ステップＳ４でＴＣＰパッシブオープンも許可されていない場合は、ＴＣＰｓｅｃもＴＣＰも許可されていないことになり、ＴＣＰ／ＴＣＰｓｅｃパッシブオープンは失敗となり、処理を中断する（ステップＳ７）。

【０１１４】

判断ステップＳ４でＴＣＰパッシブオープンが許可されている場合、すなわちＴＣＰｓｅｃパッシブオープンは許可されていないがＴＣＰパッシブオープンは許可されているときは、後述する図８に示すＴＣＰパッシブオープン処理が実行される（ステップＳ５）．判断ステップＳ３で、ＴＣＰｓｅｃのパッシブオープンの許可状態が確認された場合は、同じく後述する図９に示すＴＣＰｓｅｃのパッシブオープン処理が実行される（ステップＳ６）。

【０１１５】

ステップＳ５又はステップＳ６におけるＴＣＰパッシブオープン処理又はＴＣＰｓｅｃのパッシブオープン処理が終了するとＴＣＰ／ＴＣＰｓｅｃパッシブオープン処理を終了する（ステップＳ７）。このように、本例では、上位であるアプリケーションからは、ＴＣＰでパッシブオープンを行っているが、ＴＣＰ２の判断により、ＴＣＰｓｅｃがサポートされていればＴＣＰｓｅｃで通信を行い、ＴＣＰｓｅｃがサポートされていなければＴＣＰで通信することとなる。

【０１１６】

次に、図６に基づいて本発明のＴＣＰ並びにＴＣＰｓｅｃのアクティブオープン処理について説明する。ＴＣＰ／ＴＣＰｓｅｃのアクティブオープンとは、接続要求のオープン

であり、例えば、Ｗｅｂブラウザを実装するクライアント端末が、この状態でオープンすることになる。図７でいうとホストＡ側の処理がこれに相当する。図６は、このアクティブオープンにおける処理の流れ図であり、上位アプリケーション・プログラムにおいて接続要求オープンがなされた場合に、このＴＣＰ／ＴＣＰｓｅｃのアクティブオープン処理が開始される（ステップＳ８）。

【０１１７】

まず、最初に、オープンするポート番号の解析がなされる（ステップＳ９）。この解析は、例えば、Ｗｅｂブラウザを実装するクライアント端末アプリケーションが、ＴＣＰポートの３０００番を使おうとした場合は、ＴＣＰポートの３０００番の定義状態を確認する。

【０１１８】

次にこのポート番号３０００に対してＴＣＰｓｅｃのアクティブオープンが許可されているかどうかが判断される（ステップＳ１０）。ステップＳ１０においてＴＣＰｓｅｃのアクティブオープンが許可されていないと判定された場合は、続いてＴＣＰアクティブオープンが許可されているかどうかが判断される（ステップＳ１１）。判断ステップＳ１１でＴＣＰアクティブオープンも許可されていない場合は、ＴＣＰｓｅｃ、ＴＣＰのいずれのアクティブオープンも許可されていないことになり、ＴＣＰ／ＴＣＰｓｅｃアクティブは失敗となり、接続処理は中断される（ステップＳ１４）。

【０１１９】

判断ステップＳ１１でＴＣＰアクティブオープンが許可されている場合、すなわちＴＣＰｓｅｃアクティブオープンは許可されていないがＴＣＰアクティブオープンは許可されているときは、後述する図１０に示すＴＣＰアクティブオープン処理が実行される（ステップＳ１２）。

【０１２０】

判断ステップＳ１０で、ＴＣＰｓｅｃのアクティブオープンの許可状態が確認された場合は、後述する図１１に示すＴＣＰｓｅｃのアクティブオープン処理が実行される（ステップＳ１３）。ステップＳ１２又はステップＳ１３におけるＴＣＰｓｅｃアクティブオープン処理又はＴＣＰｓｅｃのアクティブオープン処理が終了するとＴＣＰ／ＴＣＰｓｅｃアクティブオープン処理が終わる（ステップＳ１４）。ＴＣＰ／ＴＣＰｓｅｃアクティブオープンの場合も、ＴＣＰ／ＴＣＰｓｅｃパッシブオープン（図５）の場合と同様に、上位であるアプリケーションからは、ＴＣＰでアクティブブオープンを行っているが、ＴＣＰ２の判断により、ＴＣＰｓｅｃがサポートされていればＴＣＰｓｅｃで通信を行い、ＴＣＰｓｅｃがサポートされていなければＴＣＰで通信することとなる。

【０１２１】

次に、図７に基づいてアクティブオープン側のホストＡとパッシブオープン側のホストＢ間のシーケンス処理について、本発明のＴＣＰｓｅｃを使った通信の処理を説明する。

【０１２２】

図７は、本発明の暗号処理プロトコルであるＴＣＰｓｅｃを用いたときの接続シーケンス、データ通信シーケンス及び切断シーケンスを、標準のＴＣＰと比較して示したものである。図７（ａ）は標準ＴＣＰ，図７（ｂ）は本発明のＴＣＰｓｅｃを用いた時の通信のシーケンスを示した図である。

図７（ａ）に示すように、標準のＴＣＰは、ホストＢのアプリケーションがＴＣＰパッシブオープンし、ホストＡのアプリケーションがＴＣＰアクティブオープンしている。

【０１２３】

ホストＢのアプリケーションがＴＣＰパッシブオープンをすると、ＴＣＰパッシブオープン処理（図５のステップ５及び図８参照）を開始し、後述する図８のステップＳ１５に示すように受信待ちとなる。ホストＡのアプリケーションがＴＣＰアクティブオープンをすると、ＴＣＰアクティブオープン処理（図６のステップＳ１２及び図１０参照）を開始し、後述する図１０のステップＳ５２に示すようにホストＡからホストＢに対して接続要求（ＳＹＮ）が送信される。これにより、標準ＴＣＰの接続シーケンスが開始状態となる

。
【０１２４】
ホストＢ側では、この接続要求（ＳＹＮ）を受信するとこの接続要求の受信パケットの解析を終了し、ホストＡ側に接続応答（ＳＹＮ・ＡＣＫ）を送信する。ここでＡＣＫとは、Acknowledgementの略であり、データ転送が正常に終了したときなどに送信されるものである。ホストＡ側は、この接続応答（ＳＹＮ・ＡＣＫ）を受信すると、接続が完了した旨のＡＣＫ（肯定応答）を送信し、標準ＴＣＰの接続シーケンスを終了する。
【０１２５】
この標準ＴＣＰの接続シーケンスを終了すると、標準ＴＣＰによるデータ通信シーケンスが有効となり、ホストＡ側、又は、ホストＢ側の何れかがデータを送信後、データを受信した側からＡＣＫ（肯定応答）を返すという基本パターンを繰り返してデータの送受信が行われる。
【０１２６】
この標準ＴＣＰのデータ通信シーケンスでは、ホストＡ又はホストＢの何れかが、相手に対して切断要求をすることができる。
図７（ａ）では、アクティブオープン側のホストＡからパッシブオープン側のホストＢに対して切断要求が送信された場合を示している。ホストＡのアプリケーションから切断要求があると、ホストＡは、切断要求（ＦＩＮ）を送信する。ホストＢは、この切断要求（ＦＩＮ）を受信すると、後述する図８のステップＳ２３に示すように切断応答（ＦＩＮ・ＡＣＫ）を送信する。ホストＡは、この切断応答（ＦＩＮ・ＡＣＫ）を受信すると、ＡＣＫ（肯定応答）を送信し、標準ＴＣＰの切断シーケンスを終了する。
【０１２７】
次に、本発明のＴＣＰｓｅｃによる通信のシーケンスを図７（ｂ）に基づいて説明する。図７（ｂ）では、ホストＢのアプリケーションがＴＣＰｓｅｃパッシブオープンし、ホストＡのアプリケーションがＴＣＰｓｅｃアクティブオープンしている。
ホストＢのアプリケーションがＴＣＰｓｅｃパッシブオープンをすると、ＴＣＰｓｅｃパッシブオープン処理（図５のステップＳ６及び図９を参照）を開始し、後述する図９のステップＳ３１に示すように受信待ちとなる。ホストＡのアプリケーションがＴＣＰｓｅｃアクティブオープンをすると、ＴＣＰｓｅｃアクティブオープン処理（図６のステップＳ１３及び図１１を参照）を開始し、図１１のステップＳ６９に示すようにホストＡからホストＢに対して接続要求（ＳＹＮ）が送信される。これにより、ＴＣＰｓｅｃの接続シーケンスが開始状態となる。なお、接続要求（ＳＹＮ）には、オプションで、ＴＣＰ２の固有情報を暗号化して付加し、正しい相手であることを相手に通知するようにしている。すなわち、ホストＡとホストＢの間で次のＴＣＰｓｅｃネゴシエーションデータを交換する以前に相手方端末がＴＣＰ２をサポートする端末であるか否か、言い換えると通信する正しい相手であるか否かを確認することができる。
【０１２８】
ホストＢ側では、ホストＡから送信された接続要求（ＳＹＮ）を受信すると、正しい相手であれば、ホストＡに対して接続応答（ＳＹＮ・ＡＣＫ）を送信する。そして、ホストＡ側は、このホストＢからの接続応答（ＳＹＮ・ＡＣＫ）を受信するとＡＣＫ（肯定応答）を送信する。続いてホストＡとホストＢの間でＴＣＰｓｅｃネゴシエーションデータを交換し、正しい相手であれば、ＴＣＰｓｅｃの接続シーケンスを終了する。
【０１２９】
この接続シーケンスが終了すると、ＴＣＰｓｅｃのデータ通信シーケンスが有効となり、ホストＡ側又はホストＢ側の何れかがデータを送信後、データを受信した側からＡＣＫ（肯定応答）を返す基本パターンを繰り返してデータの送受信が行われる。なお、このデータは、すべて暗号データであることは言うまでもない。
【０１３０】
なお、ＴＣＰｓｅｃのデータ通信シーケンスでは、ホストＡ又はホストＢの何れかが相手方に対して切断要求をすることができる。図７（ｂ）では、アクティブオープン側のホ

ストＡから切断を開始している。ホストＡのアプリケーションから切断要求があると、ホストＡは切断要求（ＦＩＮ）を送信する。この切断要求（ＦＩＮ）には、オプションで、ＴＣＰ２の固有情報を暗号化して付加し、正しい相手であることを相手に通知することができるものである。ホストＢは、この切断要求（ＦＩＮ）を受信すると、正しい相手であれば、後述する図９のステップＳ４２に示すように切断応答（ＦＩＮ・ＡＣＫ）を送信する。ホストＡは、この切断応答（ＦＩＮ・ＡＣＫ）を受信すると、ＡＣＫ（肯定応答）を送信し、ＴＣＰｓｅｃの切断シーケンスを終了する。

【０１３１】

以上、図７に基づいて、標準ＴＣＰと本発明のＴＣＰ２のひとつであるＴＣＰｓｅｃについて通信の接続から切断までのシーケンスを説明したが、以下、ＴＣＰ及びＴＣＰｓｅｃのパッシブオープン処理、及びアクティブオープン処理について流れ図に従って順に説明する。

【０１３２】

まず、図５の流れ図のステップＳ５において、ＴＣＰパッシブオープン処理がスタートした場合の詳細について図８の流れ図に基づいて説明する。

図５のステップＳ５で処理するプロトコルがＴＣＰと決定した場合に、この図８のＴＣＰパッシブオープン処理がスタートする。最初に、受信待ちをして、受信した受信パケットの解析を行う（ステップＳ１５）。続いてこの受信したパケットが正しいパケットであるか否か、つまりＤｏＳ攻撃におけるＴＣＰプロトコル攻撃パターンであるかどうかを判断する（ステップＳ１６）。そしてステップＳ１６の判断の結果、不正パケットであると判定された場合にはその受信パケットを廃棄し（ステップＳ１７）、次のパケットの受信を待つ。

【０１３３】

判断ステップＳ１６において、受信パケットが正しいＴＣＰパケットであると判断された場合は、続いて接続中か否か、つまり図７のホストＡとホストＢの接続シーケンスが完了しているかどうかが判断される（ステップＳ１８）。判断ステップＳ１８において、接続中であると判定された場合は、次にパケットが切断要求（図７（ａ）のＦＩＮ）であるか否かが判断される（ステップＳ１９）。切断要求でなければ、続いて切断応答（図７（ａ）のＦＩＮ／ＡＣＫ）であるか否かが判断される（ステップＳ２０）。切断要求でもなく、切断応答でもない場合には、ＴＣＰデータの送受信処理が行われ（ステップＳ２１）、受信パケットが切断応答である場合は、図７のホストＡからＡＣＫが送信され、ＴＣＰ接続が切断される（ステップＳ２５）。判断ステップＳ１９でホストＡからの切断要求であると判断されると、これに対する切断応答がホストＢから送信される（ステップＳ２３）。

【０１３４】

ステップＳ２３で切断応答が送信された場合には、最終のＡＣＫを待つ（ステップＳ２４）。そして、最終ＡＣＫを受信した後にＴＣＰを切断状態にして（ステップＳ２５）、ＴＣＰパッシブオープンを終了する（ステップＳ２６）。

判断ステップＳ１８において、受信ポートが接続中でないとされた場合には、受信したパケットがパッシブオープン許可ポートであるか否かが判定される（ステップＳ２７）。そして受信パケットが許可されていない場合は、受信パケットを廃棄して（ステップＳ２８）次のパケットを待つ。また、判断ステップＳ２７において、受信パケットがパッシブオープン許可になっているとされた場合は、次にパケットが接続要求であるか否かが判断され（ステップＳ２９）、接続要求でない場合は、パケットを廃棄して（ステップＳ２８）次のパケットを待つ。また、判断ステップＳ２９で接続要求であると判断された場合には、接続応答を送信し、ＴＣＰを接続状態とする（ステップＳ３０）。

【０１３５】

次に、図５のＴＣＰｓｅｃのパッシブオープンにおける処理ステップＳ６の詳細について、図９の流れ図に基づいて説明する。この処理は、図５のステップＳ６に示されるように、受信パケットの処理がＴＣＰｓｅｃの処理であると決定された場合の処理である。最

初に、受信待ちをして、受信した受信パケットの解析がなされる（ステップＳ３１）。続いてこの受信したパケットが正しいパケットであるか否か、つまりＤｏＳ攻撃におけるＴＣＰプロトコル攻撃パターンでないかどうかが判断される（ステップＳ３２）。このステップＳ３２の判断の結果、不正パケットであると判定された場合にはその受信パケットを廃棄し（ステップＳ３３）、ステップＳ３１に戻り、次のパケットの受信を待つ。

【０１３６】

判断ステップＳ３２において、受信パケットが正しいパケットであると判断された場合は、続いてホストＡとホストＢの接続が完了しているかどうか（接続中かどうか）が判断される（ステップＳ３４）。判断ステップＳ３４において、ホストＡとホストＢが接続中であると判定された場合は、次に受信したパケットが切断要求（ＦＩＮ）であるのか否かが判断される（ステップＳ３５）。切断要求でなければ、今度は受信したパケットが切断応答（ＦＩＮ・ＡＣＫ）であるか否かが判断される（ステップＳ３６）。そして、受信したパケットが切断要求でもなく、切断応答でもないということであれば、後述する図１２に示されるＴＣＰｓｅｃデータの送受信処理が行われ（ステップＳ３７）、ステップＳ４９に進む。次に、判断ステップＳ３６で切断応答があった場合には、切断鍵が一致しているかどうかが判断される（ステップＳ３８）。ここで、切断鍵とはホストＡとホストＢの間で図７の接続シーケンスにおけるネゴシエーションにおいて取り交わした共通鍵（秘密鍵）であり、この鍵が一致したときだけ両者間の通信を切断することができるものである。判断ステップＳ３８で切断鍵が一致した場合には、ＡＣＫを送信して（ステップＳ３９）、ホストＡとホストＢ間のＴＣＰｓｅｃを切断する（ステップＳ４４）。判断ステップＳ３８で切断鍵が一致しなかった場合には、不正パケットとしてパケットを廃棄し（ステップＳ４１）、次の受信パケットを待つ。また、判断ステップＳ３５において、受信パケットが切断要求（ＦＩＮ）であると判断された場合も、切断鍵が一致しているか否かが判断される（ステップＳ４０）。そして、切断鍵が一致しない場合は、受信パケットが不正なパケットとして廃棄され（ステップＳ４１）、切断鍵が一致した場合は、切断応答（ＦＩＮ・ＡＣＫ）の送信が行われる（ステップＳ４２）。ステップＳ４２で切断応答が送信された場合には、相手方からの最終のＡＣＫを待ち（ステップＳ４３）、この最終ＡＣＫを受信するとＴＣＰｓｅｃを切断状態にして（ステップＳ４４）、ＴＣＰｓｅｃパッシブオープンを終了する（ステップＳ４５）。

【０１３７】

判断ステップＳ３４において、ホストＡとホストＢが接続中でないと判断された場合には、受信したパケットがパッシブオープンの許可ポートであるか否かが判断される（ステップＳ４６）。そして受信パケットがパッシブオープンの許可ポートではない場合は、受信パケットを廃棄して（ステップＳ４７）、ステップＳ３１に戻り次のパケットを待つ。また、判断ステップＳ４６おいて、受信パケットがパッシブオープン許可ポートになっているとされた場合は、後述する図１３に示すＴＰＣｓｅｃパッシブ接続処理が実行される（ステップＳ４８）。

【０１３８】

続いて、共通鍵と認証データに基づいて通信相手が正常、つまり正当な権限を持った相手であるか否かが判断される（ステップＳ４９）。正常な相手であると判断されれば、ステップＳ３１に戻り、次の受信パケットを待つが、通信相手が正常の相手ではないと判断されると、ＴＰＣｓｅｃの接続を強制切断し（ステップＳ５０）、ＴＣＰｓｅｃのパッシブオープンの処理を終了する（ステップＳ５１）。

【０１３９】

次に、図６のステップＳ１２に示す、ＴＣＰアクティブオープンの処理について図１０の流れ図に基づいて説明する。

図１０は、図６における処理するプロトコルがＴＣＰであると決定した場合の処理を示す図であり、最初に、送信側ホストＡから受信側ホストＢに対して接続要求（ＳＹＮ）が送信される（ステップＳ５２）。この接続要求に対して受信側ホストＢから接続応答（ＳＹＮ・ＡＣＫ）が送信されると、次に受信待ちをし、受信したパケットの解析が行われる

（ステップＳ５３）。次に、この受信したパケットが正しいパケットであるか否か、つまりＤｏＳ攻撃におけるＴＣＰプロトコル攻撃パターンでないかどうかを判断する（ステップＳ５４）。このステップＳ５４における判断の結果、不正パケットであると判定された場合にはその受信パケットを廃棄し（ステップＳ５５）、ステップＳ５３に戻り、次のパケットの受信を待つ。

【０１４０】

判断ステップＳ５４において、受信パケットが正しいパケットであると判断された場合は、続いて送信側（アクティブ側）ホストＡと受信側（パッシブ側）ホストＢが、接続中かどうかが判断される（ステップＳ５６）。この判断ステップＳ５６において、接続中であると判定された場合は、次に受信パケットが送信側ホストＡから受信側ホストＢに対しての切断要求であるか否かが判断される（ステップＳ５７）。これが切断要求でなければ、今度は受信側ホストＢから送信側ホストＡに対する切断応答（ＦＩＮ・ＡＣＫ）であるか否かが判断される（ステップＳ５８）。切断要求でもなく切断応答でもないということになれば、ＴＣＰデータの送受信処理が行われ（ステップＳ５９）、次の受信パケットを待つ。判断ステップＳ５８でホストＢからホストＡへの切断応答であると判断されると、ホストＡは切断を肯定するＡＣＫを送信し（ステップＳ６０）、ＴＣＰを切断する（ステップＳ６３）。

【０１４１】

判断ステップＳ５７において、受信したパケットが切断要求である場合は、ホストＢからホストＡに対して切断応答が送信され（ステップＳ６１）、ホストＢはホストＡからの最終のＡＣＫの受信を待つ（ステップＳ６２）。そして、ホストＢがホストＡから最終ＡＣＫを受信した後にＴＣＰを切断状態にして（ステップＳ６３）、ＴＣＰアクティブオープンを終了する（ステップＳ６４）。

判断ステップＳ５６において、送信側ホストＡと受信側ホストＢが接続中でないとされた場合には、受信したパケットがアクティブオープン許可ポートであるか否かが判定される（ステップＳ６５）。そして受信パケットが許可されていない場合は、受信パケットを廃棄して（ステップＳ６６）次のパケットを待つ。また、判断ステップＳ６５において、受信パケットがアクティブオープン許可になっているとされた場合は、次に受信側ホストＢからの接続応答があったか否かが判断され（ステップＳ６７）、接続応答がなければ、パケットを廃棄して（ステップＳ６６）次のパケットを待ち、受信側ホストＢから接続応答がなされた場合には、ＴＣＰの接続状態として（ステップＳ６８）、ステップＳ５３に戻り、次の受信パケットを待つ。

【０１４２】

次に、図６のステップＳ１３のＴＣＰｓｅｃアクティブオープンが開始された場合の詳細な処理について図１１の流れ図に基づいて説明する。

図１１の流れ図に示す処理は、図６のステップＳ１３で処理するプロトコルがＴＣＰｓｅｃであると決定した場合に開始されるものである。最初に、送信側ホストＡから受信側ホストＢに対して接続要求（ＳＹＮ）が送信される（ステップＳ６９）。これに対して、受信側ホストＢから接続応答（ＳＹＮ・ＡＣＫ）があってパケットの受信が開始され、受信したパケットの解析が行われる（ステップＳ７０）。

【０１４３】

次に、受信パケットの解析の結果、受信したパケットが正しいＴＣＰのパケットであるか否か、すなわち、ＤｏＳ攻撃におけるＴＣＰプロトコル攻撃パターンでないか否かが判断される（ステップＳ７１）。この結果、不正パケットであると判定された場合には、そのパケットを廃棄（ステップＳ７２）し、ステップＳ７０に戻り次のパケットを待つ。

【０１４４】

判断ステップＳ７１において、受信したパケットが正しいＴＣＰパケットであると判定された場合は、次に送信側ホストＡと受信側ホストＢの接続が完了しているかどうか（接続中かどうか）が判断される（ステップＳ７３）。そしてホストＡとホストＢが接続中であれば、今度は受信したパケットが切断要求（ＦＩＮ）であるか否かが判断される（ステ

ップＳ７４）。受信したパケットが切断要求ではないときは、次に受信側ホストＢから切断応答があるかどうかが判断される（ステップＳ７５）。切断要求もなく切断応答もない場合は、図１２に示すＴＣＰｓｅｃデータの送受信処理が行われ（ステップ７６）、その後ステップＳ８９に進む。

【０１４５】

判断ステップＳ７５で切断応答があった場合には、切断鍵が一致しているかどうかが判断される（ステップＳ７７）。この切断鍵については図９において説明したとおりである。判断ステップＳ７７で切断鍵が一致した場合には、送信側ホストＡから受信側ホストＢに対してＡＣＫを送信して（ステップＳ７８）、ホストＡとホストＢ間のＴＣＰｓｅｃを切断する（ステップＳ８３）。判断ステップＳ７７で切断鍵が一致しなかった場合には、不正パケットとしてパケットを廃棄し（ステップＳ８０）、次の受信パケットを待つ。また、判断ステップＳ７４において、受信パケットが切断要求（ＦＩＮ）であると判断された場合も、切断鍵が一致しているか否かが判断される（ステップＳ７９）。そして、切断鍵が一致しない場合は、受信パケットが不正なパケットとして廃棄され（ステップＳ８０）、切断鍵が一致した場合は、切断応答（ＦＩＮ・ＡＣＫ）の送信が行われる（ステップＳ８１）。ステップＳ８１で切断応答が送信された場合には、相手方からの最終のＡＣＫを待ち（ステップＳ８２）、この最終ＡＣＫを受信するとＴＣＰｓｅｃを切断状態にして（ステップＳ８３）、ＴＣＰｓｅｃアクティブオープンを終了する（ステップＳ８４）。

【０１４６】

判断ステップＳ７３において、送信側ホストＡと受信側ホストＢの接続が完了していない、つまり接続中でないとされた場合には、受信したパケットがアクティブオープン許可ポートであるか否かが判定される（ステップＳ８５）。そして受信されたパケットが許可されていない場合は、その受信パケットを廃棄して（ステップＳ８７）ステップＳ７０に戻り、次のパケットを待つ。また、判断ステップＳ８５において、受信パケットがアクティブオープン許可になっているとされた場合は、受信されるパケットが受信側ホストＢからの接続応答（ＳＹＮ・ＡＣＫ）のパケットであるか否かが判断され（ステップＳ８６）、接続応答のパケットでない場合は、パケットを廃棄して（ステップＳ８７）次のパケットを待ち、判断ステップＳ８６で接続応答のパケットであると判断された場合には、図１４でその詳細を示すＴＣＰｓｅｃアクティブ接続処理が行われる（ステップＳ８８）。

【０１４７】

ステップＳ８８でＴＣＰｓｅｃのアクティブ接続処理がなされると、次に受信側のホストＢが正常な相手か否か、つまり接続を許可されている相手であるか否かが判断される（ステップＳ８９）。そして、接続が許されている相手であると判断されれば、ステップＳ７０に戻って次のパケットの受信を待ち、ステップＳ８９で接続が許可されていない相手であると判断されると、ＴＣＰｓｅｃによる送受信を強制的に切断して（ステップＳ９０）、ＴＣＰｓｅｃのアクティブオープンを終了する（ステップＳ９１）。

【０１４８】

次に、上述した図９ステップＳ３７及び図１１のステップＳ７６が選択された場合のＴＣＰｓｅｃデータの送受信処理の詳細について図１２の流れ図に基づいて説明する。

【０１４９】

まず、図９のステップＳ３７及び図１１のステップＳ７６でＴＣＰｓｅｃデータの送受信処理が開始すると、最初に、ホストＡの上位アプリケーションからの送信要求があるか否かが判断される（ステップＳ９２）。そして、ホストＡの上位アプリケーションから送信要求があった場合は、送信側ホストＡで送信データが暗号化され（ステップＳ９３）、それに認証データが付加されて（ステップＳ９４）、受信側ホストＢに暗号化され認証データが付加されたパケットが送信される（ステップＳ９５）。

【０１５０】

次に、受信側ホストＢで、受信データがあるか否かが判断され（ステップＳ９６）、受信データがある場合には、受信データの復号化が行われる（ステップＳ９７）。次に、この受信され復号化されたデータが正しいデータであるかどうかが判断される（ステップＳ

９８）。この判断は、復号化したデータと受信された認証データとを確認することによって行われるが、復号データを確認した結果、正しいデータでないと判定された場合には、ＴＣＰ／ＴＣＰｓｅｃを強制的に切断する（ステップＳ９９）。この強制切断は、受信したデータを廃棄するとともに、送信側に切断要求をすることによって行われる。判断ステップＳ９８で、復号化したデータが正しいデータであると判定された場合には、受信データの取り込み、すなわち上位プロトコルスタックへのデータの配達が行われ（ステップＳ１００）、ＴＣＰｓｅｃのデータ送受信処理が完了する（ステップＳ１０１）。

【０１５１】

次に、図９のステップＳ４８でＴＣＰｓｅｃパッシブ接続処理が開始された場合の詳細の処理を図１３の流れ図に基づいて説明する。

最初に、相手が正しい相手、つまり自コンピュータに接続する権限を持つコンピュータであるか否かを判断し（ステップＳ１０２）、正しい相手でない場合にはＴＣＰｓｅｃの強制切断のための処理を実施する（ステップＳ１０３）。判断ステップＳ１０２において接続相手が正しい相手であると判断されれば、受信側ホストＢから接続応答を送信する（ステップＳ１０４）。

【０１５２】

そして、接続応答を送信してきた相手の情報が自コンピュータ内にあるかどうかを確認する（ステップＳ１０５）。相手情報がコンピュータ内にない場合は、本システム、すなわちＴＣＰ２をインストールする際に使用した、インストールサーバから相手情報を取得する（ステップＳ１０６）。または、第三者認証のサーバから相手情報を取得してステップＳ１０７に進む。この取得する情報としては、相手側のＴＣＰ２のＩＤ、ユーザＩＤ、パスワード、バイオメトリックス情報、機器固有情報、ＬＡＮ接続機器情報等の内、１つ、若しくは、複数を使用することができる。なお、サーバからの取得情報を既に自コンピュータが保有している場合であっても、有効期限若しくは有効使用回数を超えているような場合には、改めて取得動作を行う必要がある。

【０１５３】

次に、相手情報が正しい相手であるか否か、つまり、自分のコンピュータにアクセスすることを許容されている相手であるかどうかが判断される（ステップＳ１０７）。ここで、接続する相手が正しい相手であれば、ＴＣＰｓｅｃのパッシブ接続を完了する（ステップＳ１０８）が、正しい相手でない場合にはＴＣＰｓｅｃの強制切断を行って接続を中止する（ステップＳ１０３）。

【０１５４】

次に、図１１のステップＳ８８でＴＣＰｓｅｃのアクティブ接続処理が開始された場合の詳細の処理を図１４の流れ図に基づいて説明する。

図１３のパッシブ接続処理の場合と同様に、最初に、接続要求をしてきた相手が正しい相手であるどうか、つまり自コンピュータにアクセス権限を持っている相手からの通信であるかどうかを判断する（ステップＳ１０９）。正当なアクセス権限を持たない相手からの通信であれば、ＴＣＰｓｅｃのアクティブ接続を強制切断して終了する（ステップＳ１１０）。

【０１５５】

判断ステップＳ１０９で正しい相手であると判定されれば、送信側ホストから受信側ホストＢに対して肯定的な接続応答（ＡＣＫ）を送信する（ステップＳ１１１）。

次に、自コンピュータが相手側の情報を保有しているかどうかが判断される（ステップＳ１１２）。相手情報がコンピュータ内にない場合は、本システム、すなわちＴＣＰ２をインストールする際に使用した、インストールサーバから相手情報を取得する（ステップＳ１１３）。または、第三者認証のサーバから相手情報を取得してステップＳ１１４に進む。ここで、この取得する情報は、図１３ステップＳ１０６と同様に、相手側のＴＣＰ２のＩＤ、ユーザＩＤ、パスワード、バイオメトリックス情報、機器固有情報、ＬＡＮ接続機器情報等の内、１つ、若しくは、複数とすることができる。なお、サーバからの取得情報を既に自コンピュータが保有していたとしても、有効期限若しくは有効使用回数を超え

ている場合には、改めて取得動作を行う必要がある。

【０１５６】

次に、相手情報が正しい相手であるか否か、つまり、自分のコンピュータにアクセスすることを許容されている相手であるかどうかが判断される（ステップＳ１１４）。接続する相手が正しい相手であれば、ＴＣＰｓｅｃのアクティブ接続を完了する（ステップＳ１１５）が、正しい相手でない場合にはＴＣＰｓｅｃの強制切断を行って接続を中止する（ステップＳ１１０）。

【０１５７】

以上、本発明のＴＣＰ２のうち、ＴＣＰ／ＴＣＰｓｅｃを用いたパッシブオープン及びアクティブオープンの通信処理について説明した。

次に、図３で示すような、本発明の第２の実施形態であるＵＤＰ／ＵＤＰｓｅｃを用いた通信システム及び通信方法について説明する。

【０１５８】

図１５は、本発明の第２の実施の形態に用いられるＵＤＰ／ＵＤＰｓｅｃのパッシブオープン処理について説明するための流れ図である。

この処理は、上位アプリケーション・プログラムにより開始される（ステップ１２０）。最初に、オープンするポート番号の解析、すなわちポート番号の定義状態が確認される（ステップ１２１）。次に、このポート番号がＵＤＰｓｅｃオープンになっているか否かが判断される（ステップＳ１２２）。ＵＤＰｓｅｃがオープンになっていない場合はＵＤＰがオープンになっているかどうかが判断される（ステップ１２３）。そして、ＵＤＰｓｅｃ、ＵＤＰが両方ともオープン許可になっていない場合は、ＵＤＰ／ＵＤＰｓｅｃは終了する（ステップＳ１２６）。判断ステップＳ１２３で、ＵＤＰがオープン許可になっている場合、つまりＵＤＰｓｅｃはオープン許可になっていないけれどもＵＤＰがオープン許可になっている場合は、図１８に示すＵＤＰオープン処理が実施される（ステップＳ１２４）また、判断ステップＳ１２２においてＵＤＰｓｅｃがオープンである場合は、ＵＤＰがオープンであるか否かにかかわらず、ＵＤＰｓｅｃのオープン処理が実施されて（ステップＳ１２５）、ＵＤＰ／ＵＤＰｓｅｃオープン処理は終了する（ステップＳ１２６）。なお、上位であるアプリケーションからは、ＵＤＰでオープンを行っていたとしても、ＴＣＰ２の判断により、ＵＤＰｓｅｃ若しくは、ＵＤＰで通信することも可能である。

【０１５９】

次に、本発明の第２実施の形態の１つであるＵＤＰ／ＵＤＰｓｅｃを用いたユニキャスト通信におけるシーケンス処理について図１６に基づいて説明する。

図１６は、標準のＵＤＰ、及び、本発明のＴＣＰ２の中のＵＤＰｓｅｃにおけるユニキャスト通信の開始シーケンス、データ通信シーケンスのパケット（ヘッダと、ペイロードで構成する）及び、その流れを説明した図である。

【０１６０】

図１６（ａ）が標準のＵＤＰを用いた通信シーケンスを示し、図１６（ｂ）は、ＵＤＰｓｅｃによる暗号化通信のシーケンスを示す。

図１６（ａ）の標準のＵＤＰは、ホストＡ、ホストＢともにアプリケーションがＵＤＰオープンしている例を示している。ホストＢのアプリケーションがＵＤＰオープンをすると、ＵＤＰオープン処理（図１５のステップＳ１２４及び図１８参照）を開始する。また、同様にホストＡのアプリケーションがＵＤＰオープンした場合も上記のＵＤＰオープン処理を開始する。これにより、ＵＤＰのデータ通信を行うことが可能となる。ここで図１６（ａ）に示すユニキャスト通信では、ホストＡ、ホストＢの何れからもデータの送信が可能である。

【０１６１】

次に、本発明のＴＣＰ２の暗号化方式の１つであるＵＤＰｓｅｃによる通信処理のシーケンスを説明する。

図１６（ｂ）は、本発明のＴＣＰ２が持つＵＤＰｓｅｃにより暗号化通信する場合の例であるが、この例は、ホストＡ、ホストＢともにアプリケーションがＵＤＰオープンし、

ＴＣＰ２がＵＤＰｓｅｃでオープンと判断したケースである。
ホストＢがＵＤＰｓｅｃオープンをすると、ＵＤＰｓｅｃオープン処理（図１５のステップＳ１２５及び図１９を参照）が開始される。また、ホストＡがＵＤＰｓｅｃオープンした場合も同様にＵＤＰｓｅｃオープン処理が開始される。そして、ＵＤＰｓｅｃのデータ通信が実現可能となる。

【０１６２】

この図１６（ｂ）に示したＵＤＰｓｅｃを用いたユニキャスト通信においても、ＵＤＰのときと同様に、ホストＡ側又はホストＢ側の何れからもデータを送信することができる。図１６（ｂ）の場合、まず、ホストＡのアプリケーションからＵＤＰデータの送信要求があったとして説明する。アプリケーションからＵＤＰデータの送信要求を受け取ると、ホストＢは、ＵＤＰｓｅｃユニキャスト受信開始処理を開始し、ネゴシエーションを開始する。ネゴシエーションをして、相手が正しい相手であれば、ネゴシエーションを完了して、アプリケーションからＵＤＰデータの送信要求をＵＤＰｓｅｃデータ（暗号データ）として送信する。このＵＤＰｓｅｃユニキャスト通信では、データを受信した側からＡＣＫ（肯定応答）を返さない。このため、送達確認とデータの保証をする機能はないが、その分データの通信が高速になり、大容量の画像データなどの通信に適している。

【０１６３】

図１７は、標準ＵＤＰと本発明のＴＣＰ２の暗号化方式であるＵＤＰｓｅｃを用いたブロードキャスト通信の開始シーケンス、データ通信シーケンスのパケット（ヘッダと、ペイロードで構成する）及び、その流れを説明した図である。

図１７（ａ）が、標準のＵＤＰ、図１７（ｂ）が本発明のＴＣＰ２のＵＤＰｓｅｃによる通信のシーケンス図である。

【０１６４】

図１７（ａ）の標準のＵＤＰは、ホストＡ、ホストＢともにアプリケーションがＵＤＰオープンしている。そして、ホストＢのアプリケーションがＵＤＰオープンをすると、ＵＤＰオープン処理（図１５のステップＳ１２４及び図１８参照）を開始する。また、ホストＡのアプリケーションがＵＤＰオープンした場合も、同様にＵＤＰオープン処理を開始する。これにより、ＵＤＰのデータ通信を行うことができる状態となる。

【０１６５】

また、データはホストＡ、ホストＢの何れも発生することはできるが、図１７（ａ）では、ブロードキャスト通信ということもあって、ホストＡ側からホストＢ側に一方向的にデータが流れる図としている。データを受信したホストＢ側からＡＣＫ（肯定応答）を返さないため、送達確認とデータの保証をする機能は持たない。なお、データをブロードキャストする場合には、ＩＰアドレスのサブネットアドレスをオール１にすることで、データをブロードキャストすることが可能となる。

【０１６６】

次に、図１７（ｂ）のＵＤＰｓｅｃによる暗号化通信について説明する。この場合も、ホストＡ、ホストＢともにアプリケーションがＵＤＰオープンし、ＴＣＰ２がＵＤＰｓｅｃでオープンとしている。
ホストＢがＵＤＰｓｅｃオープンをすると、ＵＤＰｓｅｃオープン処理（図１５のステップＳ１２５及び図１９）を開始する。また、ホストＡがＵＤＰｓｅｃオープンしても同様に、ＵＤＰｓｅｃオープン処理を開始する。これにより、ＵＤＰｓｅｃのデータ通信を行うことができる状態となる。

【０１６７】

図１７（ｂ）示すように、ホストＡのアプリケーションからＵＤＰのブロードキャストデータ（ＩＰアドレスがブロードキャストを示しているデータ）の送信要求があった場合を説明する。アプリケーションからＵＤＰのブロードキャストデータの送信要求を受け取ると、ネゴをしないで、ＵＤＰｓｅｃで暗号データとして不特定ホストに配信する。ホストＢは、ブロードキャストデータを受け取ると、後述する図１９のステップＳ１４１のＵＤＰｓｅｃブロードキャスト受信開始処理を開始する。ホストＡとホストＢ間でネゴシエ

ーションを開始し、相手が正しい相手であれば、ネゴシエーションを完了して、ブロードキャストデータを復号化して、アプリケーションへ送る。このとき、データを受信した側からＡＣＫ（肯定応答）を返さないため、送達確認とデータの保証をする機能はない。

【０１６８】

次に、図１８に基づいて、図１５のステップＳ１２４の標準ＵＤＰのオープン処理について説明する。

図１８は、ＵＤＰのオープン処理の流れ図であり、この処理は図１５のステップＳ１２４で、処理するプロトコルがＵＤＰと決定した場合に開始される処理である。

【０１６９】

最初に、アプリケーションからの送信要求、又は受信パケットを待ち、送信要求又はパケットを受信したときに、パケットの解析を行う（ステップＳ１２７）。ここで、受信パケットだけでなく送信要求も解析するのは、悪意を持った第三者が送信するホストＡ踏み台にして、ホストＡを加害者として不特定多数に通信することを防ぐためである。この送受信パケットの解析を行った後に、正しいパケットであるかどうか、つまりＤｏＳ攻撃におけるＵＤＰプロトコル攻撃パターンでないかどうかを判断する（ステップＳ１２８）。この判断ステップＳ１２８において、不正パケットであると判定された場合には、パケットを廃棄し（ステップＳ１２９）、次のパケットを待つ。

【０１７０】

判断ステップＳ１２８で正しいパケットであると判定された場合は、ＵＤＰデータの送受信処理が行われ（ステップＳ１３０）、続いて上位アプリケーションからＵＤＰのクローズ要求があったかどうかが判断される（ステップＳ１３１）。上位アプリケーションよりＵＤＰクローズ要求があった場合には、ＵＤＰオープン処理を終了する（ステップＳ１３２）。

【０１７１】

次に、図１９に基づいて、図１５のステップＳ１２５のＵＤＰｓｅｃのオープン処理について説明する。図１９は、ＵＤＰｓｅｃのオープンにおける処理の流れ図であり、図１５のステップＳ１２５に示すように、処理するプロトコルがＵＤＰｓｅｃと決定した場合にこの処理が開始される。

【０１７２】

最初に、アプリケーションからの送信要求又は受信パケットを待ち、送信要求又は受信したパケットの解析が行われる（ステップＳ１３３）。次に、上位アプリケーションからの送信要求あるいは送受信パケットが正しいＵＤＰパケットであるか否か、つまりＤｏＳ攻撃におけるＴＣＰプロトコル攻撃パターンでないかどうかを判断する（ステップＳ１３４）。判断ステップＳ１３４において正しいＵＤＰパケットではないと判断された場合は、パケットを廃棄し（ステップＳ１３５）、次のパケットを待つ。

【０１７３】

判断ステップＳ１３４において、正しいＵＤＰパケットであると判定された場合は、次にＵＤＰｓｅｃネゴシエーションがなされた受信パケットであるか否かが判断される（ステップＳ１３６）。

【０１７４】

そして、この結果、ＵＤＰｓｅｃのネゴシエーションパケットであると判断された場合は、後述する図２３で示すＵＤＰｓｅｃユニキャスト受信開始処理が行われ（ステップＳ１３７）、ステップＳ１４７へ進む。

また、判断ステップＳ１３６において、ＵＤＰｓｅｃのネゴシエーションパケットではないと判断されると、続いてブロードキャスト通信であるか否かを判断する（ステップＳ１３８）。そして、ブロードキャスト通信であると判定された場合は、通信の開始パケットであるか、つまりオープン後の最初の通信パケットであるか否かを判断し（ステップＳ１３９）、開始パケットでない場合は図２２で説明するＵＤＰｓｅｃデータ送受信処理を行う（ステップＳ１４４）。判断ステップＳ１３９において通信の開始パケットであると判定された場合は、次に送信パケットであるか否かが判断される（ステップＳ１４０）。

そして、この結果、送信パケットであれば、上述したＵＤＰｓｅｃデータ送受信処理が行われる（ステップＳ１４４）が、送信パケットではないと判定された場合は、後述する図２０のＵＤＰｓｅｃブロードキャスト受信開始処理が実施される（ステップＳ１４１）。この受信開始処理の後で、送信されたパケットが正しい相手からのものであるかどうかを判断する（ステップＳ１４２）。そして、判断ステップＳ１４２で、送信されたパケットが正しい相手からのものではないと判断されると、パケットを廃棄し（ステップＳ１４３）、次のパケットを待つ。判断ステップＳ１４２で、正しい相手であると判定された場合には、図２２で示すＵＤＰｓｅｃデータ送受信処理が行われる（ステップＳ１４４）。

【０１７５】

また、判断ステップＳ１３８において、ブロードキャスト通信でない、すなわちユニキャスト通信であると判定された場合は、通信の開始パケット、すなわちオープン後の最初の通信パケットであるか否かが判断される（ステップＳ１４５）。この結果、開始パケットではないと判断された場合には、図２２で詳述するＵＤＰｓｅｃデータ送受信処理がなされる（ステップＳ１４４）。

【０１７６】

また、判断ステップＳ１４５で、オープン後の最初の通信パケットであると判断された場合は、図２１で後述するＵＤＰｓｅｃユニキャスト送信開始処理が行われる（ステップＳ１４６）。その後、通信の相手が、正しい相手であるかを判断し（ステップＳ１４７）、正しい相手である場合には、引き続きＵＤＰｓｅｃデータ送受信処理がなされ（ステップＳ１４４）、正しい相手ではないと判定された場合には、受信したパケットを廃棄し（ステップＳ１４８）、ステップＳ１３３に戻って次のパケットを待つ。

【０１７７】

次に、図１９のステップＳ１４１のＵＤＰｓｅｃブロードキャスト受信の開始における処理について、図２０に示す流れ図に基づいて説明する。

最初に、ブロードキャストを配信してきた相手の情報を自コンピュータが保有しているか否かを判断する（ステップＳ１４９）。そして、情報を保有していない場合には、本システムをインストールする際に使用した、インストールサーバから相手情報を取得する（ステップＳ１５０）。または、第三者認証のサーバから情報を取得する。この取得する情報は、相手のＴＣＰ２のＩＤ、ユーザＩＤ、パスワード、バイオメトリックス情報、機器固有情報、ＬＡＮ接続機器情報等の内、１つ、若しくは、複数を使用する。次に、ブロードキャストを配信してきた相手が正しい相手であるかどうかを判断する（ステップＳ１５１）。そして、正しい相手であると判断されれば、ＵＤＰｓｅｃでの受信が可能であることになり、ＵＤＰｓｅｃブロードキャストの通信開始処理を終了し（ステップＳ１５３）、受信可能であることを図１９のステップＳ１４２へ指示する。判断ステップＳ１５１で正しい相手ではないとされた場合には、通信の拒否を行い（ステップＳ１５２）、データを取得しない旨を同じく図１９のステップＳ１４２へ指示する。なお、ステップＳ１４９で仮に相手方に関する取得情報があったとしても有効期限、若しくは、有効使用回数を超えている場合には、改めてステップＳ１５０で相手情報の取得動作を行ったほうがよい。

【０１７８】

次に、図１９のステップＳ１４６のＵＤＰｓｅｃユニキャスト送信開始処理について、図２１に示す流れ図に基づいて説明する。

最初に、送信する相手の情報を自コンピュータが保有しているか否かを確認する（ステップＳ１５４）。そして、情報を保有していない場合には、図２０のステップＳ１５０と同様な方法で相手情報を取得する（ステップＳ１５５）。この取得する情報も図２０の場合と同じである。

【０１７９】

次に、送信する相手が正しい相手であるかどうかを判断する（ステップＳ１５６）。そして、正しい相手であると判断されれば、ＵＤＰｓｅｃでの送信が可能であることになり、ＵＤＰｓｅｃユニキャストの通信開始処理を終了し（ステップＳ１５８）、送信可能であることを図１９のステップＳ１４７へ指示する。判断ステップＳ１５６で正しい相手で

はないとされた場合には、通信の拒否を行い（ステップＳ１５７）、データを取得しない旨を図１９のステップＳ１４７へ指示する。

【０１８０】

次に、図２２に基づいて、図１９のステップＳ１４４に示すＵＤＰｓｅｃデータの送受信処理について説明する。

最初に、ホストＡのアプリケーションから送信要求があったか否かを判断する（ステップＳ１５９）。送信要求があれば送信側ホストＡにおいてデータを暗号化し（ステップＳ１６０）、この暗号化したデータに認証データが付加されて（ステップＳ１６１）、受信側ホストＢに暗号化され認証データが付加されたパケットが送信される（ステップＳ１６２）。

【０１８１】

次に、受信側ホストＢで、受信データがあるか否かが判断され（ステップＳ１６３）、受信データがある場合には、受信データの復号化が行われる（ステップＳ１６４）。次に、この受信され復号化されたデータが正しいデータであるかどうかが判断される（ステップＳ１６５）。この判断は、復号化したデータと受信された認証データとを確認することによって行われるが、復号データを確認した結果、正しいデータでないと判定された場合には、ＵＤＰ／ＵＤＰｓｅｃを強制的に切断する（ステップＳ１６６）。判断ステップＳ１６５で、復号化したデータが正しいデータであると判定された場合には、受信データの取り込み、すなわち上位プロトコルスタックへのデータの配達が行われ（ステップＳ１６７）、ＵＤＰｓｅｃのデータ送受信処理が完了する（ステップＳ１６８）。

【０１８２】

次に、図２３の流れ図に基づいて、図１９のステップＳ１３７に示すＵＤＰｓｅｃユニキャスト受信の開始処理について説明する。

最初に、ユニキャストで受信したパケットの相手情報を自コンピュータが保有しているか否かを判断する（ステップＳ１６９）。相手情報を持っていない場合には、本システムをインストールする際に使用した、インストールサーバあるいは第三者認証のサーバから相手情報を取得する（ステップＳ１７０）。取得する情報としては、図２０のステップＳ１５０あるいは図２１のステップＳ１５５と同じであり、相手のＴＣＰ２のＩＤ、ユーザＩＤ、パスワード、バイオメトリックス情報、機器固有情報、ＬＡＮ接続機器情報等の内、１つ、若しくは、複数がこれに相当する。

【０１８３】

次に、ユニキャストを送信してきた相手が正しい相手であるか否かを判断する（ステップＳ１７１）。正しい相手であると判定されれば、ＵＤＰｓｅｃでの受信が可能であることを図１９のステップＳ１４７へ伝えてＵＤＰｓｅｃブロードキャスト通信開始処理を終了する（ステップＳ１７３）。判断ステップＳ１７１で正しい相手ではないと判断された場合には、データを取得しない旨を図１９のステップＳ１４７へ伝え、通信を拒否する（ステップＳ１７２）。

【０１８４】

以上、本発明の第１の実施形態であるＴＣＰｓｅｃを用いた暗号化処理及び本発明の第２の実施形態であるＵＤＰｓｅｃを用いた暗号化処理について流れ図及びシーケンス図に基づいて詳しく説明した。

【０１８５】

次に、本発明のＴＣＰ２が、従来のＩＰｓｅｃあるいはＳＳＬと比べて如何に優位であるかという点について、表２及び図２４に基づいて説明する。

表２は、表１のＩＰｓｅｃとＳＳＬの機能比較表にＴＣＰ２の機能を追加して示したものである。

【０１８６】

この表２から明らかなように、ＩＰｓｅｃ及びＳＳＬが持っている様々な問題点（これらについては背景技術において示した）を、ＴＣＰ２を採用することによりことごとく解決していることが分かる。

例えば、ＳＳＬでは対応が困難であった、クライアントークライアント間の通信、ＴＣＰ／ＩＰプロトコルへのＤｏＳ攻撃、全てのＵＤＰポートあるいはＴＣＰポートのセキュアな通信、ソケットプログラムを変更しなければならなかったアプリケーションでの制限などに、ＴＣＰ２は完全に対応している。

【０１８７】

また、ＩＰｓｅｃでは対応が困難であった、エラーが多発する劣悪な環境下での通信、異なったＬＡＮ間での通信、複数キャリア経由の接続、ＰＰＰモバイル環境、ＡＤＳＬ環境での通信に対しても、ＴＣＰ２は完全にサポートしている。

さらに、モバイル環境下やＡＤＳＬ環境下でＶｏＩＰ（Voice over Internet Protocol）を使ったインターネットに対しては、ＩＰｓｅｃ及びＳＳＬとも表１及び表２に示すように問題があるが、本発明のＴＣＰ２によれば、どちらの環境下でも対応可能である。

【０１８８】

また、異なったＬＡＮ間や複数キャリアにまたがったＬＡＮ間でＶｏＩＰを使ったインターネット電話に対しても、ＩＰｓｅｃとＳＳＬでは対応不可能であったが、本発明のＴＣＰ２によれば完全に対応することができる。

【０１８９】

図２４は、ＴＣＰ２の優位性を説明するための図であるが、セキュリティのないプロトコルスタック（ａ）に、従来のＳＳＬを適用したケース（ｂ）と、ＩＰｓｅｃを適用したケース（ｃ）と、本発明のＴＣＰ２（ＴＣＰｓｅｃ／ＵＤＰｓｅｃ）を適用したケース（ｄ）を比較して示している。図２４（ｂ）のＳＳＬは既に述べたように、セッション層（第５層）のソケットに設けられているので、上位のアプリケーションに対する互換性がないのである。このため、ＳＳＬ自体、上述のような問題を抱えることになっている。また図２４（ｃ）のＩＰｓｅｃは、ネットワーク層（第３層）にあり、ＩＰ層での互換性がないため、ネットワークを構成する上で上述したような様々な制約を受けることになる。これに対して、図２４（ｄ）のＴＣＰ２（ＴＣＰｓｅｃ／ＵＤＰｓｅｃ）は、トランスポート層（第４層）に導入する暗号化技術であり、このためアプリケーションから見るとソケットをそのまま利用することができ、またネットワークから見るとＩＰもそのまま利用できるのでネットワークを構成する上での制約は受けない。

【０１９０】

以上のように、本発明のＴＣＰ２を用いた暗号化通信システム、暗号化通信方法は、既存の暗号化処理システムに比べても、特にデータの漏洩、改ざん、なりすまし、進入そして攻撃に対して極めて高いセキュリティ機能を有するものである。

なお、本発明は、以上説明した実施の形態に限定されるものではなく、特許請求項に記載した本発明の要旨を逸脱しない範囲において、さらに多くの実施形態を含むものであることは言うまでもない。

【図面の簡単な説明】

【０１９１】

【図１】本発明の通信システムに用いられるＴＣＰ２のプロトコルスタックを示す図である。

【図２】本発明のＴＣＰ２を用いた通信システムの第１の実施形態（ＴＣＰｓｅｃによるＥＣアプリケーション）におけるシステムの全体構成図である。

【図３】本発明のＴＣＰ２を用いた通信システムの第２の実施形態（ＵＤＰｓｅｃによる放送アプリケーション）におけるシステムの全体構成図である。

【図４】本発明のＴＣＰ２の中の３つのプロトコルスタックのパケット構造とその暗号化範囲及び認証範囲を示す図である。（ａ）、（ｂ）、（ｃ）は、それぞれＴＣＰｓｅｃ／ＩＰｓｅｃ、ＴＣＰｓｅｃ／ＩＰ、ＵＤＰｓｅｃ／ＩＰのパケット構造、及び各暗号化範囲、完全性認証の適用範囲を示す図である。

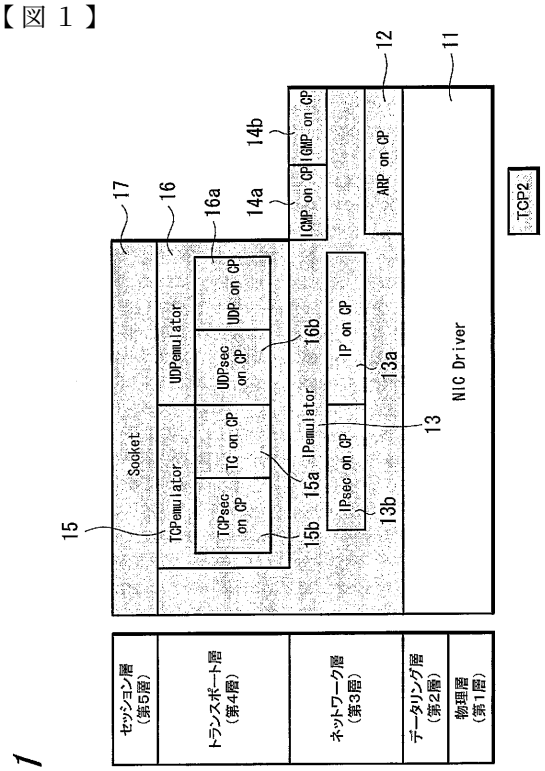
【図５】本発明のＴＣＰ２の実施形態であるＴＣＰ／ＴＣＰｓｅｃパッシブオープンの処理を示す流れ図である。

【図６】本発明のＴＣＰ２の実施形態であるＴＣＰ／ＴＣＰｓｅｃアクティブオープンの

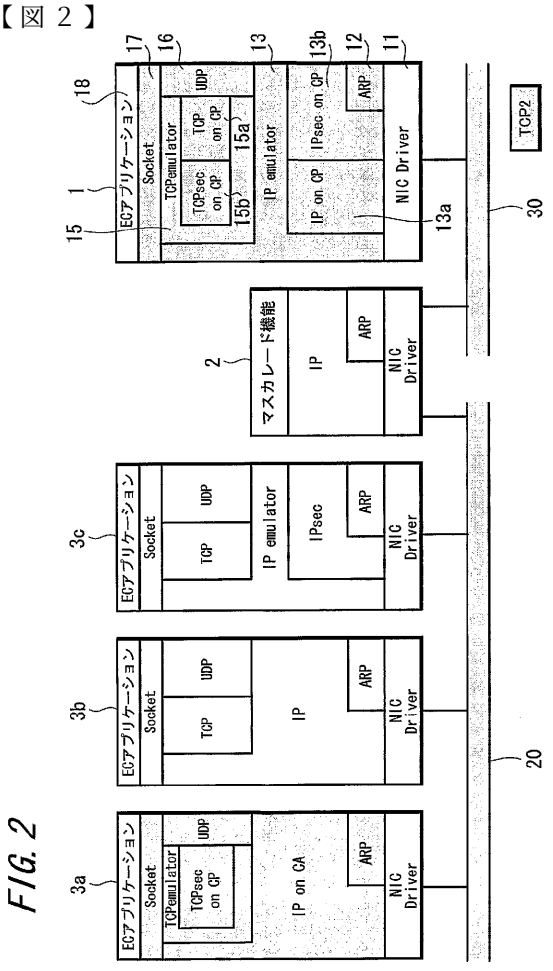
処理を示す流れ図である。
【図７】標準ＴＣＰと本発明のＴＣＰｓｅｃのホストＡ（アクティブオープン）とホストＢ（パッシブオープン）間の通信のやり取りを示すシーケンス図である。
【図８】図５のＴＣＰパッシブオープン処理Ｓ５の詳細を示す流れ図である。
【図９】図５のＴＣＰｓｅｃパッシブオープン処理Ｓ６の詳細を示す流れ図である。
【図１０】図６のＴＣＰアクティブオープン処理Ｓ１２の詳細を示す流れ図である。
【図１１】図６のＴＣＰｓｅｃアクティブオープン処理Ｓ１３の詳細を示す流れ図である。
【図１２】図９のＴＣＰｓｅｃ送受信処理Ｓ３７及び図１１のＴＣＰｓｅｃ送受信処理Ｓ７６の詳細を示す流れ図である。
【図１３】図９のＴＣＰｓｅｃパッシブ接続処理Ｓ４８の詳細を示す流れ図である。
【図１４】図１１のＴＣＰｓｅｃアクティブ接続処理Ｓ８８の詳細を示す流れ図である。
【図１５】本発明のＴＣＰ２の実施形態であるＵＤＰ／ＵＤＰｓｅｃオープンの処理を示す流れ図である。
【図１６】本発明のＴＣＰ２を用いたＵＤＰ／ＵＤＰｓｅｃユニキャスト通信のシーケンス図である。
【図１７】本発明のＴＣＰ２を用いたＵＤＰ／ＵＤＰｓｅｃブロードキャスト通信のシーケンス図である。
【図１８】図１５のＵＤＰオープン処理Ｓ１２４の詳細を示す流れ図である。
【図１９】図１５のＵＤＰｓｅｃオープン処理Ｓ１２５の詳細を示す流れ図である。
【図２０】図１９のＵＤＰｓｅｃブロードキャスト受信開始処理Ｓ１４１の詳細を示す流れ図である。
【図２１】図１９のＵＤＰｓｅｃユニキャスト送信開始処理Ｓ１４６の詳細を示す流れ図である。
【図２２】図１９のＵＤＰｓｅｃデータ送受信処理Ｓ１４４の詳細を示す流れ図である。
【図２３】図１９のＵＤＰｓｅｃユニキャスト受信開始処理Ｓ１３７の詳細を示す流れ図である。
【図２４】本発明のＴＣＰ２を従来のＩＰｓｅｃ又はＳＳＬを適用した場合と比較したメリットを説明するための図である。
【図２５】従来のＩＰｓｅｃを用いた標準的な通信のプロトコルスタックを示す図である。
【図２６】従来のＳＳＬを用いた標準的な通信のプロトコルスタックを示す図である。
【符号の説明】
【０１９２】
１・・・ホストコンピュータ、
２・・・ネットワーク制御機器（ルータ）
３ａ、３ｂ、３ｃ・・・クライアント端末
４ａ、４ｂ、４ｃ・・・クライアント端末
１１・・・ＮＩＣドライバ
１２・・・ＡＲＰ　又は　ＡＲＰ　ｏｎ　ＣＰ
１３・・・ＩＰエミュレータ
１３ｂ・・・ＩＰｓｅｃ　ｏｎ　ＣＰ
１５・・・ＴＣＰエミュレータ
１５ｂ・・・ＴＣＰｓｅｃ　ｏｎ　ＣＰ
１６・・・ＵＤＰエミュレータ
１６ｂ・・・ＵＤＰｓｅｃ　ｏｎ　ＣＰ
１７・・・ソケット（Socket）
４１・・・ＩＰヘッダ
４２・・・ＥＳＰヘッダ
４３・・・ＴＣＰヘッダ

４４・・・ＴＣＰｓｅｃ付加情報
４５・・・データ（ペイロード）
４６・・・ＴＣＰｓｅｃ付加トレーラ
４７・・・ＴＣＰｓｅｃ付加認証データ
４８・・・ＥＳＰトレーラ
４９・・・ＥＳＰ認証データ

【図１】



【図２】



【図３】

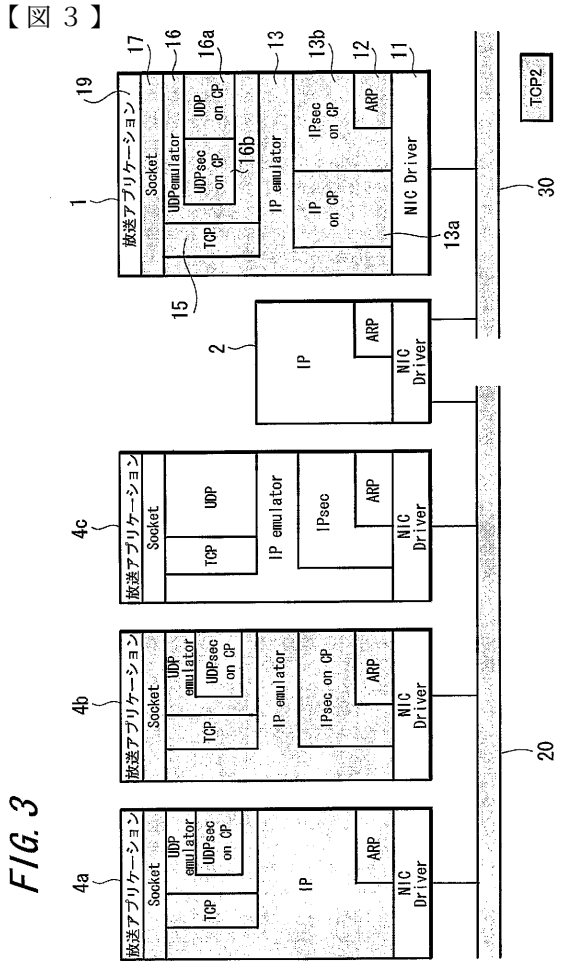


FIG. 3

【図４】

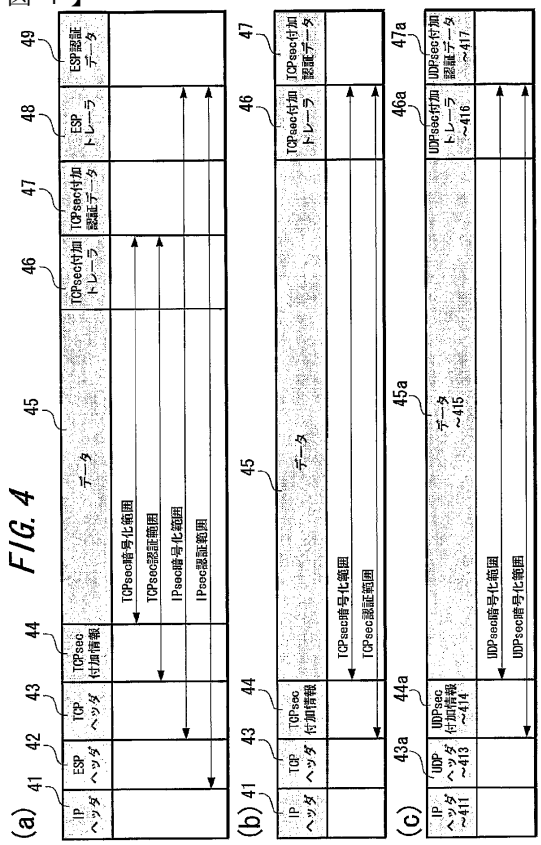
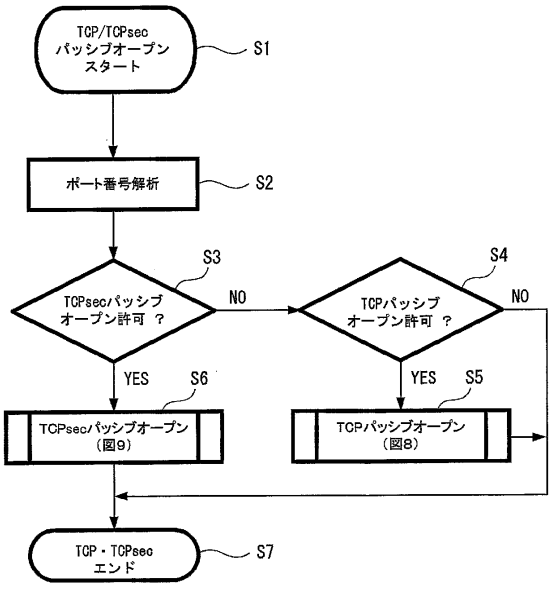


FIG. 4

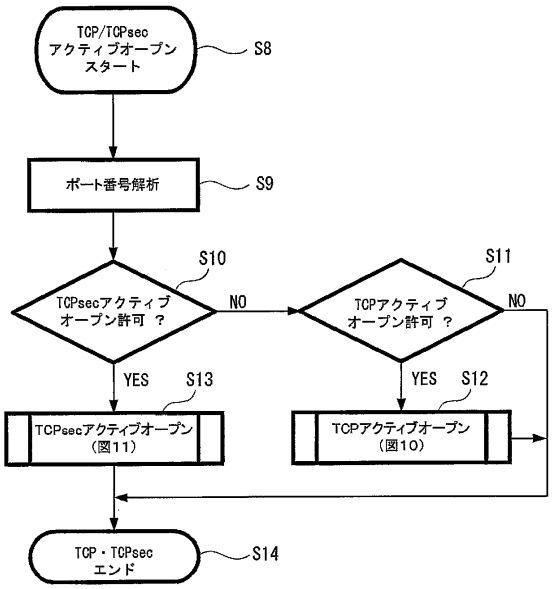
【図５】

FIG. 5



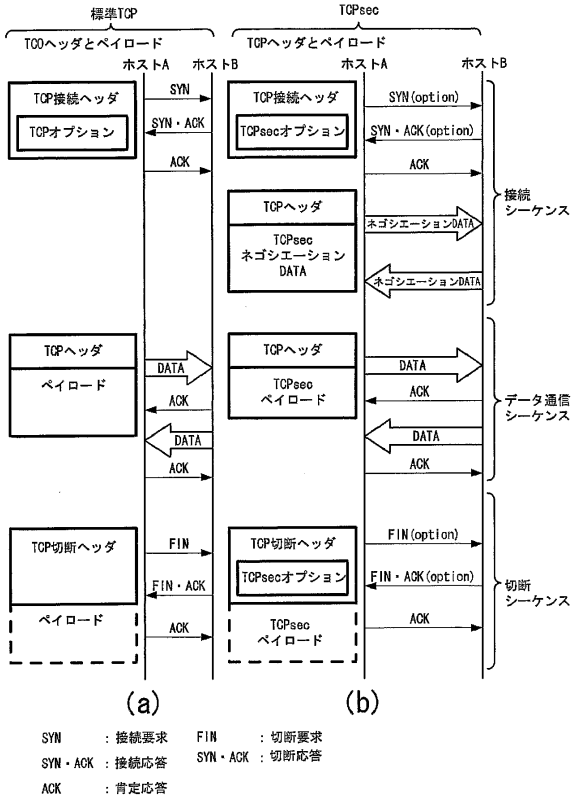
【図６】

FIG. 6



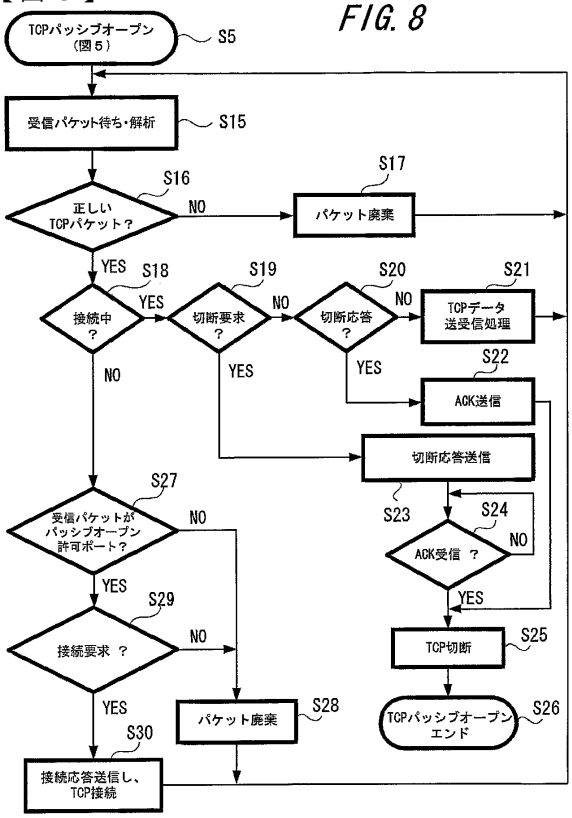
【図７】

FIG. 7



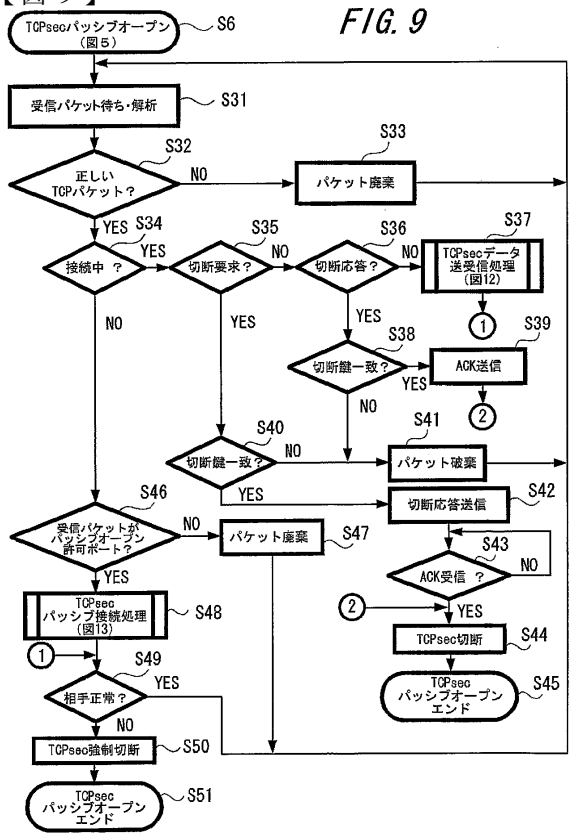
【図８】

FIG. 8



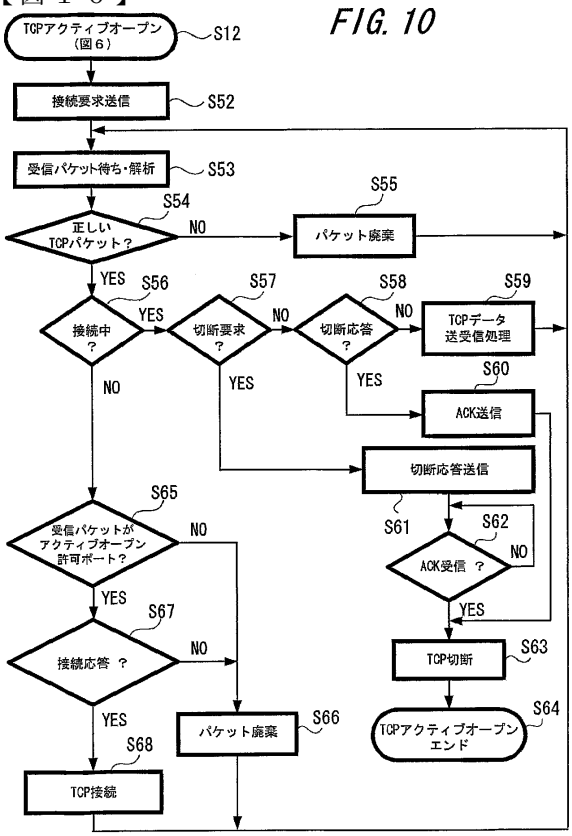
【図９】

FIG. 9



【図１０】

FIG. 10



【図１１】

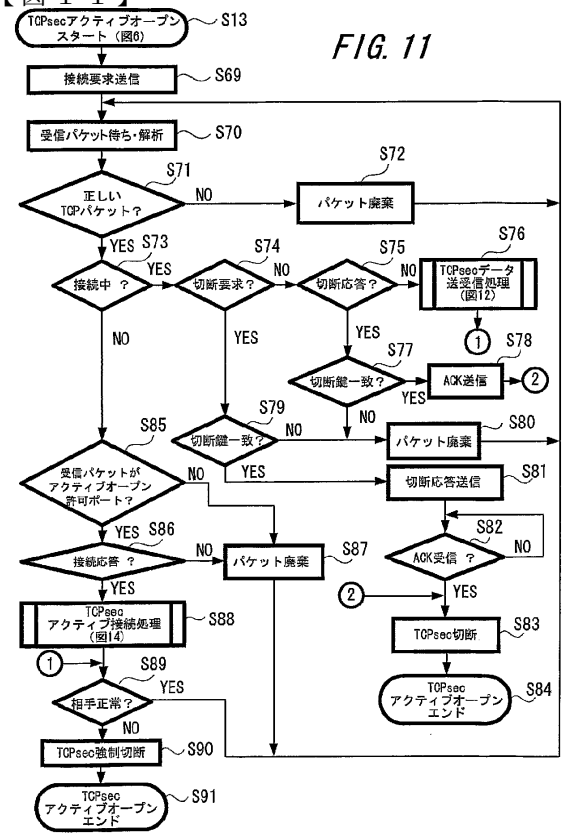


FIG. 11

【図１２】

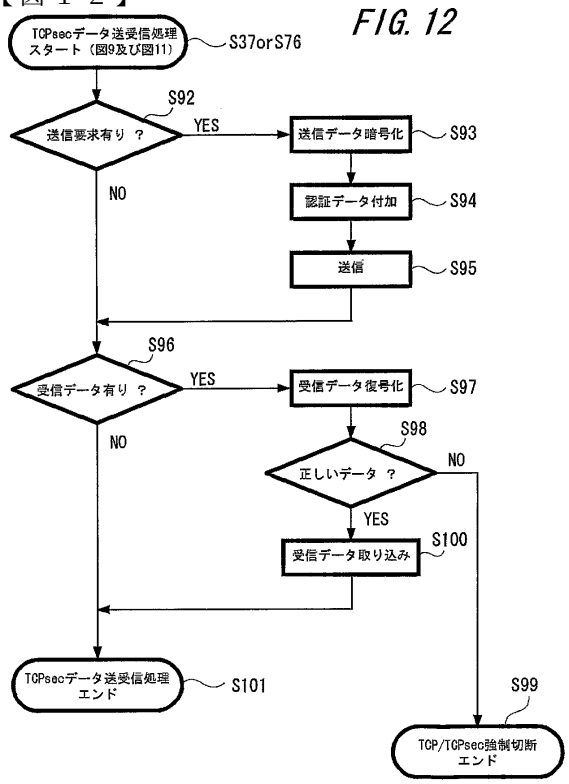
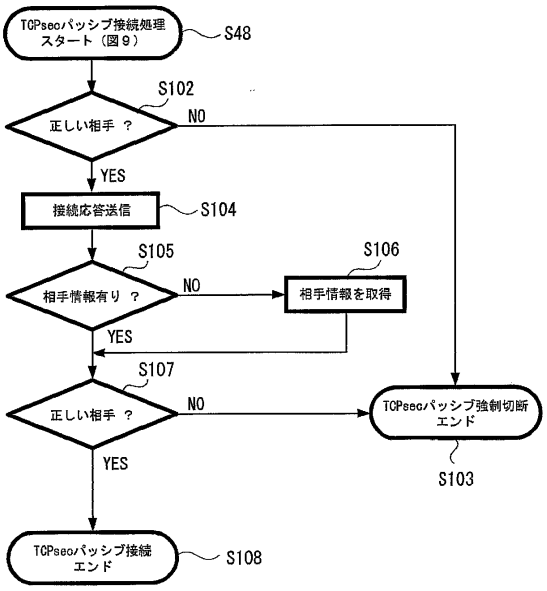


FIG. 12

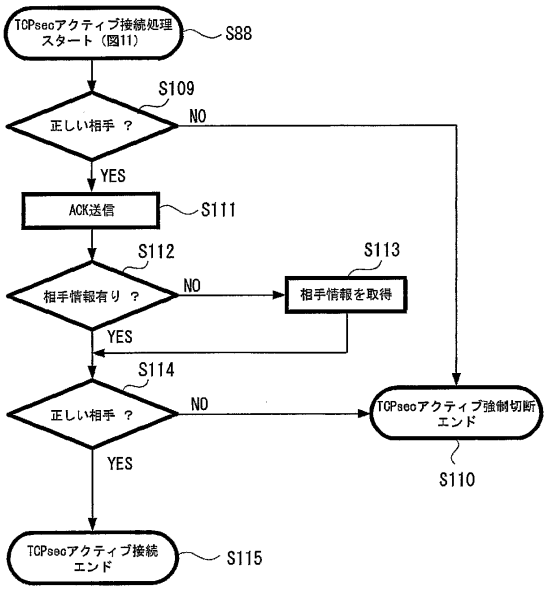
【図１３】

FIG. 13

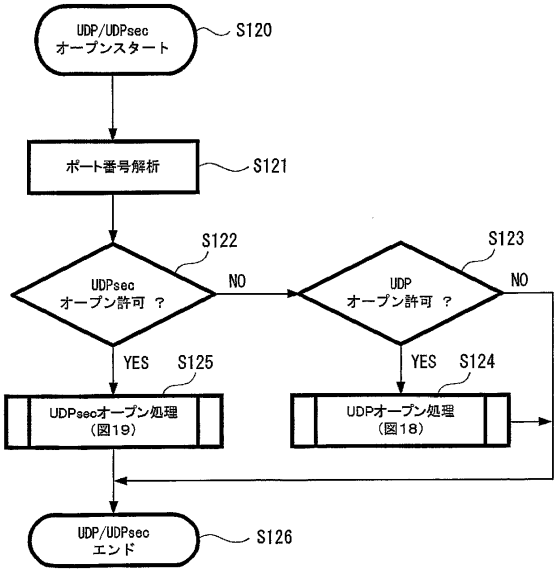


【図１４】

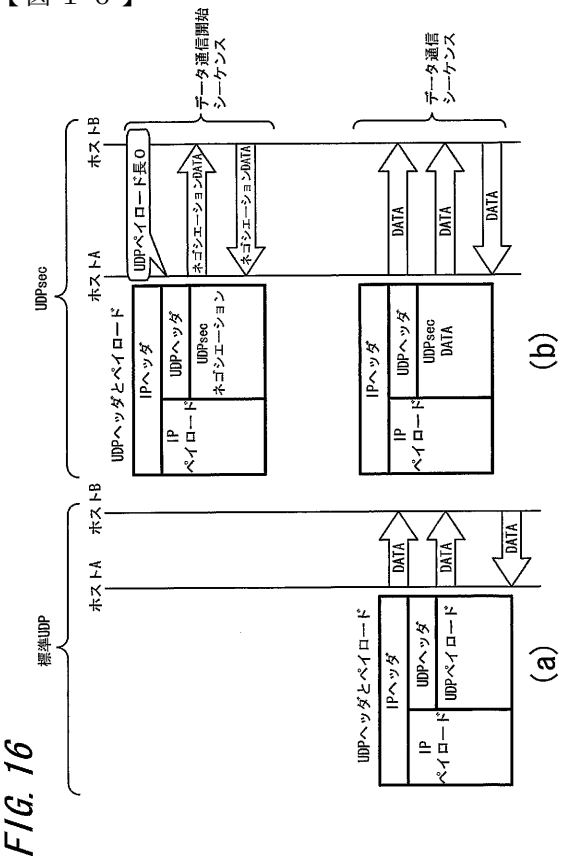
FIG. 14



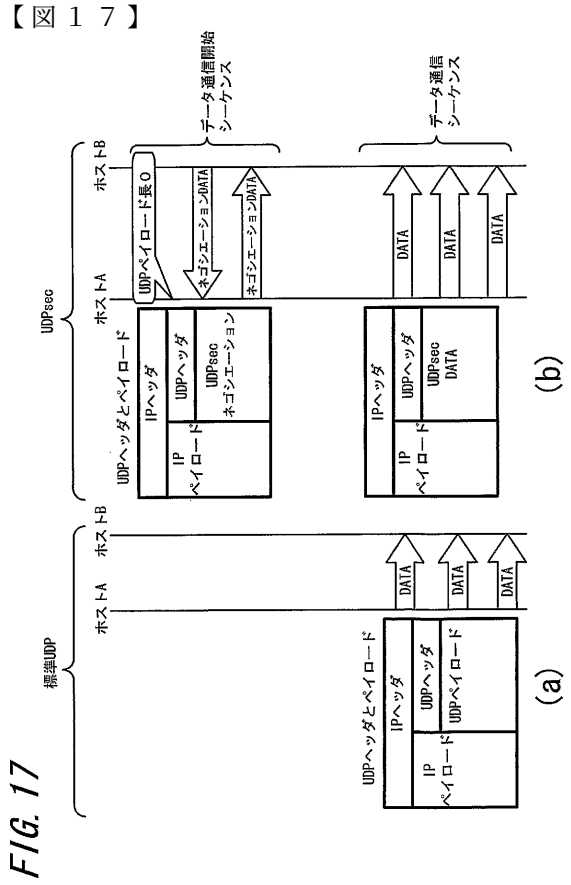
【図１５】
FIG. 15



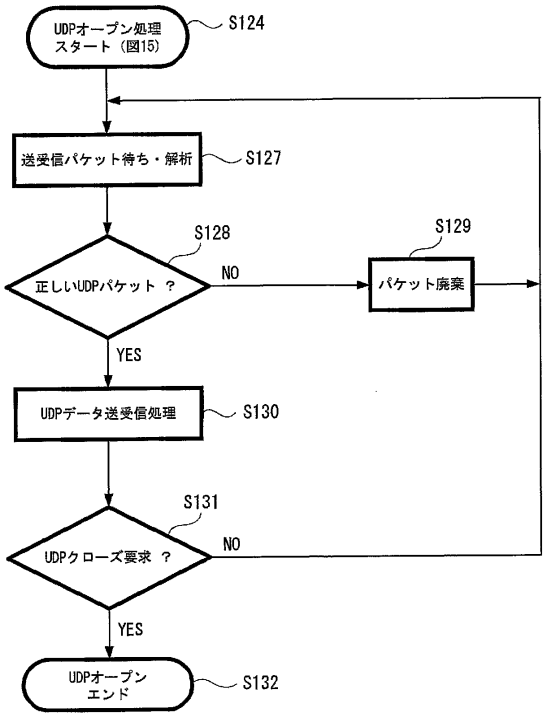
【図１６】
FIG. 16



【図１７】
FIG. 17

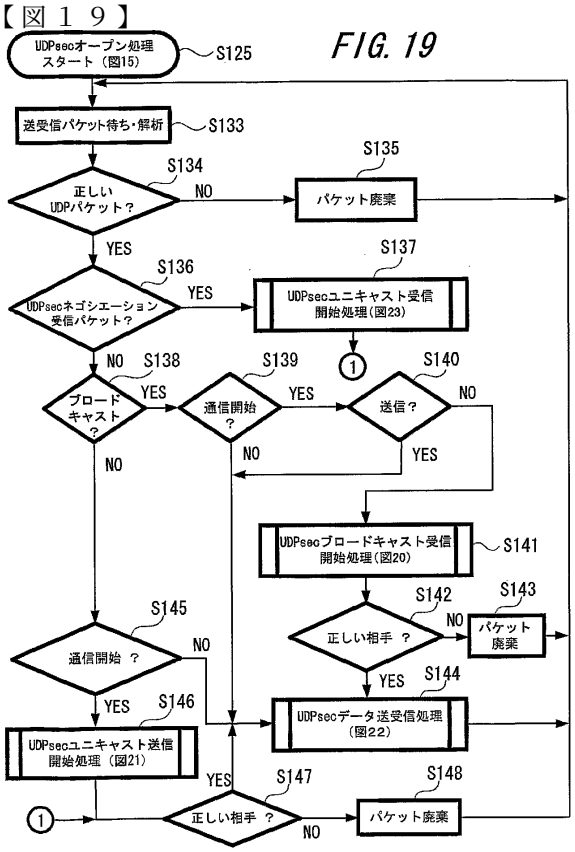


【図１８】
FIG. 18



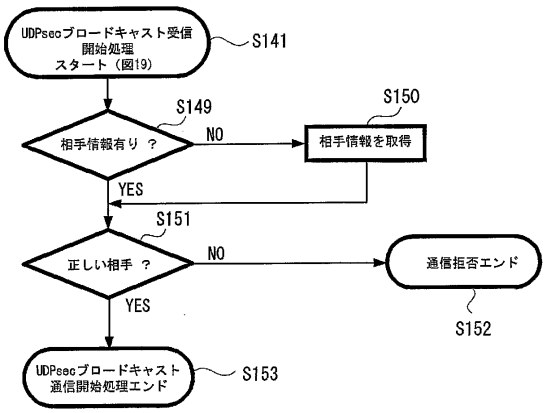
【図１９】

FIG. 19



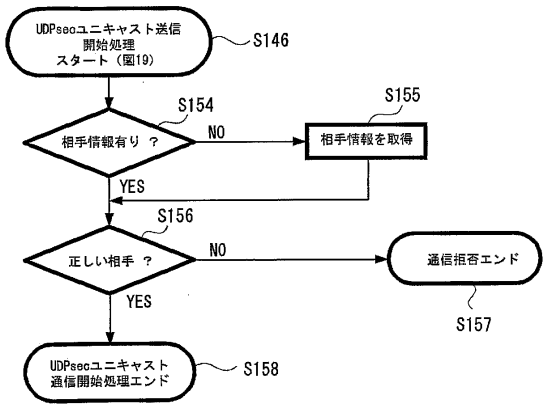
【図２０】

FIG. 20



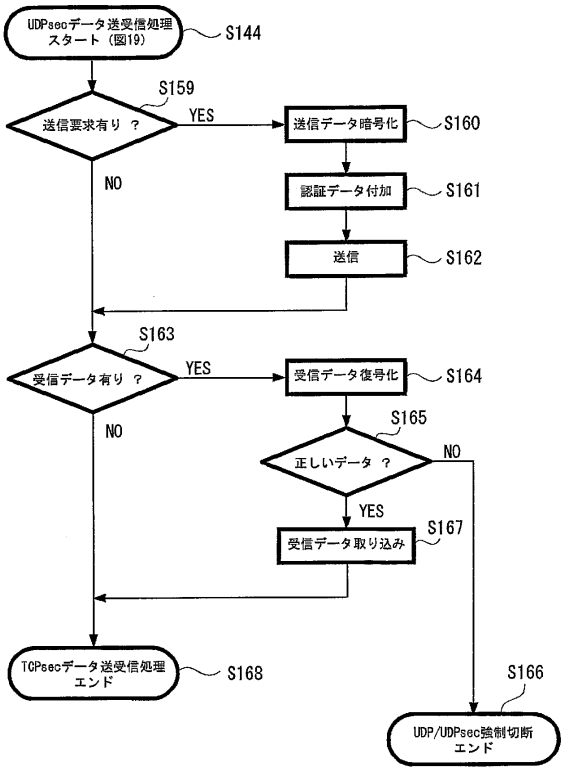
【図２１】

FIG. 21

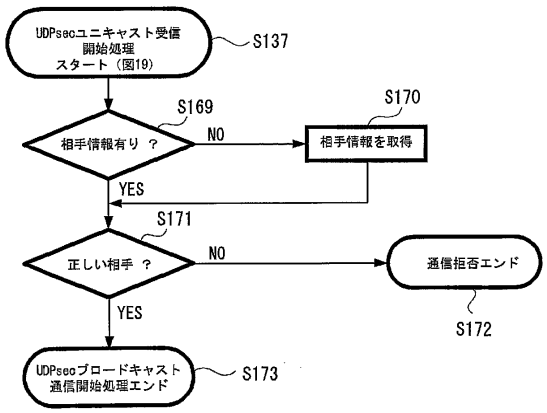


【図２２】

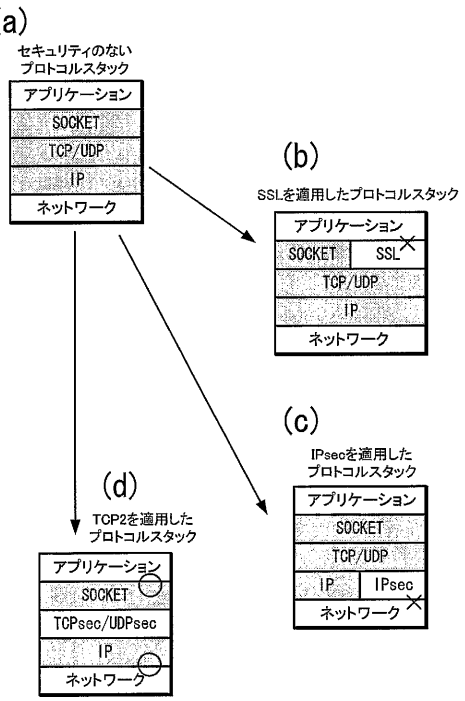
FIG. 22



【図２３】
FIG. 23



【図２４】
FIG. 24



【図２５】

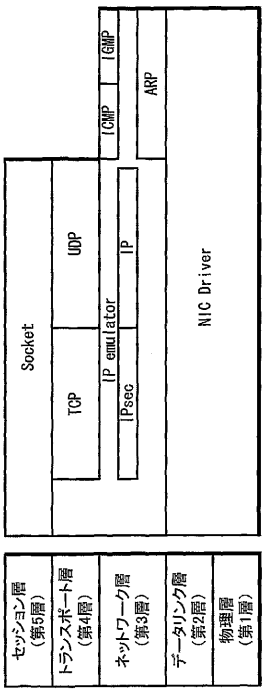


FIG. 25

【図２６】

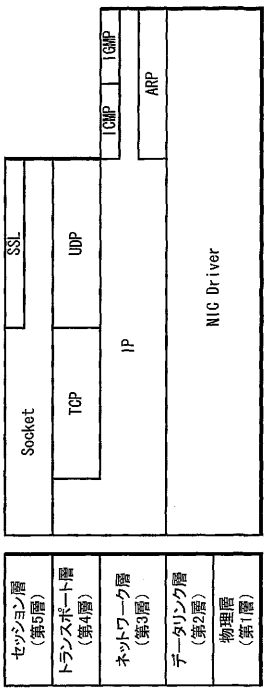


FIG. 26

フロントページの続き

(72)発明者　小川　恵子
大阪府大阪市北区東天満１丁目１番１９号　アーバンエース東天満ビル　ティー・ティー・ティー株式会社内

審査官　清水　稔

(56)参考文献　特開２００２－２３６６１８（ＪＰ，Ａ）
上野英俊他，マルチキャスト通信のためのトランスポートデータ暗号化プロトコルの提案と実装，信学技報ＮＳ２００３－４０，２００３年　6月１３日
Camillo Saers，暗号化技術が拓くインターネット新時代－トランスポート層プロトコルＳＳＨが可能にする安全な遠隔ログイン，ＤＤＪ第７巻第２号，株式会社翔泳社，１９９８年　2月１日
山口利和，ＬＡＮセキュリティ通信技術－ＴＣＰレイヤにおける通信データの暗号化－，ＮＴＴ　Ｒ＆Ｄ，１９９５年　8月１０日，Vol.44　No.8

(58)調査した分野(Int.Cl.，ＤＢ名)
H04L 12/22
H04L 29/06