(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5685326号
(P5685326)
(45)発行日　平成27年3月18日(2015.3.18)　(24)登録日　平成27年1月23日(2015.1.23)

(51)Int.Cl.		F I		
H04L 12/70	(2013.01)	H04L	12/70	A
H04L 12/22	(2006.01)	H04L	12/22	
H04L 12/66	(2006.01)	H04L	12/66	B

請求項の数 20　外国語出願　(全 38 頁)

(21)出願番号	特願2014-51(P2014-51)
(22)出願日	平成26年1月6日(2014.1.6)
(62)分割の表示	特願2011-81417(P2011-81417)の分割
原出願日	平成11年10月29日(1999.10.29)
(65)公開番号	特開2014-90493(P2014-90493A)
(43)公開日	平成26年5月15日(2014.5.15)
審査請求日	平成26年1月7日(2014.1.7)
(31)優先権主張番号	60/106,261
(32)優先日	平成10年10月30日(1998.10.30)
(33)優先権主張国	米国(US)
(31)優先権主張番号	60/137,704
(32)優先日	平成11年6月7日(1999.6.7)
(33)優先権主張国	米国(US)

(73)特許権者　509316316
バーネットエックス　インコーポレーティッド
アメリカ合衆国　ネバダ州　ゼファー　コーブ　ピーオー　ボックス　４３９
(74)代理人　100102978
弁理士　清水　初志
(74)代理人　100102118
弁理士　春名　雅夫
(74)代理人　100160923
弁理士　山口　裕孝
(74)代理人　100119507
弁理士　刑部　俊
(74)代理人　100142929
弁理士　井上　隆一

最終頁に続く

(54)【発明の名称】保証されたシステム可用性を有する安全な通信のためのアジル・ネットワーク・プロトコル

(57)【特許請求の範囲】

【請求項１】

以下の段階を含む、第1のデバイスと第2のデバイスとの間で情報を送信する方法：

第1のデバイスから安全な通信セッションの要求を送信する段階であって、該要求が、該第1のデバイスを識別するトークンを含む、段階；

該第1のデバイスにおいて該安全な通信セッションの要求に応答して、該第2のデバイスとの該安全な通信セッションを確立する際に該第1のデバイスによる使用のための複数の送信元／着信先インタネット・プロトコル・アドレス・ペアを含む要求承認を受信する段階；

該要求承認に含まれる該複数のインタネット・プロトコル・アドレス・ペアのうちの少なくとも１つを使用して第1のデバイスと第2のデバイスとの間で該安全な通信セッションを確立する段階；および

該安全な通信セッションを用いて、第1のデバイスから第2のデバイスへとデータ・パケットを送信し、かつ、第2のデバイスからのデータ・パケットを第1のデバイスで受信する、段階であって、送信される第1のデバイスからの該データ・パケットと受信される第2のデバイスからの該データ・パケットが、オーディオ・データまたはビデオ・データを含む、段階。

【請求項２】

第1のデバイスから第2のデバイスへと送信されるデータ・パケットのそれぞれにディスクリミネータ値を埋め込む段階をさらに含み、各ディスクリミネータ値が、連続するデー

タ・パケット間で周期的に変化し、かつ各データ・パケット内の他のデータのみに基づくわけではない、請求項1記載の方法。

【請求項３】

インタネット・プロトコル・ヘッダ内のインタネット・プロトコル・アドレスがディスクリミネータ値として使用され、該インタネット・プロトコル・アドレスが、インタネットを介してデータ・パケットをルーティングするために使用される、請求項2記載の方法。

【請求項４】

ディスクリミネータ値として媒体アクセス制御（MAC）ハードウェア・アドレスが使用され、該MACハードウェア・アドレスがローカル・エリア・ネットワーク上でデータ・パケットをルーティングするために使用される、請求項2記載の方法。

【請求項５】

ディスクリミネータ値がさらに、擬似乱数的に生成された値である、請求項2記載の方法。

【請求項６】

以下の段階をさらに含む、請求項1記載の方法：

受信された第2のデバイスからのデータ・パケットのそれぞれにおけるディスクリミネータ値を、1組の有効なディスクリミネータ値と比較する段階；

一致を検出した場合、該受信されたデータ・パケットをさらなる処理のために受け入れる段階；および

一致が検出されない場合、該受信されたデータ・パケットを拒否する段階。

【請求項７】

1組の有効なディスクリミネータ値が、有効なディスクリミネータ値のウインドウであり、該ウインドウが、一致を検出したことに応答して移動する、請求項6記載の方法。

【請求項８】

送信テーブルおよび受信テーブルを第1のデバイス内に維持する段階をさらに含み、該送信テーブルが、発信データ・パケットに挿入すべき有効なディスクリミネータ値のリストを含み、該受信テーブルが、着信データ・パケットと比較すべき有効なディスクリミネータ値のリストを含む、請求項1記載の方法。

【請求項９】

第1のデバイスと第2のデバイスとの間で同期要求を送信する段階をさらに含み、第2のデバイスが、該同期要求を使用して有効なディスクリミネータ値の同期を維持する、請求項1記載の方法。

【請求項１０】

以下の段階をさらに含む、請求項1記載の方法：

複数のデータ・パケットにオーディオ・データおよびビデオ・データをどのように分配するかを決定する共通のアルゴリズムを、第1のデバイスと第2のデバイスとの間で確立する段階；ならびに

該共通のアルゴリズムを用いて、該オーディオ・データおよびビデオ・データを、第1のデバイスから第2のデバイスへと送信されるデータ・パケットに分配する段階。

【請求項１１】

第1のデバイスから第2のデバイスへと送信されるデータ・パケットのそれぞれについて、複数のコンピュータを通る複数の物理送信パスのうちの1つを選択する段階をさらに含み、第1のデバイスから第2のデバイスへと送信される該データ・パケットが、該選択された物理送信パスを用いて送信される、請求項1記載の方法。

【請求項１２】

1つまたは複数のプロセッサ；ならびに

該1つまたは複数のプロセッサにより実行されると、以下を含む動作：

第1のデバイスから安全な通信セッションの要求を送信することであって、ここで該要求は該第1のデバイスを識別するトークンを含む；

該第1のデバイスにおいて該安全な通信セッションの要求に応答して、該第2のデバイスとの該安全な通信セッションを確立する際に該第1のデバイスによる使用のための複数の送信元／着信先インタネット・プロトコル・アドレス・ペアを含む要求承認を受信すること；

該要求承認に含まれる該複数のインタネット・プロトコル・アドレス・ペアのうちの少なくとも１つを使用して第1のデバイスと第2のデバイスとの間で該安全な通信セッションを確立すること；および

該安全な通信セッションを用いて、第1のデバイスから第2のデバイスへとデータ・パケットを送信し、かつ、第2のデバイスからのデータ・パケットを第1のデバイスで受信することであって、ここで、送信される第1のデバイスからの該データ・パケットと受信される第2のデバイスからの該データ・パケットは、オーディオ・データまたはビデオ・データを含む

を実施する命令を備える、機械可読媒体

を含む、第1のデバイスと第2のデバイスとの間で情報を送信するためのシステム。

【請求項１３】

動作が、第1のデバイスから第2のデバイスへと送信されるデータ・パケットのそれぞれにディスクリミネータ値を埋め込むことをさらに含み、各ディスクリミネータ値が、連続するデータ・パケット間で周期的に変化し、かつ各データ・パケット内の他のデータのみに基づくわけではない、請求項12記載のシステム。

【請求項１４】

動作が、

受信された第2のデバイスからのデータ・パケットのそれぞれにおけるディスクリミネータ値を、1組の有効なディスクリミネータ値と比較すること；

一致を検出した場合、該受信されたデータ・パケットをさらなる処理のために受け入れること；および

一致が検出されない場合、該受信されたデータ・パケットを拒否すること

をさらに含む、請求項12記載のシステム。

【請求項１５】

動作が、第1のデバイスと第2のデバイスとの間で同期要求を送信することをさらに含み、第2のデバイスが、該同期要求を使用して有効なディスクリミネータ値の同期を維持する、請求項12記載のシステム。

【請求項１６】

動作が、

複数のデータ・パケットにオーディオ・データおよびビデオ・データをどのように分配するかを決定する共通のアルゴリズムを、第1のデバイスと第2のデバイスとの間で確立すること；ならびに

該共通のアルゴリズムを用いて、該オーディオ・データおよびビデオ・データを、第1のデバイスから第2のデバイスへと送信されるデータ・パケットに分配すること

をさらに含む、請求項12記載のシステム。

【請求項１７】

動作が、第1のデバイスから第2のデバイスへと送信されるデータ・パケットのそれぞれについて、複数のコンピュータを通る複数の物理送信パスのうちの1つを選択することをさらに含み、第1のデバイスから第2のデバイスへと送信される該データ・パケットが、該選択された物理送信パスを用いて送信される、請求項12記載のシステム。

【請求項１８】

以下の段階を含む、第1のデバイスと第2のデバイスとの間で情報を送信する方法：

該第1のデバイスから安全な通信セッションの要求を送信する段階；

該第1のデバイスにおいて該安全な通信セッションの要求に応答して、該第2のデバイスとの該安全な通信セッションを確立する際に該第1のデバイスによる使用のための複数の送信元／着信先インタネット・プロトコル・アドレス・ペアを含む要求承認を受信する段

階；
該要求承認に含まれる該複数インタネット・プロトコル・アドレスペアのうちの少なくとも１つを使用して該第1のデバイスと該第2のデバイスとの間で該安全な通信セッションを確立する段階；および
該安全な通信セッションを用いて、該第1のデバイスから該第2のデバイスへとデータ・パケットを送信する段階。
【請求項１９】
安全な通信セッションを用いて、該第2のデバイスからのデータ・パケットを該第1のデバイスで受信する段階をさらに含む、請求項18記載の方法。
【請求項２０】
送信される該第1のデバイスからのデータ・パケットと受信される該第2のデバイスからのデータ・パケットが、オーディオ・データおよびビデオ・データを含む、請求項19記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
関連出願
本出願は、すでに出願されている2つの特許仮出願第60/106261号（1998年10月30日出願）および第60/137704号（1999年6月7日出願）の主題の優先権を主張するものであり、その主題は全体的に組み入れられる。
【背景技術】
【０００２】
発明の背景
インタネットを介した通信のセキュリティおよび匿名性を実現するために、極めて様々な方法が提案され実施されている。このように種類が多いのは、1つには様々なインタネット・ユーザのニーズがそれぞれ異なるためである。これらの様々なセキュリティ技法を論じるうえで助けになる基本的なヒューリスティック構造を図1に示す。2つの端末、すなわち、発信元端末100と着信先端末110はインタネットを介して通信する。通信を安全なものし、すなわち、通信が盗聴されないものであることが望ましい。たとえば、端末100がインタネット107を介して端末110に秘密情報を送信することがある。また、端末100が端末110と通信していることを盗聴者が気付かないようにすることが望ましい場合もある。たとえば、端末100がユーザであり、端末110がウェブ・サイトを運営している場合、端末100のユーザは、どのウェブ・サイトに「アクセスしている」かを介在するネットワーク内の誰にも知られたくない場合がある。したがって、たとえば、自分たちの市場研究の対象を公開しないことを望み、したがってウェブ・サイトまたはその他のインタネット資源のうちのどれに「アクセスしている」かを部外者が知るのを防止することを望む会社には匿名性が重要である。セキュリティに関するこの2つの問題はそれぞれ、データ・セキュリティおよび匿名性と呼ぶことができる。
【０００３】
データ・セキュリティは通常、ある形式のデータ暗号化を使用して対処される。発信元端末100と110の両方で暗号化鍵48が知られている。この鍵は、発信元端末100および着信先端末110のそれぞれでの専用鍵および公開鍵でも、あるいは対称鍵（同じ鍵が、両当事者によって暗号化および復号のために使用される）でもよい。多くの暗号化方法が知られており、この場合に使用可能である。
【０００４】
トラフィックをローカル管理者またはISPから隠すために、ユーザは、このローカル管理者またはISPが暗号化されたトラフィックのみを見るように、暗号化されたチャネルを通して外部エポキシと通信する際にローカル・エポキシ・サーバを使用することができる。プロキシ・サーバは、着信先サーバが発信元クライアントのIDを判定するのを妨げる。このシステムはクライアントと着信先サーバとの間に介在する中間サーバを使用する。着

信先サーバは、プロキシ・サーバのインタネット・プロトコル（IP）アドレスのみを見て、発信元クライアントは見ない。ターゲット・サーバは外部プロキシのアドレスのみを見る。この方式は、信用できる外部プロキシ・サーバに依存する。また、プロキシ方式は、送信側および受信側のIDを判定するトラフィック分析方法の影響を受けやすい。プロキシ・サーバの他の重要な制限は、サーバが呼出し側と被呼出し側の両方のIDを知ることである。多くの例では、端末Aなどの発信元端末は、インタネット・サービス・プロバイダ（ISP）によってプロキシ・サーバが設けられている場合、端末のIDをプロキシから隠しておくことを望む。

【０００５】

トラフィック分析を無効にするために、Chaumのミックスと呼ばれる方式では、ダミー・メッセージを含む固定長メッセージを送受信するプロキシ・サーバが使用される。複数の発信元端末がミックス（サーバ）を通して複数のターゲット・サーバに接続される。どの発信元端末が、接続されているターゲット・サーバのうちのどれと通信しているかを判定するのは困難であり、ダミー・メッセージによって、盗聴者がトラフィックを分析することによって通信ペアを検出することは困難になる。この方法の欠点は、ミックス・サーバが打破される恐れがあることである。この危険に対処する1つの方法は、複数のミックス間で責任を分散することである。すなわち、1つのミックスが打破された場合でも、発信元端末およびターゲット端末のIDを隠されたままにすることができる。この方式では、複数のミックスをコンプロマイズしないかぎり、発信元端末とターゲット端末との間に介在する中間サーバを判定できないように、いくつかの代替ミックスが必要である。この方式では、メッセージが複数の暗号化アドレスレイヤで覆われる。シーケンス中の第1のミックスはメッセージのアウタ・レイヤのみを復号して、シーケンス中の次の着信先ミックスを知ることができる。第2のミックスはこのメッセージを復号して次のミックスを知り、以下同様な手順が繰り返される。ターゲット・サーバは、メッセージを受信し、任意選択で、データを同様に送り返すためのリターン情報を含むマルチレイヤ暗号化ペイロードを受信する。このようなミックス方式を無効にするには、ミックスを共謀させるしかない。パケットがすべて固定長であり、パケットにダミー・パケットが混合されている場合、どんな種類のトラフィック分析も不可能である。

【０００６】

「クラウド」と呼ばれる他の匿名性技法は、発信元端末をクラウドと呼ばれるプロキシ・グループに属させることによって発信元端末のIDを中間プロキシから保護する技法である。クラウド・プロキシは、発信元端末とターゲット端末との間に介在する。メッセージが通過する各プロキシは、上流側プロキシによって無作為に選択される。各中間プロキシは、「クラウド」内の無作為に選択された他のプロキシまたは着信先にメッセージを送信することができる。したがって、クラウド・メンバーでも、メッセージの発信側が前のプロキシであるかどうかや、メッセージが他のプロキシから転送されたものに過ぎないのかどうかを判定することができない。

【０００７】

ZKS（ゼロ・ノリッジ・システム）匿名IPプロトコルは、ユーザが5つの異なる匿名のいずれかを選択できるようにし、同時に、デスクトップ・ソフトウェアが発信トラフィックを暗号化し、ユーザ・データグラム・プロトコル（UDP）パケット内に隠される。2+-ホップ・システム内の第1のサーバは、UDPパケットを得て、1つの暗号化レイヤを除去して別の暗号化レイヤを付加し、次いでこのトラフィックを次のサーバに送信する。このサーバはさらに別の暗号化レイヤを除去して新しい暗号化レイヤを付加する。ユーザはホップの数を制御することができる。最後のサーバで、トラフィックは、追跡不能なIPアドレスを用いて復号される。この技法はオニオン・ルーティングと呼ばれている。この方法は、トラフィック分析を使用して無効にすることができる。簡単な例では、低デューティ期間中のユーザからのパケットのバーストによって送信側および受信側のIDが知られることがある。

【０００８】

ファイアウォールは、許可されていないアクセスや、LANに接続されたコンピュータの悪意ある使用またはそのようなコンピュータに対する損害からLANを保護することを試みる。ファイアウォールは、管理オーバヘッドを維持することを必要とする中央化システムである。ファイアウォールは、仮想マシン・アプリケーション（「アプレット」）によって打破することができる。このようなアプリケーションは、たとえば、ユーザがファイアウォールの外部のサーバに機密情報を送信したり、モデムを使用してファイアウォール・セキュリティを回避することを勧めたりすることによってセキュリティ違反を引き起こすセキュリティの誤った意味を浸透させる。ファイアウォールは、出張旅行者、エクストラネット、小規模なチームなどの分散システムには有用ではない。

【発明の概要】

【０００９】

発明の概要

トンネル・アジル・ルーティング・プロトコル（TARP）と呼ばれるプロトコルを含む、インタネットを介して通信する安全機構は、固有の2レイヤ暗号化フォーマットおよび特殊なTARPルータを使用する。TARPルータは、機能が正規のIPルータと類似している。各TARPルータは、1つまたは複数のアドレスを有し、通常のIPプロトコルを使用してIPパケット・メッセージ（「パケット」または「データグラム」）を送信する。TARPルータを介してTARP端末間で交換されるIPパケットは、TARPルータおよびサーバ以外には真の着信先アドレスが隠される実際に暗号化されたパケットである。TARP　IPパケットに付加された通常のIPヘッダまたは「平文」IPヘッダまたは「外部」IPヘッダは、次のホップ・ルータまたは着信先サーバのアドレスのみを含む。すなわち、TARPパケットのIPヘッダは、IPヘッダの着信先フィールドで最終着信先を示すのではなく、常に、一連のTARPルータ・ホップ内の次のホップまたは最終着信先を指し示す。このことは、着信先は常に次のホップTARPルータおよび最終着信先であるので、傍受されたTARPパケットにはそのTARPパケットの真の着信先を明白に示すものがないことを意味する。

【００１０】

各TARPパケットの真の着信先は、リンク鍵を使用して生成された暗号化層に隠される。リンク鍵は、発信元TARP端末と着信先TARP端末との間に介在するホップ間の暗号化された通信に使用される暗号化鍵である。各TARPルータは、暗号化アウタ・レイヤを除去して各TARPパケットの着信先ルータを知ることができる。受信側TARPまたはルーティング端末は、TARPパケットの暗号化アウタ・レイヤを復号するのに必要なリンク鍵を識別するために、平文IPヘッダ内の送信側／受信側IP番号によって送信側端末を識別することができる。

【００１１】

暗号化アウタ・レイヤが除去された後、TARPルータは最終着信先を判定する。各TARPパケット140は、トラフィック分析を無効にする助けとなるように最小数のホップを受ける。ホップは、無作為に選択することができ、あるいは一定の値でもよい。その結果、各TARPパケットは、着信先に到着する前に地理的に異なるいくつかのルータの間を無作為に移動することができる。各移動は、独立に無作為に決定されるので、所与のメッセージを構成する各パケットごとに異なる可能性が非常に高くなる。この機能をアジル・ルーティングと呼ぶ。様々なパケットがそれぞれの異なる経路をとるため、侵入者が、マルチパケット・メッセージ全体を形成するすべてのパケットを得ることは困難になる。これに伴う利点は、後述の暗号化インナレイヤに関する利点である。アジル・ルーティングは、この目的を促進する他の機能、すなわち、あらゆるメッセージが複数のパケットに分割されるようにする機能と組み合される。

【００１２】

TARPルータのIPアドレスは一定でなくてよく、この機能をIPアジリティと呼ぶ。各TARPルータは、独立して、あるいは他のTARP端末またはTARPルータからの指示の下で、IPアドレスを変更することができる。変更可能な別の識別子またはアドレスも定義される。このアドレスは、TARPアドレスと呼ばれ、TARPルータおよびTARP端末のみに知られ、いつでもTARPルータまたはTARP端末によって参照テーブル（LUT）を使用して相関付けることがで

きる。TARPルータまたはTARP端末は、IPアドレスを変更したときに、他のTARPルータまたはTARP端末を更新し、これらのTARPルータまたはTARP端末はそれぞれのLUTを更新する。

【００１３】

メッセージ・ペイロードは、セッション鍵を使用しないかぎりロック解除できない、TARPパケット内の暗号化インナレイヤに隠される。セッション鍵は、介入するTARPルータがいずれも利用できない鍵である。セッション鍵は、TARPパケットのペイロードを復号し、データ・ストリームを再構成できるようにするために使用される。

【００１４】

リンク鍵およびセッション鍵を使用して通信を公開されないようにすることができ、任意の所望の方法に従ってリンク鍵およびセッション鍵を共用し使用することができる。たとえば、公開／専用鍵または対称鍵を使用することができる。

【００１５】

TARP発信元端末は、データ・ストリームを送信する場合、ネットワーク（IP）層プロセスによって生成された一連のIPパケットから一連のTARPパケットを構成する。（本明細書で使用される「ネットワーク層」、「データ・リンク層」「アプリケーション層」などが開放形システム間相互接続（OSI）ネットワーク用語に対応することに留意されたい。）これらのパケットのペイロードは、セッション鍵を使用して暗号化されたブロックおよびチェーン・ブロックとして組み立てられる。もちろん、この場合、すべてのIPパケットが同じTARP端末を着信先としているものと仮定する。このブロックは次いでインタリーブされ、インタリーブされた暗号化済みブロックは、生成すべき各TARPパケットごとに1つの一連のペイロードに分割される。次いで、データ・ストリーム・パケットから得たIPヘッダを使用して、各ペイロードに特殊なTARPヘッダIP_rが付加される。TARPヘッダは、通常のIPヘッダと同一であってよく、あるいは何らかの方法でカスタマイズすることができる。TARPヘッダは、着信先TARP端末でデータをインタリーブ解除するための方式またはデータ、実行すべきホップの数を示すタイム・ツー・リブ（TTL）パラメータ、ペイロードがたとえば、TCPデータを含むか、それともUDPデータを含むかを示すデータ・タイプ識別子、送信側のTARPアドレス、着信先TARPアドレス、パケットが実際のデータを含むか、それともデコイ・データを含むかに関するインディケータ、またはデコイ・データがTARPペイロード・データを通じてある方法で流布される場合にデコイ・データを除去する方式を含むべきである。

【００１６】

本明細書ではセッション鍵に関してチェーン・ブロック暗号化を論じているが、任意の暗号化方法を使用できることに留意されたい。好ましくは、チェーン・ブロック暗号化と同様に、暗号化プロセスの結果全体が得られないかぎり許可されない復号が困難にする方法を使用すべきである。したがって、暗号化されたブロックを複数のパケット間で分離し、侵入者がすべてのそのようなパケットにアクセスするのを困難にすることによって、通信のコンテンツに特別なセキュリティ・レイヤが与えられる。

【００１７】

ピーク・ツー・アベレージ・ネットワーク負荷を低減させることによってトラフィック分析を無効にするための助けになるように、デコイ・データまたはダミー・データをストリームに付加することができる。インタネット内のある点での通信バーストを他の点での通信バーストに結合して通信エンドポイントを知ることができなくなるように、時間またはその他の基準に応答して低トラフィック期間中により多くのデコイ・データを生成する能力をTARPプロセスに付与することが望ましい。

【００１８】

ダミー・データは、データをより多くの目立たないサイズのパケットに分割して、インタリーブウインドウのサイズを大きくすることを可能にし、同時に各パケットごとに合理的なサイズを維持するうえでも助けになる。（パケット・サイズは、単一の標準サイズでよく、あるいは一定の範囲のサイズから選択することができる。）各メッセージを複数のパケットに分割することが望ましい1つの主要な理由は、インタリーブの前にチェーン・

ブロック暗号化方式を使用して第1の暗号化レイヤが形成される場合に明らかである。メッセージの一部または全体に単一のブロック暗号化を適用し、次いでその部分または全体をインタリーブしていくつかの別々のパケットを得ることができる。パケットのアジルIPルーティングと、それに伴い、パケットのシーケンス全体を再構成して、ブロック暗号化された単一のメッセージ要素を形成するのが困難であることを考えると、デコイ・パケットがデータ・ストリーム全体を再構成することの困難さを著しく増大させることがわかる。

【００１９】

上記の方式は、データ・リンク層と、TARPシステムに参加している各サーバまたは端末のネットワーク層との間で動作するプロセスによって完全に実施することができる。上述の暗号化システムはデータ・リンク層とネットワーク層との間に挿入することができるので、暗号化された通信をサポートするうえで使用されるプロセスは、IP（ネットワーク）層以上でのプロセスに対して完全に透過的であってよい。TARPプロセスは、データ・リンク層のプロセスに対しても完全に透過的であってよい。したがって、ネットワーク層以上での動作も、データ・リンク層以下での動作も、TARPスタックを挿入することの影響を受けない。これにより、（たとえば、ハッカーによる）ネットワーク層への許可されないアクセスの困難さが大幅に増大するので、ネットワーク層以上でのすべてのプロセスに追加のセキュリティがもたらされる。セッション層で実行される新たに開発されたサーバの場合でも、セッション層よりも下のすべてのプロセスが侵害される可能性がある。このアーキテクチャでは、セキュリティが分散されることに留意されたい。すなわち、たとえば、移動中の管理職によって使用されるノートブック・コンピュータは、セキュリティを損なうことなくインタネットを介して通信することができる。

【００２０】

TARP端末およびTARPルータによるIPアドレス変更は、一定の間隔または無作為の間隔で行うことも、あるいは「侵入」が検出されたときに行うこともできる。IPアドレスを変更することにより、どのコンピュータが通信しているかを知ることのできるトラフィック分析が抑制され、侵入に対するある程度の保護ももたらされる。侵入に対する保護の程度は、ホストのIPアドレスが変更される率にほぼ比例する。

【００２１】

前述のように、IPアドレスは侵入に応答して変更することができる。たとえば、ルータがある方法で調べられていることを示す組織だった一連のメッセージによって、侵入を知ることができる。侵入が検出されると、TARP層プロセスは、そのIPアドレスを変更することによってこのイベントに応答することができる。また、TARP層プロセスは、最初のIPアドレスを維持し、ある方法で侵入者との対話を続けるサブプロセスを作成することができる。

【００２２】

デコイ・パケットは、アルゴリズムによって決定されるある基準に従って各TARP端末によって生成することができる。たとえば、アルゴリズムは、端末がアイドル状態であるときに無作為にパケットを生成することを要求するランダム・アルゴリズムでよい。あるいは、アルゴリズムは、時間または低トラフィックの検出に応答して低トラフィック時により多くのデコイ・パケットを生成することができる。パケットが好ましくは、1つずつ生成されるのではなく、実際のメッセージをシミュレートするようなサイズのグループとして生成されることに留意されたい。また、デコイ・パケットを通常のTARPメッセージ・ストリームに挿入できるように、バックグラウンド・ループは、メッセージ・ストリームが受信されているときにデコイ・パケットを挿入する可能性を高くするラッチを有することができる。あるいは、多数のデコイ・パケットが正規のTARPパケットと共に受信される場合、アルゴリズムは、これらのデコイ・パケットを転送するのではなくそれらを除去する率を高くすることができる。このようにデコイ・パケットを除去し生成すると、見掛けの着信メッセージ・サイズが見掛けの発信メッセージ・サイズと異なるものになり、トラフィック分析を無効にすることができる。

【００２３】
本発明の他の様々な態様では、ネットワーク内の各通信ノード・ペアに複数のIPアドレスが事前に割り当てられるシステムのスケーリング可能なバージョンを構成することができる。各ノード・ペアは、IPアドレス（送信側と受信側の両方）間の「ホッピング」に関するアルゴリズムに合意し、したがって、盗聴者は、通信ノード・ペアの間で送信されるパケットにおいて見掛け上連続する無作為のIPアドレスを見る。各ノードは、合意されたアルゴリズムによる妥当な送信元／着信先ペアを含むことを検証するに過ぎないので、同じサブネット上の数人の異なるユーザに重複するIPアドレスまたは「再使用可能な」IPアドレスを割り付けることができる。好ましくは、IPブロック・サイズが限られているか、あるいはセッションが長いために必要とされないかぎり、所与のエンド・ツー・エンド・セッション中に2つのノード間で送信元／着信先ペアを再使用することはない。
【図面の簡単な説明】
【００２４】
【図１】従来技術の態様によるインタネットを介した安全な通信の図である。
【図２】本発明の態様によるインタネットを介した安全な通信の図である。
【図３Ａ】本発明の態様によるトンネル化IPパケットを形成するプロセスの図である。
【図３Ｂ】本発明の他の態様によるトンネル化IPパケットを形成するプロセスの図である。
【図４】本発明を実施するために使用できるプロセスのOSI層位置の図である。
【図５】本発明によるトンネル化パケットをルーティングするプロセスを示すフローチャートである。
【図６】本発明の態様による、トンネル化パケットを形成するプロセスを示すフローチャートである。
【図７】本発明の態様による、トンネル化パケットを受信するプロセスを示すフローチャートである。
【図８】クライアントとTARPルータとの間で安全なセッションを確立し同期させるにはどうすべきかを示す図である。
【図９】各コンピュータ内の送信テーブルおよび受信テーブルを使用したクライアント・コンピュータとTARPルータとの間のIPアドレス・ホッピング方式を示す図である。
【図１０】３つのインタネット・サービス・プロバイダ（ISP）およびクライアント・コンピュータの間の物理リンク冗長性を示す図である。
【図１１】イーサネット（登録商標）・フレームなど単一の「フレーム」に複数のIPパケットを埋め込むにはどうすべきかを示し、ディスクリミネータ・フィールドを使用して真のパケット受信側を隠すにはどうすべきかをさらに示す図である。
【図１２Ａ】ホップされるハードウェアのアドレス、ホップされるIPアドレス、およびホップされるディスクリミネータ・フィールドを使用するシステムを示す図である。
【図１２Ｂ】ハードウェア・アドレス、IPアドレス、ディスクリミネータ・フィールドの組合せをホップするためのいくつかの異なる手法を示す図である。
【図１３】部分的に公開される同期値を使用することによって送信側と受信側との間の同期を自動的に再確立する技法を示す図である。
【図１４】送信側と受信側との間の同期を回復する「チェックポイント」方式を示す図である。
【図１５】図14のチェックポイント方式の詳細を示す図である。
【図１６】２つのアドレスを複数のセグメントに分解して存在ベクトルと比較するにはどうすべきかを示す図である。
【発明を実施するための形態】
【００２５】
態様の詳細な説明
図2を参照するとわかるように、インタネットを介して通信する安全機構は、各ルータが、1つまたは複数のIPアドレスを有しており、かつ通常のIPプロトコルを使用して、TAR

Pパケット140と呼ばれる、通常のパケットと同様なIPパケット・メッセージを送信するという点で、正規のIPルータ128～132と類似している、TARPルータ122～127と呼ばれるいくつかの特殊なルータまたはサーバを使用する。TARPパケット140は、それぞれが通常のIPパケットと同様に着信先アドレスを含むので、正規のIPルータ128～132によってルーティングされる通常のIPパケット・メッセージと同一である。しかし、TARPパケット140　IPヘッダは、IPヘッダの着信先フィールドで最終着信先を示す示すのではなく、常に、一連のTARPルータ・ホップ内の次のホップまたは最終着信先、すなわちTARP端末110を指し示す。TARPパケットのヘッダが次のホップ着信先のみを含むため、着信先は常に、次のホップTARPルータおよび最終着信先、すなわちTARP端末110であるので、傍受されたTARPパケットにはそのTARPパケットの真の着信先を明白に示すものがない。

【００２６】

各TARPパケットの真の着信先は、リンク鍵146を使用して生成された暗号化アウタ・レイヤに隠される。リンク鍵146は、発信元TARP端末100と着信先TARP端末110を接続するホップのチェーン内の単一のリンクのエンド・ポイント（TARP端末またはTARPルータ）間の暗号化された通信に使用される暗号化鍵である。各TARPルータ122～127は、チェーン内の前のホップとに通信に使用するリンク鍵146を使用してTARPパケットの真の着信先を知ることができる。受信側TARPまたはルーティング端末は、TARPパケットの暗号化アウタ・レイヤを復号するのに必要なリンク鍵を識別するために、平文IPヘッダの送信側フィールドによって（使用されるリンク鍵を示すことのできる）送信側端末を識別することができる。あるいは、平文IPヘッダ内の利用可能なビット内の他の暗号化レイヤにこのIDを隠すことができる。各TARPルータは、TARPメッセージを受信すると、TARPパケット内の認証データを使用することによって、そのメッセージがTARPメッセージであるかどうかを判定する。これは、TARPパケットのIPヘッダ内の利用可能なバイトに記録することができる。あるいは、リンク鍵146を使用して復号を試み、結果が予期されていた結果であるかどうかを判定することによって、TARPパケットを認証することができる。この方法は復号プロセスを伴わないので計算面の利点を有する。

【００２７】

TARPルータ122～127によって復号アウタ・レイヤが完成した後、TARPルータは最終着信先を判定する。システムは好ましくは、トラフィック分析を無効にする助けになるように各TARPパケット140に最小数のホップを受けさせるように構成される。TARPメッセージのIPヘッダ内のタイム・ツー・リブ・カウンタを使用して、完了すべき残りのTARPルータ・ホップの数を示すことができる。各TARPルータは次いで、このカウンタを減分し、それにより、TARPパケット140を他のTARPルータ122～127に転送すべきか、それとも着信先TARP端末110に転送すべきかを判定する。タイム・ツー・リブ・カウンタが減分後にゼロ以下になった場合、使用例として、TARPパケット140を受信したTARPルータは、このTARPパケット140を着信先TARP端末110に転送することができる。タイム・ツー・リブ・カウンタが減分後にゼロを超えた場合、使用例として、TARPパケット140を受信したTARPルータは、この現在のTARP端末によって無作為に選択されたTARP端末122～127にこのTARPパケット140を転送することができる。その結果、各TARPパケット140は、無作為に選択されたTARPルータ122～127の最小数のホップを通してルーティングされる。

【００２８】

したがって、各TARPパケットは、インタネット内のトラフィックを判定する従来の因子とは無関係に、着信先に到着する前に地理的に異なるいくつかのルータの間を無作為に移動し、各移動は、上述のように独立に無作為に決定されるので、所与のメッセージを構成する各パケットごとに異なる可能性が非常に高くなる。この機能をアジル・ルーティングと呼ぶ。まもなく明らかになる理由により、様々なパケットがそれぞれの異なる経路をとるため、侵入者が、マルチパケット・メッセージ全体を形成するすべてのパケットを得ることは困難になる。アジル・ルーティングは、この目的を促進する他の機能、すなわち、あらゆるメッセージが複数のパケットに分割されるようにする機能と組み合される。

【００２９】

TARPルータは、TARPルータによって使用されているIPアドレスがTARPパケットのIPヘッダIPC内のIPアドレスと一致するときにTARPパケットを受信する。しかし、TARPルータのIPアドレスは一定でなくてよい。侵入を回避し管理するために、各TARPルータは、独立して、あるいは他のTARP端末またはTARPルータからの指示の下で、IPアドレスを変更することができる。変更可能な別の識別子またはアドレスも定義される。このアドレスは、TARPアドレスと呼ばれ、TARPルータおよびTARP端末のみに知られ、いつでもTARPルータまたはTARP端末によって参照テーブル（LUT）を使用して相関付けることができる。TARPルータまたはTARP端末は、IPアドレスを変更したときに、他のTARPルータまたはTARP端末を更新し、これらのTARPルータまたはTARP端末はそれぞれのLUTを更新する。実際、TARPルータは、暗号化されたヘッダ内の着信先のアドレスを参照するときは常に、ルータ自体のLUTを使用してTARPアドレスを実際のIPアドレスに変換しなければならない。

【００３０】

TARPパケットを受信したあらゆるTARPルータは、パケットの最終着信先を判定することができるが、メッセージ・ペイロードは、セッション鍵を使用しないかぎりロック解除できないTARPパケット内の暗号化インナレイヤに埋め込まれる。発信元TARP端末100と着信先TARP端末110との間に介在するTARPルータ122～127はセッション鍵を利用することができない。セッション鍵を使用してTARPパケット140のペイロードを復号し、メッセージ全体を再構成することができる。

【００３１】

一態様では、リンク鍵およびセッション鍵を使用して通信を公開されないようにすることができ、任意の所望の方法に従ってリンク鍵およびセッション鍵を共用し使用することができる。たとえば、公開鍵法を使用して、リンク・エンドポイントまたはセッション・エンドポイント間で公開鍵または対称鍵を伝達することができる。許可されたコンピュータのみがTARPパケット140内のプライベート情報にアクセスできるようにデータのセキュリティを確保する他の様々な機構のいずれかを必要に応じて使用することができる。

【００３２】

図3Aを参照するとわかるように、一連のTARPパケットを構成する場合、IPパケット207a、207b、207cなどのデータ・ストリーム300、すなわち、ネットワーク（IP）層プロセスによって形成された一連のパケットが、一連の小さなセグメントに分割される。この例では、等しいサイズのセグメント1～9が定義され、これらを使用して、1組のインタリーブされたデータ・パケットA、B、およびCが構成される。この場合、形成されるインタリーブされたパケットA、B、およびCの数を3つと仮定し、インタリーブされた3つのパケットA、B、およびCを形成するために使用されるIPパケット207a～207cの数も3つと仮定する。もちろん、インタリーブされたパケットのグループに展開されるIPパケットの数は、着信データ・ストリームが展開されたインタリーブされたパケットの数と同様に任意の好都合な数でよい。データ・ストリームが展開されたインタリーブされたパケットの数をインタリーブウインドウと呼ぶ。

【００３３】

送信側ソフトウェアは、パケットを作成する場合、通常のIPパケット207aおよび後続のパケットをインタリーブして、新しい1組のインタリーブされたペイロード・データ320を形成する。このペイロード・データ320は次いで、各データA、B、およびCがTARPパケットのペイロードを形成する、セッション鍵で暗号化された1組のペイロード・データ330を、セッション鍵を使用して形成することによって暗号化される。最初のパケット207a～207cのIPヘッダ・データを使用して、新しいTARPパケットIP_Tが形成される。TARPパケットIP_Tは、通常のIPヘッダと同一でよく、あるいは何らかの方法でカスタマイズすることができる。好ましい態様では、TARPパケットIP_Tは、メッセージのルーティングおよび再構成に必要な以下の情報を提供する追加のデータを含むIPヘッダである。このデータのいくつかは通常、通常のIPヘッダに含まれており、あるいは通常のIPヘッダに含めることができる。

1．ウインドウシーケンス番号－最初のメッセージ・シーケンス内でパケットがどこに

属するかを示す識別子。
2．インタリーブ・シーケンス番号－パケットをインタリーブウインドウ内の他のパケットと共にインタリーブ解除できるようにパケットを形成するために使用されるインタリーブ・シーケンスを示す識別子。
3．タイム・ツー・リブ（TTL）データ－パケットがその着信先に到着する前に実行すべきTARPルータ・ホップの数を示す。TTLパラメータが、パケットを着信先にルーティングすべきか、それとも他のホップにルーティングすべきかを判定するための確率公式で使用すべきデータを提供できることに留意されたい。
4．データ・タイプ識別子－ペイロードが、たとえばTCPデータを含むべきか、それともUDPデータを含むべきかを示す。
5．送信側のアドレス－TARPネットワーク内の送信側のアドレスを示す。
6．着信先アドレス－TARPネットワーク内の着信先端末のアドレスを示す。
7．デコイ／実－パケットが実際のメッセージ・データを含むか、それともダミー・デコイ・データを含むか、それとも組合せを含むかのインディケータ。

【００３４】
自明のことながら、単一のインタリーブウインドウに入るパケットは、共通の着信先を有するパケットのみを含まなければならない。したがって、図の例では、IPパケット207a～207cのIPヘッダはすべて、同じ着信先アドレスを含むか、あるいはインタリーブ解除できるように少なくとも同じ端末によって受信されるものと仮定されている。他の場合に所与のメッセージのサイズによって必要とされるよりも大きなインタリーブウインドウを形成するようにダミーまたはデコイ・データまたはパケットを追加できることに留意されたい。デコイ・データまたはダミー・データをストリームに付加してネットワーク上の負荷を一様にすることにより、トラフィック分析を無効にする助けにすることができる。したがって、インタネット内のある点での通信バーストを他の点での通信バーストに結合して通信エンドポイントを知ることができなくなるように、時間またはその他の基準に応答して低トラフィック期間中により多くのデコイ・データを生成する能力をTARPプロセスに付与することが望ましい。

【００３５】
ダミー・データは、データをより多くの目立たないサイズのパケットに分割して、インタリーブウインドウのサイズを大きくすることを可能にし、同時に各パケットごとに合理的なサイズを維持するうえでも助けになる。（パケット・サイズは、単一の標準サイズでよく、あるいは一定の範囲のサイズから選択することができる。）各メッセージを複数のパケットに分割することが望ましい1つの主要な理由は、インタリーブの前にチェーン・ブロック暗号化方式を使用して第1の暗号化レイヤが形成される場合に明らかである。メッセージの一部または全体に単一のブロック暗号化を適用し、次いでその部分または全体をインタリーブしていくつかの別々のパケットを得ることができる。

【００３６】
図3Bを参照するとわかるように、TARPパケット構成の代替態様では、一連のIPパケットが蓄積され所定のインタリーブウインドウが形成される。パケットのペイロードを使用し、セッション鍵を使用して、チェーン・ブロック暗号化用の単一のブロック520が構成される。ブロックを形成するために使用されるペイロードは、同じ端末を着信先とするものと仮定する。ブロック・サイズは、図3Bの態様例に示されたインタリーブウインドウと一致することができる。暗号化の後で、暗号化されたブロックは別々のペイロードおよびセグメントに分割され、これらのペイロードおよびセグメントが図3Aの態様のようにインタリーブされる。結果として得られるインタリーブされたパケットA、B、およびCは次いで、図3Aの例のようにTARPヘッダを有するTARPパケットとしてパッケージングされる。残りのプロセスは、図3Aに示され、図3Aを参照して論じられるとおりである。

【００３７】
TARPパケット340が形成された後、TARPヘッダIP_Tを含む各TARPパケット340全体が、第1のホップTARPルータと通信するためのリンク鍵を使用して暗号化される。第1のホップTAR

Pルータは無作為に選択される。暗号化された各TARPパケット240に最後の未暗号化IPヘッダIPCが付加され、TARPルータに送信できる通常のIPパケット360が形成される。TARPパケット360を構成するプロセスを前述のように段階的に行う必要がないことに留意されたい。上記の説明は、最終プロダクト、すなわち、TARPパケットを説明するための有用なヒューリスティックに過ぎない。

【００３８】

TARPパケットIP_Tを、上記で識別された情報を含むことを除いて、通常のIPヘッダとはまったく異なる完全なカスタム・ヘッダ構成にすることができることに留意されたい。これは、このヘッダがTARPルータによってのみ解釈されるからである。

【００３９】

上記の方式は、TARPシステムに参加している各サーバまたは端末のデータ・リンク層とネットワーク層との間で動作するプロセスによって完全に実施することができる。図4を参照するとわかるように、TARPトランシーバ405は発信元端末100、着信先端末110、またはTARPルータ122～127でよい。各TARPトランシーバ405において、ネットワーク（IP）層から通常のパケットを受信し、ネットワークを介して伝達できるTARPパケットを生成する送信側プロセスが生成される。TARPパケットを含む通常のIPパケットを受信し、このIPパケットから、ネットワーク（IP）層に「渡される」通常のIPパケットを生成する受信側プロセスが生成される。TARPトランシーバ405がルータである場合、受信されたTARPパケット140は、適切なTARPパケットとして認証され、次いで他のTARPルータまたはTARP着信先端末110に渡されるだけでよいので、IPパケット415のストリームとして処理されないことに留意されたい。介入するプロセス、すなわち「TARP層」420をデータ・リンク層430またはネットワーク層410と組み合わせることができる。いずれの場合も、このプロセスは、埋め込まれたTARPパケットを含む正規のIPパケットを受信し、一連の組立て直されたIPパケットをネットワーク層410に「渡す」ようにデータ・リンク層430に介入する。TARP層420をデータ・リンク層430と組み合わせる例として、プログラムによって、通信カード、たとえばイーサネット（登録商標）・カードを実行する通常のプロセスを拡張することができる。あるいは、TARP層プロセスは、動的にロード可能なモジュールの、ロードされネットワーク層とデータ・リンク層の間の通信をサポートするように実行される部分を形成することができる。

【００４０】

上述の暗号化システムはデータ・リンク層とネットワーク層との間に挿入することができるので、暗号化された通信をサポートするうえで使用されるプロセスは、IP（ネットワーク）層以上でのプロセスに対して完全に透過的であってよい。TARPプロセスは、データ・リンク層に対しても完全に透過的であってよい。したがって、ネットワーク層以上での動作も、データ・リンク層以下での動作も、TARPスタックを挿入することの影響を受けない。これにより、（たとえば、ハッカーによる）ネットワーク層への許可されないアクセスの困難さが大幅に増大するので、ネットワーク層以上でのすべてのプロセスに追加のセキュリティがもたらされる。セッション層で実行される新たに開発されたサーバの場合でも、セッション層よりも下のすべてのプロセスが侵害される可能性がある。このアーキテクチャでは、セキュリティが分散されることに留意されたい。すなわち、たとえば、移動中の管理職によって使用されるノートブック・コンピュータは、セキュリティを損なうことなくインタネットを介して通信することができる。

【００４１】

TARP端末およびTARPルータによるIPアドレス変更は、一定の間隔または無作為の間隔で行うことも、あるいは「侵入」が検出されたときに行うこともできる。IPアドレスを変更することにより、どのコンピュータが通信しているかを知ることのできるトラフィック分析が抑制され、侵入に対するある程度の保護ももたらされる。侵入に対する保護の程度は、ホストのIPアドレスが変更される率にほぼ比例する。

【００４２】

前述のように、IPアドレスは侵入に応答して変更することができる。たとえば、ルータ

がある方法で調べられていることを示す組織だった一連のメッセージによって、侵入を知ることができる。侵入が検出されると、TARP層プロセスは、そのIPアドレスを変更することによってこのイベントに応答することができる。TARPプロセスは、これを行うために、一例としてインタネット制御メッセージ・プロトコル（ICMP）データグラムの形式で、TARPフォーマットのメッセージを構成する。このメッセージは、マシンのTARPアドレス、マシンの前のIPアドレス、およびマシンの新しいIPアドレスを含む。TARP層は、このパケットを少なくとも1つの既知のTARPルータに送信し、このTARPルータは、メッセージを受信し、その妥当性を確認した後、前述のTARPアドレスの新しいIPアドレスでルータ自体のLUTを更新する。TARPルータは次いで、同様なメッセージを作成し、他のTARPルータがLUTを更新できるようにそれらのTARPルータにこのメッセージをブロードキャストする。所与のサブネット上のTARPルータの総数は比較的小さいことが予期されるので、この更新プロセスは比較的高速であるべきである。しかし、このプロセスは、比較的小数のTARPルータおよび／または比較的多数のクライアントがあるときにはうまく作用しないことがある。このため、このアーキテクチャはスケーリング可能性を実現するように改良されている。この改良によって、後述の第2の態様が得られた。

【００４３】

TARPプロセスは、侵入を検出したときに、最初のIPアドレスを維持し、侵入者との対話を続けるサブプロセスを作成することもできる。この対話によって、侵入者を追跡するか、あるいは侵入者の方法を研究する機会を得ることができる（金魚鉢のことを海と「考えている」が、実際には捕えられ観察されている金魚鉢内の小さな魚にたとえて「フィッシュボウリング」と呼ばれる）。侵入者と破棄された（フィッシュボウルされた）IPアドレスとの間の通信の履歴を、人間が分析できるように記録または送信するか、あるいはある方法で応答するためにさらに合成することができる。

【００４４】

上述のように、TARP端末またはTARPルータによって発信データにデコイまたはダミー・データまたはパケットを付加することができる。このようなデコイ・パケットは、より多くの別々のパケットにデータを好都合に分散するだけでなく、トラフィック分析の試みを無効にする助けになるようにインタネットのイナクティブ部分上の付加を均一にすることもできる。

【００４５】

デコイ・パケットは、アルゴリズムによって決定されるある基準に従って各TARP端末100、110または各ルータ122～127によって生成することができる。たとえば、アルゴリズムは、端末がアイドル状態であるときに無作為にパケットを生成することを要求するランダム・アルゴリズムでよい。あるいは、アルゴリズムは、時間または低トラフィックの検出に応答して低トラフィック時により多くのデコイ・パケットを生成することができる。パケットが好ましくは、1つずつ生成されるのではなく、実際のメッセージをシミュレートするようなサイズのグループとして生成されることに留意されたい。また、デコイ・パケットを通常のTARPメッセージ・ストリームに挿入できるように、バックグラウンド・ループは、メッセージ・ストリームが受信されているときにデコイ・パケットを挿入する可能性を高くするラッチを有することができる。すなわち、一連のメッセージが受信されるときに、デコイ・パケット生成率を高めることができる。あるいは、多数のデコイ・パケットが正規のTARPパケットと共に受信される場合、アルゴリズムは、これらのデコイ・パケットを転送するのではなくそれらを除去する率を高くすることができる。このようにデコイ・パケットを除去し生成すると、見掛けの着信メッセージ・サイズが見掛けの発信メッセージ・サイズと異なるものになり、トラフィック分析を無効にすることができる。デコイ・パケットであるか、それとも他のパケットであるかにかかわらず、パケットの受信率を、ペリッシャブル・デコイ・正規パケット・カウンタを通してデコイ・パケット除去プロセスおよびデコイ・パケット生成プロセスに示すことができる。（ペリッシャブル・カウンタは、急速に連続して増分されたときに大きな値を含み、低速で増分されるかあるいは少ない回数だけ急速に連続して増分されたときには小さな値を含むように時間に応答し

て値をリセットまたは減分するカウンタである。）着信先TARP端末110が、単にパケットをルーティングしており、したがって、着信先端末でないように見えるように、受信されたTARPパケットと数およびサイズの等しいデコイ・パケットを生成できることに留意されたい。

【００４６】

図5を参照するとわかるように、TARPパケットをルーティングする上述の方法で以下の特定のステップを使用することができる。

・S0　デコイIPパケットの生成を決定するアルゴリズムを適用するバックグラウンド・ループ動作が実行される。このループは、暗号化されたTARPパケットが受信されたときに割り込まれる。

・S2　TARPパケットを、リンク鍵を使用して復号することを試みる前に、何らかの方法で調べて認証することができる。すなわち、ルータは、ペイロードに含まれる暗号化されたTARPパケットに付加された平文IPヘッダと共に含まれるあるデータに対して選択された動作を実行することによって、パケットが真正なTARPパケットであることを判定することができる。これによって、真正なTARPパケットではないパケットに復号を実行することを避けることが可能になる。

・S3　TARPパケットが復号され、着信先TARPアドレスと、このパケットがデコイ・パケットであるか、それとも実際のメッセージの一部であるかが明らかになる。

・S4　このパケットがデコイ・パケットである場合、ペリッシャブル・デコイ・カウンタが増分される。

・S5　ルータは、デコイ生成／除去アルゴリズムおよびペリッシャブル・デコイ・カウンタ値に基づいて、パケットがデコイ・パケットである場合には、それを破棄することを選択することができる。受信されたパケットがデコイ・パケットであり、これを破棄すべきであると判定された場合（S6）、制御はステップS0に戻る。

・S7　TARPヘッダのTTLパラメータが減分され、TTLパラメータがゼロより大きいかどうかが判定される。

・S8　TTLパラメータがゼロよりも大きい場合、ルータによって維持されているTARPアドレスのリストからTARPアドレスが無作為に選択され、このTARPアドレスに対応するリンク鍵およびIPアドレスが、このTARPパケットを含む新しいIPパケットを作成する際に使用できるように記憶される。

・S9　TTLパラメータがゼロ以下である場合、着信先のTARPアドレスに対応するリンク鍵およびIPアドレスが、このTARPパケットを含む新しいIPパケットを作成する際に使用できるように記憶される。

・S10　TARPパケットが、記憶されたリンク鍵を使用して暗号化される。

・S11　記憶されたIPアドレスを含むパケットにIPヘッダが付加され、暗号化されたTARPパケットがIPヘッダで覆われ、完成したパケットが次のホップまたは着信先に送信される。

【００４７】

図6を参照するとわかるように、TARPパケットを生成する上述の方法では以下の特定のステップを使用することができる。

・S20　バックグラウンド・ループ動作が、デコイIPパケットの生成を決定するアルゴリズムを適用する。このループは、IPパケットを含むデータ・ストリームが、後で送信できるように受信されたときに割り込まれる。

・S21　受信されたIPパケットが、一定のIP着信先アドレスを有するメッセージから成る集合としてグループ化される。この集合はさらに、インタリーブウインドウの最大サイズに一致するように分割される。集合が暗号化されインタリーブされ、TARPパケットになる予定の1組のペイロードが得られる。

・S22　IPアドレスに対応するTARPアドレスが、参照テーブルから判定され、TARPヘッダを生成するために記憶される。初期TTLカウントが生成され、ヘッダに格納される。TTLカウントは、最小値および最大値を有する無作為の値でも、あるいは一定の値でもよく、あ

るいは他の何らかのパラメータによって決定することができる。
・S23　ウインドウシーケンス番号およびインタリーブ・シーケンスが各パケットのTARPヘッダに記録される。
・S24　各TARPパケットごとに1つのTARPルータ・アドレスが無作為に選択され、このアドレスに対応するIPアドレスが、平文IPヘッダで使用できるように記憶される。このルータに対応するリンク鍵が識別され、インタリーブされ暗号化されたデータおよびTARPヘッダを含むTARPパケットが、このリンク鍵を使用して暗号化される。
・S25　第1のホップ・ルータの実際のIPアドレスを有する平文IPヘッダが生成され、暗号化されたTARPパケットおよび結果として得られるパケットのそれぞれに付加される。
【００４８】
　図7を参照するとわかるように、TARPパケットを受信する上述の方法で以下の特定のステップを使用することができる。
・S40　デコイIPパケットの生成を決定するアルゴリズムを適用するバックグラウンド・ループ動作が実行される。このループは、暗号化されたTARPパケットが受信されたときに割り込まれる。
・S42　TARPパケットが、リンク鍵を使用して復号される前に、調べられ認証される。
・S43　TARPパケットが適切なリンク鍵を用いて復号され、着信先TARPアドレスと、このパケットがデコイ・パケットであるか、それとも実際のメッセージの一部であるかが明らかになる。
・S44　このパケットがデコイ・パケットである場合、ペリッシャブル・デコイ・カウンタが増分される。
・S45　受信側は、デコイ生成／除去アルゴリズムおよびペリッシャブル・デコイ・カウンタ値に基づいて、パケットがデコイ・パケットである場合には、それを破棄することを選択することができる。
・S46　インタリーブウインドウを形成するすべてのパケットが受信されるまでTARPパケットがキャッシュされる。
・S47　インタリーブウインドウのすべてのパケットが受信された後、パケットがインタリーブ解除される。
・S48　次いで、インタリーブウインドウを形成する組み合わされた各パケットのパケット・ブロックが、セッション鍵を使用して復号される。
・S49　次いで、復号されたブロックが、ウインドウシーケンス・データを使用して分割され、IP_rヘッダが通常のIP_Cヘッダに変化される。ウインドウシーケンス番号がIP_Cヘッダに組み込まれる。
・S50　次いで、パケットがIPレイヤ・プロセスに渡される。
【００４９】
スケーリング可能性の向上
　上述のIPアジリティ機能は、IPアドレスの変更をすべてのTARPルータに送信する能力に依存する。この機能を含む各態様は、インタネットなど大規模なネットワーク向けにこのような機能をスケーリングする際の潜在的な制限のために「ブティック」態様と呼ばれる。（しかし、ブティック態様は、たとえば、小規模な仮想専用網など小規模なネットワークで使用する場合にはロバストである）。ブティック態様の1つの問題は、IPアドレスの変更が頻繁に行われる場合、すべてのルータを十分に高速に更新するのに必要なメッセージ・トラフィックのために、TARPルータおよび／またはクライアントの数が非常に多くなるとインタネットに大きな負荷が掛かる。すべてのTARPルータを更新するために使用される帯域幅負荷であって、たとえばICMPパケットにおいてネットワークに加わる帯域幅負荷は、インタネットのスケールに近い大規模なインプリメンテーションの場合にインタネットの能力を超える可能性がある。言い換えれば、ブティック・システムのスケーリング可能性は限られている。
【００５０】
　追加的なメッセージ負荷なしにIPアジリティの利益をもたらすように上記の態様の特徴

のうちのいくつかの兼合いを取るシステムを構成することができる。これは、TARPノードなどのノード間の通信セッションに参加しているリンク間で使用されるIPアドレスを管理する共用アルゴリズムに従ってIPアドレス・ホッピングによって行われる。（IPホッピング技法をブティック態様に適用することもできることに留意されたい。）ブティック・システムに関して論じたIPアジリティ機能は、スケーリング可能なこの方式の下で分散され、かつ上述の共用アルゴリズムによって管理されるように修正することができる。ブティック・システムの他の機能をこの新しい種類のIPアジリティrと組み合わせることができる。

【００５１】

この新しい態様は、通信する各ノード・ペアによって交換されるローカル・アルゴリズムおよび1組のIPアドレスによって管理されるIPアジリティをもたらすという利点を有する。このローカル管理は、直接通信するノード・ペア間で転送されるセッションまたはエンド・ポイントにかかわらず、ノード・ペア間の通信を管理できるという点でセッションに依存しない。

【００５２】

スケーリング可能なこの態様では、ネットワーク内の各ノードにIPアドレスのブロックが割り付けられる。（このスケーリング可能性は、将来において、インタネット・プロトコル・アドレスが128ビット・フィールドに増加し、明確にアドレス可能なノードの数が大幅に増加したときに高くなる）。したがって、各ノードは、そのノードに割り当てられたIPアドレスのいずれかを使用してネットワーク内の他のノードと通信することができる。実際には、通信する各ノード・ペアは複数の送信元IPアドレスおよび着信先IPアドレスを使用して互いに通信することができる。

【００５３】

任意のセッションに参加しているチェーン内の通信する各ノード・ペアは、ネットブロックと呼ばれる2つのIPアドレス・ブロックと、アルゴリズムと、次のメッセージを送信するために使用される次の送信元／着信先IPアドレス・ペアを各ネットブロックごとに選択するための無作為化シードとを記憶する。言い換えれば、このアルゴリズムは、各ネットブロックからのIPアドレス・ペア、1つの送信側IPアドレス、および1つの受信側IPアドレスの順次選択を管理する。アルゴリズムと、シードと、ネットブロック（IPアドレス・ブロック）の組合せを「ホップブロック」と呼ぶ。ルータは別々の送信ホップブロックおよび受信ホップブロックをルータのクライアントに発行する。クライアントによって送信される各発信パケットのIPヘッダの送信アドレスおよび受信アドレスには、アルゴリズムによって管理される送信IPアドレスおよび受信IPアドレスが充填される。アルゴリズムは、ペアが使用されるたびに、アルゴリズムが次に送信すべきパケット用の新しい送信ペアを作成するように、カウンタによって「クロッキング」（インデックス付け）される。

【００５４】

ルータの受信ホップブロックはクライアントの送信ホップブロックと同一である。ルータは、このクライアントからの次に予期されるパケット用の送信IPアドレス・受信IPアドレス・ペアが何であるかを、受信ホップブロックを使用して予想する。各パケットは順序正しく受信されないことがあるので、ルータが、次の順次パケット上にどんなIPアドレスが来るかを確実に予想することは不可能である。ルータは、この問題を考慮して、次に受信されるパケットの後に続く可能性のある送信パケット送信／受信アドレスの数を包含するある範囲の予想を生成する。したがって、所与のパケットが、その前にクライアントによって送信された5つのパケットよりも前にルータに到着する可能性が非常に低い場合、ルータは、次に受信されるパケットと比較すべき一連の6つの送信／受信IPアドレス・ペア（または「ホップウインドウ」）を生成することができる。パケットが受信されると、このパケットは、受信されたものとしてホップウインドウ内にマーク付けされ、したがって、同じIPアドレス・ペアを有する第2のパケットは破棄される。シーケンスからずれたパケットが所定のタイムアウト期間内に到着しない場合、その通信セッションに使用されているプロトコルに応じ、あるいは場合によっては規約によって、そのパケットを再送信

することを要求するか、あるいは単に受信テーブルから破棄することができる。

【００５５】

ルータは、クライアントのパケットを受信すると、パケットの送信IPアドレスおよび受信IPアドレスを、予想される次のN個の送信IPアドレス受信アドレス・ペアと比較し、パケットがこの集合のメンバーではない場合、パケットを拒絶する。ウインドウに収まった予想される送信元／着信先IPアドレスを有さない受信パケットは拒絶され、したがって、可能なハッカーは妨害される。（可能な組合せの数を用いた場合、かなり大きなウインドウであっても無作為にこの中に収めることは困難である。）パケットがこの集合のメンバーである場合、ルータはパケットを受け入れ、さらに処理する。リンク・ベースのこのIPホッピング方式は、「IHOP」と呼ばれ、独立しており、必ずしも上述のブティック・システムの要素を伴わないネットワーク要素である。ブティック態様に関して説明したルーティングアジリティ機能をこのリンク・ベースIPホッピング方式と組み合わせた場合、ルータの次のステップは、TARPヘッダを復号して、パケットの着信先TARPルータを判定し、かつパケットの次のホップを何にすべきかを判定することである。TARPルータは次いで、無作為のTARPルータにパケットを転送するか、あるいは送信先ルータがリンク・ベースのIPホッピング通信を確立させる着信先TARPルータにパケットを転送する。

【００５６】

図8は、クライアント・コンピュータ801およびTARPルータ811が安全なセッションを確立するにはどうすべきかを示している。クライアント801は、TARPルータ811とのHOPセッションを確立する際、TARPルータ811に「安全同期」要求（「SSYN」）パケット821を送信する。このSYNパケット821は、クライアント811の認証トークンを含み、暗号化フォーマットでルータ811に送信することができる。パケット821上の送信先IP番号および着信先IP番号は、クライアント801の現在の固定IPアドレスおよびルータ811の「既知の」固定IPアドレスである。（セキュリティのために、着信先がルータの既知の固定IPアドレスである、ローカル・ネットワークの外部からのあらゆるパケットを拒絶することが望ましい。）ルータ811は、クライアント801のSSYNパケット821を受信し、妥当性を確認した後、暗号化された「安全同期確認応答」（「SSYN　ACK」）822をクライアント801に送信することによって応答する。このSSYN　ACK822は、クライアント801がTARPルータ811と通信するときに使用する送信ホップブロックおよび受信ホップブロックを含む。クライアント801は、その固定IPアドレスからTARPルータ811の既知の固定IPアドレスに送信される暗号化されたSSYN　ACK　ACKパケット823を生成することによって、TARPルータ811の応答パケット822で確認応答する。クライアント801は、SSYN　ACK　ACKパケットを同時に生成する。このSSYN　ACKパケットは、安全セッション初期設定（SSI）パケット824と呼ばれ、TARPルータ811によってSSYN　ACKパケット822内に与えられる送信ホップブロックに指定される、クライアントの送信テーブル921（図9）内の第1の｛送信側、受信側｝IPペアと共に送信される。TARPルータ811は、TARPルータの送信テーブル923内の第1の｛送信側、受信側｝IPペアと共に送信されるSSI　ACKパケット825を用いてSSIパケット824に応答する。これらのパケットが首尾良く交換された後、安全な通信セッションが確立され、クライアント801とTARPルータ811との間の他のすべての安全な通信は、同期が維持されるかぎり、この安全なセッションを介して行われる。同期が失われた場合、クライアント801およびTARPルータ802は、図8に概略的に示し上記で説明した手順によって安全なセッションを再確立することができる。

【００５７】

安全なセッションがアクティブであるとき、クライアント901とTARPルータ911（図9）は共に、セッション同期822中にTARPルータから与えられるそれぞれの送信テーブル921、923および受信テーブル922、924を維持する。クライアントの送信テーブル921内のIPペアのシーケンスがTARPルータの受信テーブル924内のIPペアのシーケンスと同一であることが重要である。同様に、クライアントの受信テーブル922内のIPペアのシーケンスはルータの送信テーブル923内のIPペアのシーケンスと同一でなければならない。このことはセッション同期を維持するうえで必要である。クライアント701は、安全なセッションの間1

つの送信テーブル921および1つの受信テーブル922のみを維持する必要がある。クライアント901によって送信される各順次パケットは、TCPセッションであるか、それともUDPセッションであるかにかかわらず、送信テーブル内の次の｛送信側、受信側｝IPアドレス・ペアを使用する。TARPルータ911は、クライアント911から到着する各パケットが受信テーブル内に示された次のIPアドレスを保持していることを予期する。

【００５８】

しかし、パケットは順序正しく到着しないことがあるので、ルータは受信テーブル内に「ルックアヘッド」バッファを維持することができ、すでに受信されているIPペアを未来のパケットに対して無効なIPペアとしてマーク付けする。IPルック・アヘッド・バッファ内に存在するが、受信済みIPペアとしてマーク付けされているIPペアを含む未来のパケットは破棄される。TARPルータ911からクライアント901への通信も同様に維持される。特に、ルータ911は、クライアント901に送信すべきパケットを構成する際にルータ911の送信テーブル923から次のIPアドレス・ペアを選択し、クライアント901は、それが受信するパケット上の予期されるIPペアのルックアヘッド・バッファを維持する。各TARPルータは、そのTARPルータとの安全なセッションを行っているか、あるいはそのTARPルータを通して安全なセッションを行っている各クライアントごとに別々の送信テーブル・受信テーブル・ペアを維持する。

【００５９】

クライアントは、それをインタネットにリンクするクライアントのホップブロックを第1のサーバから受信し、それに対して、ルータはホップブロックを交換する。ルータが他のルータとのリンク・ベースのIPホッピング通信方式を確立すると、ペア交換の各ルータはその送信ホップブロックを交換する。各ルータの送信ホップブロックは他方のルータの受信ホップブロックになる。ルータ間の通信は、クライアントが第1のルータにパケットを送信する例で説明したように管理される。

【００６０】

上記の方式はIP環境でうまく作用するが、インタネットに接続される多くのローカル・ネットワークはイーサネット（登録商標）・システムである。イーサネット（登録商標）では、既知のプロセス（「アドレス分解プロトコル」および「逆アドレス分解プロトコル」）を使用して着信先装置のIPアドレスをハードウェア・アドレスに変換しなければならず、その逆もまた同様である。しかし、リンク・ベースのIPホッピング方式を使用する場合、相関プロセスが厄介なものになる。リンク・ベースのIPホッピング方式の代替態様はイーサネット（登録商標）・ネットワーク内で使用することができる。この解決策では、インタネットをイーサネット（登録商標）にリンクするノード（ボーダー・ノードと呼ぶ）が、リンク・ベースのIPホッピング通信方式を使用してイーサネット（登録商標）LANの外部のノードと通信できるようにする。イーサネット（登録商標）LAN内で、各TARPノードは、従来どおりにアドレス指定される単一のIPアドレスを有する。LAN内TARPノードは、｛送信側、受信側｝IPアドレス・ペアを比較してパケットを認証するのではなく、1つのIPヘッダ拡張フィールドを使用してパケットを認証する。したがって、ボーダー・ノードは、LAN内TARPノードによって共用されるアルゴリズムを使用して、IPヘッダ内の空きフィールドに格納される記号を生成し、LAN内TARPノードは、この特定の送信元IPアドレスから次に受信されることが予期されるパケットについてのノード自体の予想に基づいてある範囲の記号を生成する。パケットは、予想される記号（たとえば、数値）の集合内に収まらない場合には拒絶され、収まった場合には受け入れられる。LAN内TARPノードからボーダー・ノードへの通信も同様に行われる。ただし、セキュリティ上の理由で、アルゴリズムは必然的に異なる。したがって、各通信ノードは、図9と同様に送信テーブルおよび受信テーブルを生成する。すなわち、LAN内　TARPノードの送信テーブルはボーダー・ノードの受信テーブルと同一であり、LAN内　TARPノードの受信テーブルはボーダー・ノードの送信テーブルと同一である。

【００６１】

IPアドレス・ホッピングに使用されるアルゴリズムは任意の所望のアルゴリズムでよい

。たとえば、アルゴリズムは、所与のシードを有する許可されたIPアドレスをカバーする範囲の番号を生成する擬似乱数生成プログラムでよい。あるいは、セッション参加者が、ある種のアルゴリズムを仮定し、単にそのアルゴリズムを適用するためのパラメータを指定することができる。たとえば、仮定されるアルゴリズムは特定の擬似乱数生成プログラムでよく、セッション参加者は単にシード値を交換することができる。

【００６２】

発信元端末ノードと着信先端末ノードとの間に永久的な物理的な違いはないことに留意されたい。いずれのエンド・ポイントのいずれの装置もペアの同期を開始することができる。別々のメッセージ交換が必要にならないように認証／同期要求（および確認応答）およびホップブロック交換がすべて単一のメッセージによって実行できることにも留意されたい。

【００６３】

前述のアーキテクチャの他の拡張態様として、リンク冗長性を実現し、さらに、サービスを拒否する試みおよびトラフィック監視を妨げるために、クライアントによって複数の物理パスを使用することができる。図10に示すように、たとえば、クライアント1001は、それぞれの異なるISP1011、1012、1013から与えられる3つのTARPルータのそれぞれとの3つの同時セッションを確立することができる。一例として、クライアント1001は3つの異なる電話回線1021、1022、1023や2つの電話回線およびケーブル・モデムなどを使用してISPに接続することができる。この方式では、送信されるパケットが様々な物理パスの間で無作為に送信される。このアーキテクチャは高度の通信冗長性を実現し、サービス拒否アタックおよびトラフィック監視に対する保護を向上させる。

【００６４】

他の拡張態様

以下に、上述の技法、システム、および方法の様々な拡張態様について説明する。上述のように、コンピュータ・ネットワーク（インタネット、イーサネット（登録商標）など）内のコンピュータ間で行われる通信のセキュリティは、ネットワークを介して送信されるデータ・パケットの、見掛け上無作為の送信元インタネット・プロトコル（IP）アドレスおよび着信先IPアドレスを使用することによって向上させることができる。この機能は、盗聴者が、ネットワーク内のどのコンピュータが通信しているかを判定するのを妨げ、同時に、通信している2つのコンピュータが、受信された所与のデータ・パケットが正当なパケットであるか否かを容易に認識できるようにする。上述のシステムの一態様では、IPヘッダ拡張フィールドを使用してイーサネット（登録商標）上の着信パケットが認証される。

【００６５】

本明細書で説明する前述の技法の様々な拡張態様には、（1）ホップされるハードウェアまたは「MAC」アドレスをブロードキャスト型ネットワークで使用すること、（2）コンピュータが送信側との同期を自動的に回復できるようにする自己同期技法、（3）送信側コンピュータおよび受信側コンピュータがパケット喪失イベントまたはその他のイベントの場合に同期を迅速に再確立できるようにする同期アルゴリズム、および（4）無効なパケットを拒絶する高速パケット拒絶機構が含まれる。これらの拡張態様のいずれかまたはすべてを様々な方法のいずれかで上述の機能と組み合わせることができる。

【００６６】

A．ハードウェア・アドレス・ホッピング

LAN上のインタネット・プロトコル・ベース通信技法または任意の専用物理媒体を介したインタネット・プロトコル・ベース通信技法では通常、「フレーム」と呼ばれることの多い下位パケット内にIPパケットが埋め込まれる。図11に示すように、たとえば、第1のイーサネット（登録商標）・フレーム1150はフレーム・ヘッダ1101および埋め込まれた2つのIPパケットIP1およびIP2を備え、それに対して、第2のイーサネット（登録商標）・フレーム1160は異なるフレーム・ヘッダ1104および単一のIPパケットIP3を備えている。各フレーム・ヘッダは一般に、送信元ハードウェア・アドレス1101Aおよび着信先ハード

ウェア・アドレス1101Bを含む。図11では、図を明確にするためにフレーム・ヘッダ内の他の公知のフィールドは省略されている。物理通信チャネルを介して通信する2つのハードウェア・ノードは、チャネルまたはネットワーク上のどのノードがフレームを受信すべきかを示す適切な送信元ハードウェア・アドレスおよび着信先ハードウェア・アドレスを挿入する。

【００６７】

悪意ある盗聴者が、IPパケット自体ではなく（あるいはそれに加えて）ローカル・ネットワーク上のフレームを調べることによって、フレームの内容および／またはそのフレームの通信者に関する情報を得ることが可能である。このことは、フレームを生成したマシンのハードウェア・アドレスおよびフレームが送信されるマシンのハードウェア・アドレスをフレーム・ヘッダに挿入する必要がある、イーサネット（登録商標）などのブロードキャスト媒体に特に当てはまる。ネットワーク上のすべてのノードは場合によっては、ネットワークを介して送信されるすべてのパケットを見ることができる。これは、特に、情報交換を行っているのは誰かを第三者が識別できることを通信者が望んでいない場合に、安全な通信に対する問題となる恐れがある。この問題に対処する1つの方法は、アドレス・ホッピング方式をハードウェア層に拡張することである。本発明の様々な態様によれば、ハードウェア・アドレスは、IPアドレスを変更するために使用される方法と同様な方法で「ホップされ」、したがって、盗聴者は、特定のメッセージを生成したのはどのハードウェア・ノードであるかを判定することも、あるいは所期されている受信側はどのノードであるかを判定することもできない。

【００６８】

図12Aは、イーサネット（登録商標）などのネットワーク上のセキュリティを向上させるために媒体アクセス制御（「MAC」）ハードウェア・アドレスが「ホップされる」システムを示している。この説明ではイーサネット（登録商標）環境の例示的なケースを引用するが、本発明の原則は他の種類の通信媒体にも同様に適用することができる。イーサネット（登録商標）の場合、送信側および受信側のMACアドレスは、イーサネット（登録商標）・フレームに挿入され、そのフレームのブロードキャスト範囲内にいるLAN上のあらゆる人によって見ることができる。安全な通信を実現する場合、特定の送信側や受信側に帰属しないMACアドレスを持つフレームを生成することが望ましくなる。

【００６９】

図12Aに示すように、コンピュータ・ノード1201および1202はイサーネットなどの通信チャネルを介して通信する。各ノードは、それぞれ通信ソフトウェア1204および1217によりパケットを送信することによって通信する1つまたは複数のアプリケーション・プログラム1203および1218を実行する。アプリケーション・プログラムの例にはビデオ会議、eメール、文書処理プログラムなどが含まれる。通信ソフトウェア1204および1217は、たとえば、様々な機能レベルで実現される様々なサービスを標準化するOSI層化アーキテクチャまたは「スタック」を備えることができる。

【００７０】

通信ソフトウェア1204および1217の最下位レベルはそれぞれ、ハードウェア構成要素1206および1214と通信し、各構成要素は、様々な通信プロトコルに従ってハードウェアを再構成または制御できるようにする1つまたは複数のレジスタ1207および1215を含むことができる。ハードウェア構成要素（たとえば、イーサネット（登録商標）・ネットワーク・インタフェース・カード）は通信媒体を介して互いに通信する。各ハードウェア構成要素には通常、そのハードウェア構成要素をネットワーク上の他のノードへに対して識別する固定ハードウェア・アドレスまたはMAC番号が事前に割り当てられる。以下に詳しく説明するように、本発明の原則の様々な態様は、1つまたは複数のアルゴリズムと、受信されたパケットの妥当性を確認するためにある範囲の妥当なアドレスを追跡する1つまたは複数の移動ウインドウを使用して、様々なアドレスの「ホッピング」を可能にする。本発明の1つまたは複数の原則に従って送信されるパケットは一般に、マシンによって相関付けされた通常のアドレスを使用して平文で送信される通常のデータ・パケットと区別するた

めに「安全な」パケットまたは「安全な通信」と呼ばれる。

【００７１】

帰属不能なMACアドレスを生成する1つの簡単な方法はIPホッピング方式の拡張態様である。この場合、同じLAN上の2つのマシンは、安全に通信して乱数発生プログラムおよびシードを交換し、同期式ホッピングのための準無作為MACアドレスのシーケンスを作成する。この場合、インプリメンテーションおよび同期の問題はIPホッピングの同じ問題と類似している。

【００７２】

しかし、この手法では、LAN上で現在アクティブなMACアドレスが使用される恐れがあり、それによって、これらのマシンの通信が中断される可能性がある。イーサネット（登録商標）MACアドレスが現在の所、長さが48ビットであるので、アクティブなMACアドレスが無作為に乱用される可能性は実際には極めて低い。しかし、この数字に（広範囲のLANで見られるような）多数のノードの数、（パケット音声またはストリーミング・ビデオの場合のような）多数のフレームの数、および多数の並行仮想専用網（VPN）の数を乗じた場合、アドレス・ホッピングされるフレームで安全でないマシンのMACアドレスが使用される可能性は無視できないものになる。簡単に言えば、LAN上の他のマシンの通信を中断する可能性が少しでもあるあらゆる方式は、先見の明のあるシステム管理者から反対を受ける。それにもかかわらず、これは技術的に実施可能であり、マシンの数が少ないLAN上で安全に実施することができ、あるいはLAN上のすべてのマシンがMACホップ通信を行う場合に安全に実施することができる。

【００７３】

同期式MACアドレス・ホッピングは、セッションを確立する際、特に通信に関与する複数のセッションまたは複数のノードがある場合、ある程度のオーバヘッドを伴うことがある。MACアドレスを無作為化するより簡単な方法は、各ノードがネットワーク上で発生したあらゆるフレームを受信し処理できるようにすることである。通常、各ネットワーク・インタフェース・ドライバは、発生したあらゆるフレームのヘッダ内の着信先MACアドレスを検査し、そのマシンのMACアドレスに一致するかどうかを調べる。一致しない場合、このフレームは破棄される。しかし、一態様では、これらの検査を無効化することができ、発生したあらゆるパケットがTARPスタックに渡され処理される。これは、発生したあらゆるフレームが処理されるので「プロミスキュアス」モードと呼ばれる。プロミスキュアス・モードでは、着信先マシンがフレームを確実に処理できるので、送信側は、同期の取れていない完全に無作為のMACアドレスを使用することができる。パケットの着信先が本当にそのマシンであるかどうかに関する決定はTARPスタックによって処理され、TARPスタックは、送信元IPアドレスと着信先IPアドレスがTARPスタックのIP同期テーブルにおいて一致しているかどうかを検査する。一致が見つからない場合、このパケットは破棄される。一致した場合、パケットが開放され、インナヘッダが評価される。パケットの着信先がこのマシンであることをインナヘッダが示している場合、パケットがIPスタックに転送され、そうでない場合、パケットは破棄される。

【００７４】

純粋に無作為のMACアドレス・ホッピングの1つの欠点は、処理オーバヘッドに対するこのホッピングの影響である。すなわち、発生したあらゆるフレームを処理しなければならないので、ネットワーク・インタフェース・ドライバがパケットを一方的に区別し拒絶する場合よりもかなり頻繁にマシンのCPUが使用される。妥協的な手法として、メッセージの着信先である実際の受信側にかかわらず、MACホップ通信に使用すべきアドレスとして単一の固定MACアドレスまたは少数のMACアドレス（たとえば、イーサネット（登録商標）上の各仮想専用網ごとに1つのMACアドレス）を選択する手法がある。このモードでは、ネットワーク・インタフェースが、発生した各フレームを事前に確立された1つ（または少数）のMACアドレスと突き合わせて検査することができ、それによって、CPUは物理層パケットの区別を行わなく済む。この方式では、LAN上の侵入者に重要な情報が漏れることがない。特に、アウタヘッダ内の固有のパケット・タイプによってあらゆる安全なパケット

を事前に識別しておくことができる。しかし、安全な通信を行うすべてのマシンが同じMACアドレスを使用するか、あるいは所定のMACアドレスの小さな集合から選択するので、特定のマシンと特定のMACアドレスとの間の関連付けが実際上、失われる。

【００７５】

この方式では、ネットワーク・インタフェース・ドライバが、このマシンを着信先とする安全なパケットと他のVPNからの安全なパケットを常に一方的に区別することはできないので、CPUは、安全でない通信（または同期式MACアドレスホッピング）の場合よりも頻繁に使用される。しかし、ネットワーク・インタフェースにおいて安全でないトラフィックをなくすのは容易であり、したがって、CPUに必要な処理の量が少なくなる。これらが当てはまらない境界条件があり、たとえば、LAN上のすべてのトラフィックが安全なトラフィックである場合、CPUは純粋に無作為のアドレス・ホッピングの場合と同じ程度に使用される。あるいは、LAN上の各VPNが異なるMACアドレスを使用する場合、ネットワーク・インタフェースは、ローカル・マシンを着信先とする安全なフレームを、他のVPNを構成する安全なフレームと完全に区別することができる。これらは技術上の兼ね合せであり、ユーザがソフトウェアをインストールし、かつ／またはVPNを確立する際に管理オプションを与えることによって最もうまく処理することができる。

【００７６】

しかし、この場合でも、LAN上の1つまたは複数のノードによって使用されるMACアドレスが選択されるわずかな可能性が依然として残る。この問題に対する1つの解決策として、MACホップ通信で使用される1つのアドレスまたはある範囲のアドレスが公式に割り当てられる。これは通常、割当て番号登録権限を介して行われ、たとえば、イーサネット（登録商標）の場合、電気電子技術者協会（IEEE）によってベンダにMACアドレス範囲が割り当てられている。公式に割り当てられるアドレス範囲によって、安全なフレームはLAN上の適切に構成され適切に機能するマシンに確実に適合する。

【００７７】

次に、本発明の原則に従った多数の組合せおよび特徴について説明するために図12Aおよび図12Bを参照する。上述のように、2つのコンピュータ・ノード1201および1202がイーサネット（登録商標）などのネットワークまたは通信媒体を介して通信しているものと仮定する。各ノードの通信プロトコル（それぞれ、1204および1217）は、標準通信プロトコルから導かれるある機能を実行する修正された要素1205および1216を含む。特に、コンピュータ・ノード1201は、各パケットを他方のコンピュータ・ノードに送信するために見掛け上無作為の送信元IPアドレスおよび着信先IPアドレス（および一態様では、見掛け上無作為のIPヘッダ・ディスクリミネータ・フィールド）を選択する第1の「ホップ」アルゴリズム1208Xを実施する。たとえば、ノード1201は、送信先（S）、着信先（D），および発信IPパケット・ヘッダに挿入されるディスクリミネータ・フィールド（DS）の3つ組を含む送信テーブル1208を維持する。このテーブルは、受信側ノード1202に知られている適切なアルゴリズム（たとえば、適切なシードを用いてシードされる乱数生成プログラム）を使用することによって生成される。新しい各IPパケットが形成される際、送信側の送信テーブル1208の順次エントリを使用してIP送信元、IP着信先、およびIPヘッダ拡張フィールド（たとえば、ディスクリミネータ・フィールド）が埋められる。送信テーブルを事前に作成しておく必要がなく、その代わり、動作時に、各パケットを形成する際にアルゴリズムを実行することによって作成できることが理解されよう。

【００７８】

受信側ノード1202では、同じIPホップ・アルゴリズム1222Xが維持され、送信元IPアドレス、着信先IPアドレス、およびディスクリミネータ・フィールドの妥当な3つ組をリストした受信テーブル1222を、上記のアルゴリズムを使用して生成するために使用される。これは、送信テーブル1208の第1の5つのエントリが、受信テーブル1222の第2の5つのエントリに一致することによって示されている。（各テーブルは、パケットの喪失、パケットの順序のずれ、または送信遅延のために任意の特定の時間にわずかにずれる可能性がある）。また、ノード1202は、着信IPパケットの一部として受信されたときに受け入れられる

妥当なIP送信元、IP着信先、およびディスクリミネータ・フィールドのリストを表す受信ウインドウW3を維持する。パケットが受信されると、ウインドウW3は妥当なエントリのリストを下にスライドさせ、したがって、可能な妥当なエントリは時間の経過と共に変化する。誤った順序で到着したが、それにもかかわらずウインドウW3内のエントリと一致している2つのパケットは受け入れられる。ウインドウW3に収まらないパケットは無効なパケットとして拒絶される。ウインドウW3の長さは、ネットワーク遅延またはその他の因子を反映する必要に応じて調整することができる。

【００７９】

ノード1202は、場合によっては異なるホッピング・アルゴリズム1221Xを使用して着信先がノード1201であるIPパケットおよびフレームを作成するために同様な送信テーブル1221を維持し、ノード1201は、同じアルゴリズム1209Xを使用して一致する受信テーブル1209を維持する。ノード1202が見掛け上無作為のIP送信先、IP着信先、および／またはディスクリミネータ・フィールドを使用してノード1201にパケットを送信すると、ノード1201は着信パケット値を、ノード1201の受信テーブルに維持されているウインドウW1内に収まる値と突き合わせる。実際には、ノード1201の送信テーブル1208と受信側ノード1202の受信テーブル1222との同期が取られる（すなわち、同じ順序でエントリが選択される）。同様に、ノード1202の送信テーブル1221とノード1201の受信テーブル1209との同期が取られる。図12Aでは送信元、着信先、およびディスクリミネータ・フィールドについて共通のアルゴリズムが示されている（たとえば、3つのフィールドのそれぞれに異なるシードを使用する）が、実際には、まったく異なるアルゴリズムを使用してこれらのフィールドのそれぞれの値を確立できることが理解されよう。図のように3つのフィールドすべてではなく1つまたは2つのフィールドを「ホップ」できることも理解されよう。

【００８０】

本発明の他の態様によれば、ローカル・エリア・ネットワークまたはブロードキャスト型ネットワーク内のセキュリティを向上させるために、IPアドレスおよび／またはディスクリミネータ・フィールドの代わりにあるいはそれらと共にハードウェア・アドレスまたは「MAC」アドレスがホップされる。この目的のために、ノード1201は、ノード1202にある対応する受信テーブル1224との同期が取られるフレーム・ヘッダ（たとえば、図11のフィールド1101Aおよび1101B）に挿入される送信元ハードウェア・アドレスおよび着信先ハードウェア・アドレスを、送信アルゴリズム1210Xを使用して生成する、送信テーブル1210をさらに維持する。同様に、ノード1202は、ノード1201にある対応する受信テーブル1211との同期が取られる送信元ハードウェア・アドレスおよび着信先ハードウェア・アドレスを含む異なる送信テーブル1223を維持する。このように、発信ハードウェア・フレームは、各受信側が、所与のパケットの着信先が該受信側であるかどうかを判定できるにもかかわらず、ネットワーク上の完全に無作為のノードから発信され、かつこのようなノードに送信されるように見える。ハードウェア・ホッピング機能をIPホッピング機能とは異なる通信プロトコル・レベルで（たとえば、性能を向上させるためにカード・ドライバまたはハードウェア・カード自体で）実施できることが理解されよう。

【００８１】

図12Bは、前述の原則を用いて使用することのできる3つの異なる態様またはモードを示している。「プロミスキュアス」モードと呼ばれる第1のモードでは、ネットワーク上のすべてのノードによって共通のハードウェア・アドレス（たとえば、送信元用の固定アドレスおよび着信先用の別の固定アドレス）または完全に無作為のハードウェア・アドレスが使用され、したがって、特定のパケットを1つのノードに帰属させることはできなくなる。各ノードは最初、共通（または無作為）のハードウェア・アドレスを含むすべてのパケットを受け入れ、IPアドレスまたはディスクリミネータ・フィールドを調べて、パケットの着信先がそのノードであるかどうかを判定しなければならない。なお、IPアドレスまたはディスクリミネータ・フィールド、あるいはその両方を上述のアルゴリズムに従って変更することができる。前述のように、この場合、所与のパケットが妥当な送信元ハードウェア・アドレスおよび着信先ハードウェア・アドレスを有するかどうかを判定する追加

の処理が必要になるので、各ノードのオーバヘッドが増大する可能性がある。

【００８２】

「VPN当たりプロミスキュアス」モードと呼ばれる第2のモードでは、少数の1組の固定ハードウェア・アドレスが使用され、仮想専用網上で通信するすべてのノードに固定送信元／着信先ハードウェア・アドレスが使用される。たとえば、イーサネット（登録商標）上に6つのノードがあり、1つのVPN上の各ノードがそのVPN上の他の2つのノードのみと通信できるようにネットワークが2つの専用仮想網に分割される場合、第1のVPN用の1組と第2のVPN用の第2の組の、2組のハードウェア・アドレスを使用することができる。これにより、指定されたVPNから到着するパケットのみを検査すればよいので、妥当なフレームについての検査に必要なオーバヘッドの量が少なくなる。この場合も、VPN内で安全な通信を行えるように、前述のようにIPアドレスおよび1つまたは複数のディスクリミネータ・フィールドをホップすることができる。もちろん、この解決策では、VPNの匿名性は無効になる（すなわち、トラフィックがどのVPNに属するものであるかを部外者が容易に知ることができる。ただし、部外者がそれを特定のマシン／人と相関付けることはできない）。また、ディスクリミネータ・フィールドを使用してある種のDoSアタックを受ける可能性を低減させる必要もある。（たとえば、ディスクリミネータ・フィールドがない場合、LAN上のアタッカーが、VPNによって使用されているMACアドレスを含むフレームのストリームを作成することが可能になる。このようなフレームを拒絶するには過度の処理オーバヘッドが必要になる恐れがあるディスクリミネータ・フィールドは、擬パケットを拒絶する低オーバヘッド手段を実現する。）

【００８３】

「ハードウェア・ホッピング」モードと呼ばれる第3のモードでは、図12Aに示すようにハードウェア・アドレスが変更され、それによって、ハードウェア送信元アドレスおよびハードウェア着信先アドレスが常に変更され、アドレスは帰属不能になる。もちろん、これらの態様の変形態様が可能であり、本発明は、いかなる点においてもこれらの例によって制限されることはない。

【００８４】

B. アドレス空間の拡張

アドレス・ホッピングによってセキュリティおよびプライバシーが確保される。しかし、保護のレベルは、ホップされるブロック内のアドレスの数によって制限される。ホップブロックは、VPNを実現するためにパケットごとに調整されるフィールドを示す。たとえば、各ブロックが4つのアドレス（2ビット）から成るホップブロックを使用するIPアドレス・ホッピングを用いて2つのノードが通信する場合、16個の可能なアドレス・ペア組合せがある。サイズ16のウインドウの場合、大部分の時間において大部分のアドレス・ペアが妥当なペアとして受け入れられる。この制限は、ホップ・アドレス・フィールドに加えてあるいはその代わりにディスクリミネータ・フィールドを使用することによって解消することができる。ディスクリミネータ・フィールドは、アドレス・フィールドとまったく同じようにホップされ、パケットを受信側によって処理すべきかどうかを判定するために使用される。

【００８５】

それぞれが4ビット・ホップブロックを使用する2つのクライアントが、2つのAブロック間のIPホッピングを介して通信するクライアントに与えられるのと同じ保護レベルを望んでいるものと仮定する（ホッピングに有効な24ビット）。20ビットのディスクリミネータ・フィールドを、IPアドレス・フィールドにおけるホッピングに有効な4アドレス・ビットと共に使用すると、この保護レベルが達成される。24ビット・ディスクリミネータ・フィールドは、アドレス・フィールドがホップされず、また無視されない場合に同様な保護レベルを達成する。ディスクリミネータ・フィールドを使用すると、（1）任意に高い保護レベルを達成することができ、（2）アドレス・ホッピングなしで保護が実現されるという2つの利点が与えられる。これは、アドレス・ホッピングによってルーティングの問題が起こる環境で重要である。

【００８６】
C. 同期技法
送信側ノードと受信側ノードがアルゴリズムおよびシード（または準無作為送信元テーブルおよび準無作為着信先テーブルを生成するのに十分な同様な情報）を交換した後、2つのノード間のその後の通信は円滑に進行するものと一般に仮定される。しかし、現実的には、2つのノードは、ネットワークの遅延または障害、あるいはその他の問題のために同期を失う可能性がある。したがって、ネットワーク内の、同期を失ったノード間に同期を再確立する手段を提供することが望ましい。
【００８７】
ある可能な技法では、各ノードに、各パケットが首尾良く受信されたときに確認応答を供給させ、ある期間内に確認応答が受信されなかった場合に、非確認応答パケットを再送する。しかし、この手法は、オーバヘッド・コストを増大させ、たとえば、ストリーミング・ビデオやストリーミング・オーディオなどの高スループット環境では使用不能になる恐れがある。
【００８８】
別の手法では、本明細書で「自己同期」と呼ぶ自動同期技法が使用される。この手法では、各パケットに同期情報が埋め込まれ、それによって、受信側は、送信側との同期を失ったと判定した場合、単一のパケットが受信されたときにそれ自体の同期を取り戻すことができる。（すでに通信が進行中であり、受信側が、まだ送信側と同期していると判定した場合、再同期の必要はない。）受信側は、たとえば、ある期間が経過した時点で満了し、それぞれの妥当なパケットによってリセットされる「デッド・マン」タイマを使用することによって、同期していないことを検出することができる。パケット再試行アタックを防止するために、ハッシングによってパブリック同期フィールド（以下参照）にタイム・スタンプを付加することができる。
【００８９】
一態様では、送信側によって送信される各パケットのヘッダに「同期フィールド」が付加される。この同期フィールドは、平文であっても、あるいはパケットの暗号化された部分の一部であってもよい。送信側および受信側が乱数生成プログラム（RNG）およびシード値を選択しているものと仮定すると、RNGとシードのこの組合せを使用して乱数配列（RNS）を生成することができる。次いで、RNSを使用して、上述のように送信元／着信先IPペア（および必要に応じて、ディスクリミネータ・フィールドならびにハードウェア送信元アドレスおよびハードウェア着信先アドレス）が生成される。しかし、シーケンス全体（または最初のN-1個の値）を生成しなくてもシーケンス内のN番目の乱数を生成することができる。シーケンス・インデックスNが既知である場合、このインデックスに対応する無作為の値を直接生成することができる（以下参照）。それぞれの異なる基本周期を有する様々なRNG（およびシード）を使用して送信元IPシーケンスおよび着信先IPシーケンスを生成することができるが、この場合も基本的な概念が適用される。話を簡単にするために、以下の議論では、単一のRNGシーケンシング機構を使用してIP送信元・着信先アドレス・ペア（のみ）がホップされるものと仮定する。
【００９０】
各パケット・ヘッダ内の同期フィールドは、「自己同期」機能により、IPペアを生成するために使用されているRNSにインデックス（すなわち、シーケンス番号）付けする。RNSを生成するために使用されているRNGにこのようにインデックス付けすると特定の乱数値が生成され、それによって特定のIPペアが生成される。すなわち、RNG、シード、およびインデックス番号から直接IPペアを生成することができ、この方式では、与えられたインデックス番号に関連付けされたシーケンス値に先行する乱数のシーケンス全体を生成することは不要である。
【００９１】
通信者がすでにRNGおよびシードを交換しているものと仮定されているので、IPペアを生成するために与えなければならない新しい情報はシーケンス番号だけである。この番号

が送信側によってパケットヘッダ内に与えられる場合、受信側は、この番号をRNGに入力するだけでIPペアを生成することができ、したがって、パケットのヘッダに示されたIPペアが妥当であることを検証することができる。この方式では、送信側と受信側が同期を失った場合、受信側は、パケット・ヘッダ内のIPペアを、インデックス番号から生成されるIPペアと比較することにより、単一のパケットを受信した時点でただちに同期を取り戻すことができる。したがって、単一のパケットを受信したときに、同期の取れた通信を再開することができ、この方式はマルチキャスト通信にとって理想的な方式になる。極端な場合には、同期テーブルが完全に不要になる。すなわち、送信側と受信側は、同期フィールド内のインデックス番号を使用するだけで各パケット上のIPペアの妥当性を確認することができ、それによってテーブルを完全になくすことができる。

【００９２】

前述の方式は、それに関連する、セキュリティ上のある固有の問題を有する。すなわち、同期フィールドの配置の問題である。このフィールドをアウタ・ヘッダに配置した場合、侵入者はこのフィールドの値および該値とIPストリームとの関係を見ることができる。これにより、場合によっては、IPアドレス・シーケンスを生成するために使用されているアルゴリズムが影響を受け、したがって、通信のセキュリティが影響を受ける。しかし、この値をインナ・ヘッダに配置した場合、送信側はインナ・ヘッダを復号しないかぎり、同期値を抽出してIPペアの妥当性を確認することができなくなる。この場合、受信側は、パケット再生などある種のサービス拒否（DoS）アタックにさらされる。すなわち、受信側がパケットを復号しないかぎりIPペアの妥当性を確認することができない場合、アタッカーが単に、前に妥当であったパケットを再送する場合には、復号に関して著しい量の処理を実行しなければならなくなる恐れがある。

【００９３】

アルゴリズムのセキュリティと処理速度との可能な兼ね合せとして、インナ・ヘッダ（暗号化済み）とアウタ・ヘッダ（未暗号化）との間で同期値が分割される。すなわち、同期値が十分に長い場合、は、平文で表示することのできる急速に変化する部分と、保護されなければならない固定（または非常にゆっくりと変化する）部分とに分割することができる。平文で表示することのできる部分を「パブリック同期」部と呼び、保護されなければならない部分を「プライベート同期」部と呼ぶ。

【００９４】

完全な同期値を生成するにはパブリック同期部とプライベート同期部の両方が必要である。しかし、プライベート部は、固定されるか、あるいはときどきにのみ変化するように選択することができる。したがって、プライベート同期値を受信側によって記憶することができ、したがって、ヘッダを復号しなくても検索することができるようになる。送信側と受信側が、同期のプライベート部分が変化する頻度に関してすでに合意している場合、受信側は、同期を失う原因となった通信ギャップが前のプライベート同期の有効期間を超えた場合に、選択的に単一のヘッダを復号して新しいプライベート同期を抽出することができる。この場合、復号の量が厄介な量になることはなく、したがって、受信側が、単に単一のヘッダをときどき復号する必要があることに基づいてサービス拒否アタックにさらされることはない。

【００９５】

この1つのインプリメンテーションでは、ハッシュ関数を1対1マッピングと共に使用して同期値からプライベート同期部およびパブリック同期部が生成される。このインプリメンテーションは図13に示されており、この場合、（たとえば）第1のISP1302が送信側であり、第2のISP1303が受信側である。（図13では他の代替態様が可能である。）送信されるパケットは、暗号化されていないパブリック・ヘッダまたは「アウタ」ヘッダ1305と、たとえばリンク鍵を使用して暗号化されたプライベート・ヘッダまたは「インナ」ヘッダ1306とを備える。アウタ・ヘッダ1305はパブリック同期部を含み、それに対して、インナ・ヘッダ1306はプライベート同期部を含む。受信側ノードは、復号関数1307を使用してインナ・ヘッダを復号し、プライベート同期部を抽出する。このステップが必要になるのは、

現在バッファされているプライベート同期の有効期間が満了した場合だけである。（現在バッファされているプライベート同期がまだ有効である場合、このプライベート同期は単にメモリから抽出され、ステップ1308に示すようにパブリック同期に「付加」（逆ハッシュでよい）される。）パブリック同期部と復号されたプライベート同期部は関数1308で組み合わされ、組合せ同期1309が生成される。組合せ同期（1309）は次いで、RNG（1310）に送られ、IPアドレス・ペア（1311）と比較され、パケットの妥当性が確認されるか、あるいはパケットが拒絶される。

【００９６】

このアーキテクチャの重要な点は、パブリック同期値が関与する「未来」と「過去」の概念である。スプーフィング・アタックを防止するには同期値自体が無作為の値であるべきであるが、すでに送信された同期値を含むパケットが実際には受信側によって受信されていない場合でも、受信側がこの同期値を高速に識別できることが重要である。1つの解決策として、ハッシングによってタイム・スタンプまたはシーケンス番号がパブリック同期部に付加され、したがって、このパブリック同期部を高速に抽出し、検査し、破棄し、それによってパブリック同期部自体の妥当性を確認することができる。

【００９７】

一態様では、同期フィールドによって生成された送信元／着信先IPペアを、パケット・ヘッダに示されたペアと比較することによってパケットを検査することができる。（1）ペアが一致し、（2）タイム・スタンプが妥当であり、（3）デッドマン・タイマが満了している場合、同期が取り直される。そうでない場合、パケットは拒絶される、十分な処理能力が利用できる場合、デッドマン・タイマおよび同期テーブルを回避することができ、受信側は単にあらゆるパケットに関して同期を取り直す（たとえば、妥当性を確認する）。

【００９８】

前述の方式では、そのインプリメンテーションに影響を与える大整数（たとえば、160ビット）計算が必要になることがある。このような大整数レジスタがない場合、スループットに影響が及び、したがって、場合によってはサービス拒否の点でセキュリティに影響が及ぶ。それにもかかわらず、大整数計算処理機能が普及すると、このような機能を実施するコストが削減される。

【００９９】

D．他の同期方式

上述のように、VPN内の通信側と受信側の間で、連続するW個以上のパケットが失われた場合（Wはウインドウサイズ）、受信側のウインドウは更新されておらず、送信側は、受信側のウインドウに入っていないパケットは送信しない。送信側と受信側は、おそらくウインドウ内の無作為のペアが偶然に繰り返されるまで同期を回復しない。したがって、可能なときにはいつでも送信側と受信側を同期させ、同期が失われたときは常にそれを再確立する必要がある。

【０１００】

同期を失った送信側と受信側の間の同期を、「チェックポイント」方式を使用して回復することができる。この方式では、無作為のIPアドレス・ペアを備えたチェックポイント・メッセージを使用して同期情報が伝達される。一態様では、2つのメッセージを使用して送信側と受信側の間で以下の同期情報が伝達される。

1．SYNC_REQは、送信側が同期を取る必要があることを示すために送信側によって使用されるメッセージであり、

2．SYNC_ACKは、受信側の同期が取れたことを送信側に知らせるために受信側によって使用されるメッセージである。

この手法の一変形態様によれば、送信側と受信側は共に以下の3つのチェックポイントを維持する（図14参照）。

1．送信側において、ckpt_o（「チェックポイント・オールド」）は、最後のSYNC_REQパケットを受信側に再送するために使用されたIPペアである。受信側において、ckpt_o（

「チェックポイント・オールド」）は、送信側から繰り返しSYNC_REQパケットを受信するIPペアである。

2．送信側において、ckpt_n（「チェックポイント・ニュー」）は、次のSYNC_REQパケットを受信側に送信するために使用されるIPペアである。受信側において、ckpt_n（「チェックポイント・ニュー」）は、新しいSYNC_REQパケットを送信側から受信し、受信側のウインドウを再整列させ、ckpt_oをckpt_nに設定させ、新しいckpt_nを生成させ、新しいckpt_rを生成させるIPペアである。

3．送信側において、ckpt_rは、次のSYNC_ACKパケットを受信側に送信するために使用されるIPペアである。受信側において、ckpt_rは、新しいSYNC_ACKパケットを送信側から受信し、新しいckpt_nを生成させるIPペアである。SYNC_ACKは受信側ISPから送信側ISPに送信されるので、送信側ckpt_rは受信側のckpt_rをを指し、受信側ckpt_rは送信側のckpt_rを指す（図14参照）。

送信側が同期を開始すると、送信側が次のデータ・パケットを送信するために用いるIPペアが所定の値に設定され、受信側がまずSYNC_REQを受信すると、受信側ウインドウが、送信側の次のIPペアが中心になるように更新される。

【０１０１】

同期はパケット・カウンタによって開始することも（たとえば、N個のパケットが送信されるたびに同期を開始する）、あるいはタイマによって開始することも（S秒おきに同期を開始する）、あるいはそれらの組合せによって開始することもできる。図15を参照されたい。送信側から見ると、この技法は以下のように作用する。（1）各送信側は、受信側が同期していることを確認するために定期的に「同期要求」メッセージを受信側に送信する。（2）受信側は、まだ同期している場合、「同期確認」メッセージを送り返す。（これがうまくいった場合、さらなる処置は必要とされない）。（3）ある期間内に「同期確認」が受信されなかった場合、送信側は同期要求を再び送信する。送信側が「同期確認」応答を受信せずに次のチェックポイントに到達した場合、同期が失われ、送信側は送信を停止する必要がある。送信側はsync_ackを受信するまでsync_reqsを送信し続け、受信した時点で送信が再確立される。

【０１０２】

受信側から見ると、この方式は以下のように作用する。（1）受信側は、送信側から「同期要求」要求を受信すると、ウインドウを次のチェックポイント位置へ進め（場合によっては必要に応じてペアをスキップする）、「同期応答」メッセージを送信側に送信する。同期が失われていない場合、「ジャンプ・アヘッド」によって、テーブル内の次の利用可能なアドレス・ペアに進む（すなわち、通常の前進）。

【０１０３】

侵入者が「同期要求」メッセージを捕捉し、新しい「同期要求」メッセージを送信することによって通信の干渉を試みた場合、同期が確立されているか、あるいはこのメッセージが実際に同期を再確立する助けになる場合、このメッセージは無視される。

【０１０４】

ウインドウは、再同期が行われたときは常に再整列させられる。この再整列に伴い、SYNC_REQパケットが送信された直後に送信されたパケットによって使用されたアドレス・ペアにまたがるように受信側のウインドウが更新される。通常、送信側と受信側は互いに同期させられる。しかし、ネットワーク・イベントが起こった場合、再同期中に受信側のウインドウを多数のステップ分進めなければならないことがある。この場合、介在する乱数間を順次進む必要なしにウインドウを進めることが望ましい。（この機能は、上述の自動同期手法にも望ましい）。

【０１０５】

E．ジャンプアヘッド機能を有する乱数生成プログラム

無作為にホップされるアドレスを生成するための魅力的な方法は、送信側と受信側で同一の乱数生成プログラムを使用し、パケットが送信され受信されたときにこのプログラムを進める方法である。使用できる多数の乱数生成アルゴリズムがある。各アルゴリズムは

、アドレス・ホッピング応用例に対する利点と欠点を有する。

【０１０６】

線形乱数生成プログラム（LCR）は、明確な特徴を有する高速で簡単な乱数生成プログラムであり、効率的にnステップ先にジャンプさせることができる。LCRは、以下の反復を使用してシードX_0から始まる乱数X_1、X_2、X_3、・・・、X_Kを生成する。

$$X_i=(aX_{i-1}+b)\bmod c \quad (1)$$

上式で、a、b、およびcは特定のLCRを定義する符号である。Xiに関する別の数式、すなわち、

$$X_i=((a^i(X_0+b)-b)/(a-1))\bmod c \quad (2)$$

によって、ジャンプアヘッド機能が有効になる。係数a_iは、拘束されない場合、iが小さい場合もで非常に大きなることができる。したがって、モジュロ演算のいくつかの特殊な特性を使用して、（2）を計算するのに必要なサイズおよび処理時間を調節することができる。（2）は次式のように書くことができる。

$$X_i=(a^i(X_0(a-1)+b)-b/(a-1)\bmod c \quad (3)$$

以下のことを示すことができる。

$$(a^i(X_0(a-1)+b)-b)/(a-1)\bmod c=$$
$$((a^i\bmod((a-1)c)(X_0(a-1)+b)-b)/(a-1))\bmod c \quad (4)$$

$(X_0(a-1)+b)$を$(X_0(a-1)+b)\bmod c$として記憶し、bをb mod cとして記憶し、$a^i\bmod((a-1)c)$を計算することができる（これには0(log(i)）回のステップが必要である）。

【０１０７】

このアルゴリズムの実際的なインプリメンテーションは、各同期間で一定距離nだけジャンプする。これは、nパケットおきの同期に相当する。ウインドウは、前のウインドウが開始してからn IPペア後に開始する。ノードは、X_j^w、すなわち、J番目のチェックポイントでの乱数をX_0として使用し、nをiとして使用して、LCR当たり1度$a^n\bmod((a-1)c)$を記憶し、

$$X_{j+1}^w=X_{n(j+1)}=((a^n\bmod((a-1)c)(X_j^w(a-1)+b)-b)/(a-1))\bmod c \quad (5)$$

を、j＋1番目の同期用の乱数を生成するように設定することができる。ノードは、この構成を使用して、（nとは無関係の）一定の時間内に、各同期間で任意の（しかし、一定の）距離だけ先にジャンプすることができる。

【０１０８】

したがって、一般に擬似乱数生成プログラム、特にLCRはそのサイクルを繰り返す。この繰返しは、IPホッピング方式の弱点となる可能性がある。すなわち、繰侵入者は、繰返しを待つだけで未来のシーケンスを予想することができる。この弱点に対処する1つの方法は、既知の長いサイクルを有する乱数生成プログラムを作成することである。無作為のシーケンスが繰り返される前に、このシーケンスを新しい乱数生成プログラムで置き換えることができる。既知の長いサイクルを有するLCRを構成することができる。このことは、現在の所、多くの乱数生成プログラムには当てはまらない。

【０１０９】

乱数生成プログラムは、暗号に関して安全でない点がある。侵入者は、出力またはその一部を調べることによってRNGパラメータを導くことができる。このことはLCGに当てはまる。この弱点は、出力を乱数生成プログラムの一部とするように構成された暗号化プログラムを組み込むことによって軽減することができる。乱数生成プログラムは、侵入者が暗号プログラムにアタック、たとえば既知の平文アタックを開始するのを防止する。

【０１１０】

F．乱数生成プログラムの例

a=31、b=4、およびc=15であるRNGについて考える。この場合、数式（1）は

$$X_i=(31X_{i-1}+4)\bmod 15 \quad (6)$$

になる。

【０１１１】

X_0=1を設定した場合、数式（6）はシーケンス1、5、9、13、2、6、10、14、3、7、11、

0、4、8、12を生成する。このシーケンスは無限に繰り返される。このシーケンスで3つの数だけ先にジャンプする場合、$a^n=31^3=29791$、$c^*(a-1)=15*30=450$、および$a^n \bmod((a-1)c)=31^3 \bmod(15*30)=29791 \bmod(450)=91$である。数式（5）は

$$((91(X_i 30+4)-4)/30) \bmod 15 \quad (7)$$

表1は、（7）のジャンプアヘッド計算を示している。この計算は、5から始まり3つ先にジャンプする。

【表１】

I	X_i	$(X_i 30+4)$	$91(X_i 30+4)-4$	$((91(X_i 30+4)-4)/30$	X_{i+3}
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

【０１１２】

G．高速パケット・フィルタ

アドレス・ホッピングVPNは、パケットが妥当なヘッドを有し、したがって、さらなる処理を必要とするか、それとも不当なヘッダを有し（有害なパケット）、ただちに拒絶すべきであるかどうかを高速に判定しなければならない。このような高速の判定を「高速パケット・フィルタリング」と呼ぶ。この機能は、受信側のプロセッサを飽和させるために受信側で有害なパケットのストリームを高速に作成する侵入者によるアタック（いわゆる「サービス拒否」アタック）からVPNを保護する。高速パケット・フィルタリングは、イーサネット（登録商標）などの共用媒体上でVPNを実施するための重要な機能である。

【０１１３】

VPNのすべての参加者が、割り当てられていない「A」アドレス・ブロックを共用すると仮定した場合、1つの可能性として、共用媒体上でアドレス・ホッピングされないマシンに割り当てられることのない実験的な「A」ブロックが使用される。「A」ブロックは、「C」ブロック内の8ビットとは逆にホップできる24ビットのアドレスを有する。この場合、ホップブロックは「A」ブロックになる。イーサネット（登録商標）上では以下の理由で、実験的な「A」ブロックが使用される可能性が高い。

1．アドレスが、イーサネット（登録商標）の外部で無効であり、ゲートウェイによって妥当な外部の着信先にルーティングされることがない。

2．各「A」ブロック内でホップできる2^{24}（～1600万）個のアドレスがある。このため、＞280兆個の可能なアドレス・ペアが生成され、侵入者が妥当なアドレスを推測できる可能性は非常に低くなる。また、別々のVPN同士が衝突する可能性が許容される程度に低くなる（共用される媒体上のすべてのVPNが独立に、「A」ブロックから無作為のアドレス・ペアを生成する。）

3．（マシンがプロミスキュアス・モードでないかぎり）パケットが、イーサネット（登録商標）上のユーザであり、かつVPN上には存在しないユーザに受信されることがなく、非VPNコンピュータに対する影響が最小限に抑えられる。

【０１１４】

このイーサネット（登録商標）の例を、高速パケット・フィルタリングの1インプリメンテーションを説明するために使用する。理想的なアルゴリズムは、パケット・ヘッダを高速に調べ、パケットが有害であるかどうかを判定し、あらゆる有害なパケットを拒絶するか、あるいはパケット・ヘッダが一致するのはどのアクティブIPペアかを判定する。この場合の問題は、従来のアソシエーティブ・メモリの問題である。この問題を解決するために様々な技法が開発されている（ハッシング、Bツリーなど）。これらの手法はそれぞ

れ、利点と欠点を有する。たとえば、ハッシュ・テーブルは、統計的な意味で極めて高速に動作させることができるが、場合によってはずっと低速のアルゴリズムに退化することがある。この低速はある期間にわたって持続することがある。有害なパケットは常に高速に破棄する必要があるので、ハッシングは受け入れられない。

【０１１５】

H. 存在ベクトル・アルゴリズム

存在ベクトルとは、nビット番号（それぞれ0から2^n-1までの範囲）によってインデックス付けすることのできる長さ2nのビット・ベクトルである。各番号によってインデックス付けされた存在ベクトル内のビットを1に設定することにより、k個のnビット番号（必ずしも一意ではない）の存在を示すことができる。nビット番号xがk個の番号のうちの1つであるのは、存在ベクトルのx番目のビットが1である場合だけである。存在ベクトルにインデックス付けし、1を探すことによって高速パケット・フィルタを実施することができる。この方法を「テスト」と呼ぶ。

【０１１６】

たとえば、存在ベクトルを使用して番号135を表す必要があるものと仮定する。ベクトルの135番目のビットが設定される。したがって、1つのビット、すなわち135番目のビットを検査することによって、アドレス135が妥当であるかどうかを迅速に判定することができる。存在ベクトルは、IPアドレスのテーブル・エントリに対応するように事前に作成することができる。実際には、着信アドレスを長いベクトルのインデックスとして使用して、比較を非常に高速に行うことができる。各RNGが新しいアドレスを生成すると、存在ベクトルはこの情報を反映するように更新される。ウインドウが移動すると、存在ベクトルは、もはや妥当ではないアドレスをゼロにするように更新される。

【０１１７】

テストの効率と、存在ベクトルを記憶するのに必要なメモリの量との兼ね合せがある。たとえば、48ビットのホッピング・アドレスをインデックスとして使用する必要がある場合、存在ベクトルは35テラバイトを有する必要がある。これが実際上大き過ぎることは明らかである。この代わりに、48ビットをいくつかのより小さなフィールドに分割することができる。たとえば、48ビットを4つの12ビット・フィールドに細分することができる。これによって、記憶要件は2048バイトに削減され、その代わりに、ときどき有害なパケットを処理しなければならなくなる。実際には、1つの長い存在ベクトルではなく、分解された各アドレス部分が、4つの短い存在ベクトルのすべてに一致しないかぎり、さらなる処理が許可されなくなる。（アドレス部分の第1の部分が第1の存在ベクトルに一致しない場合、残りの3つの存在ベクトルを検査する必要はない）。

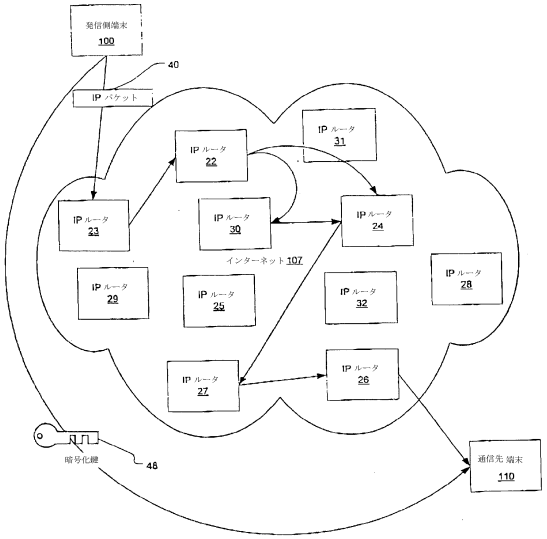
【０１１８】

存在ベクトルのy番目のビットが1であるのは、対応するフィールドがyである1つまたは複数のアドレスがアクティブである場合だけである。アドレスがアクティブであるのは、そのアドレスの適切なサブフィールドによってインデックス付けされた各存在ベクトルが1である場合だけである。

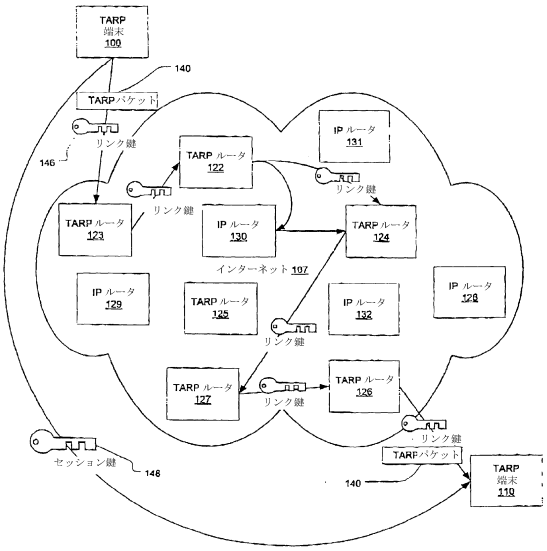
【０１１９】

32個のアクティブ・アドレスおよび3個のチェックポイントから成るウインドウについて考える。有害なパケットは、1つの存在ベクトルに99％よりも長い時間にわたってインデックス付けすることによって拒絶される。有害なパケットは、4つの存在ベクトルのすべてに99.9999995％よりも長い時間にわたってインデックス付けすることによって拒絶される。平均すると、有害なパケットは1.02回未満の存在ベクトル・インデックス動作で拒絶される。高速パケット・フィルタを通過した少数の有害パケットは、一致するペアが、アクティブウインドウ内に見つからないか、あるいはアクティブ・チェックポイントであるときに拒絶される。思いがけずヘッダに一致した有害なパケットは、VPNソフトウェアがこのヘッダの復号を試みたときに拒絶される。しかし、これらのケースは極めてまれである。空間と速度の兼合いを図るようにこの方法を構成する手段として、他に多くの方法がある。

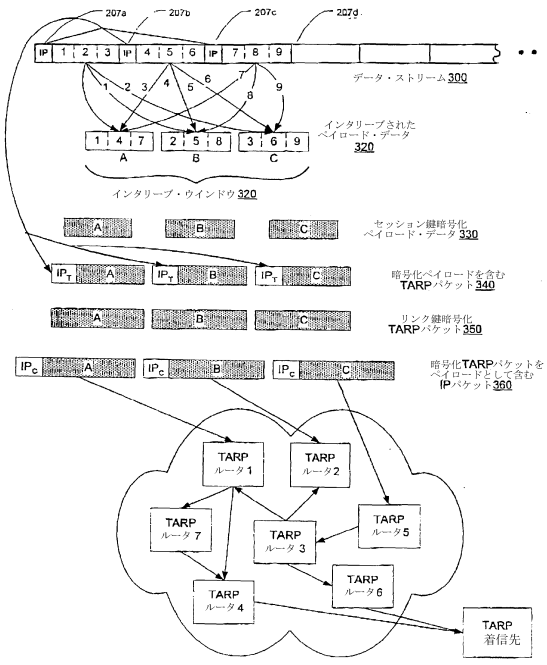
【図１】



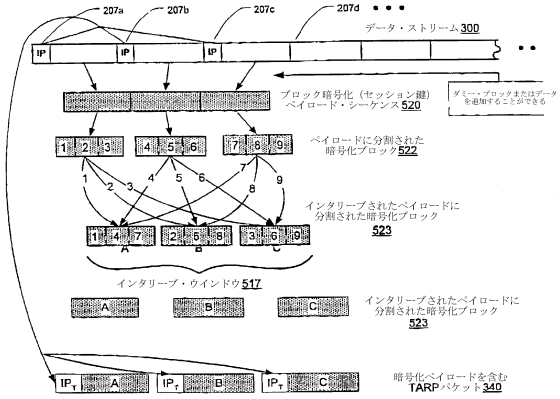
【図２】



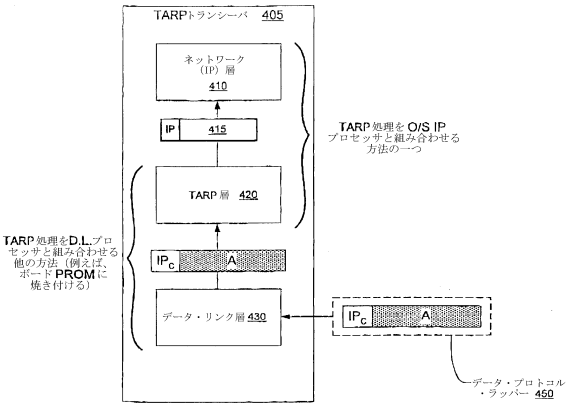
【図３Ａ】



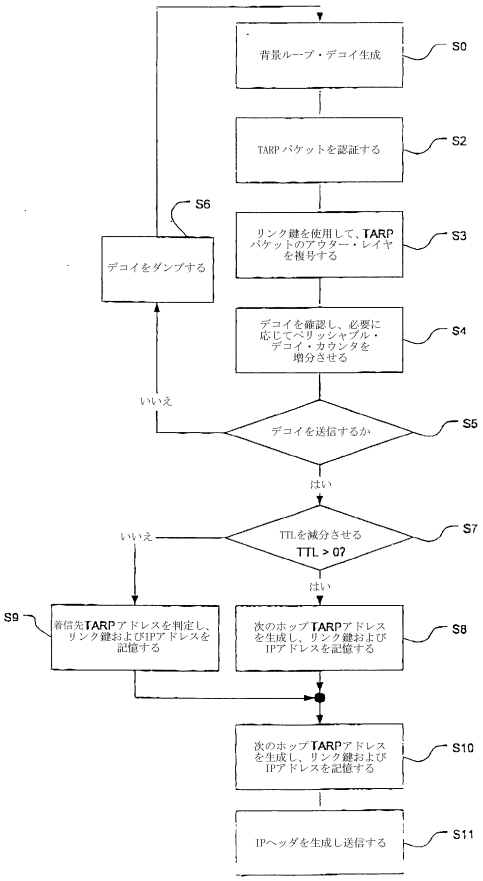
【図３Ｂ】



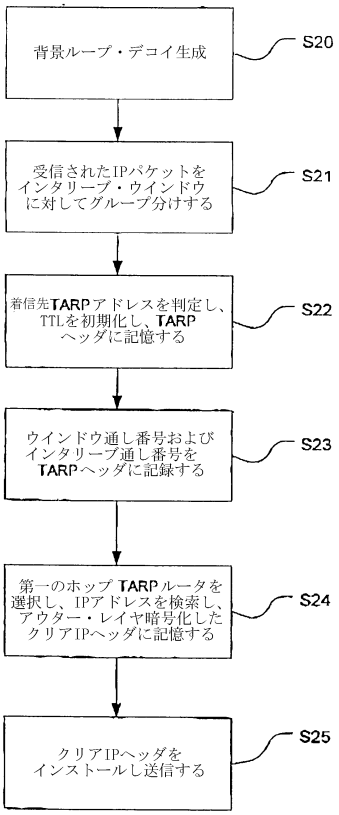
【図４】



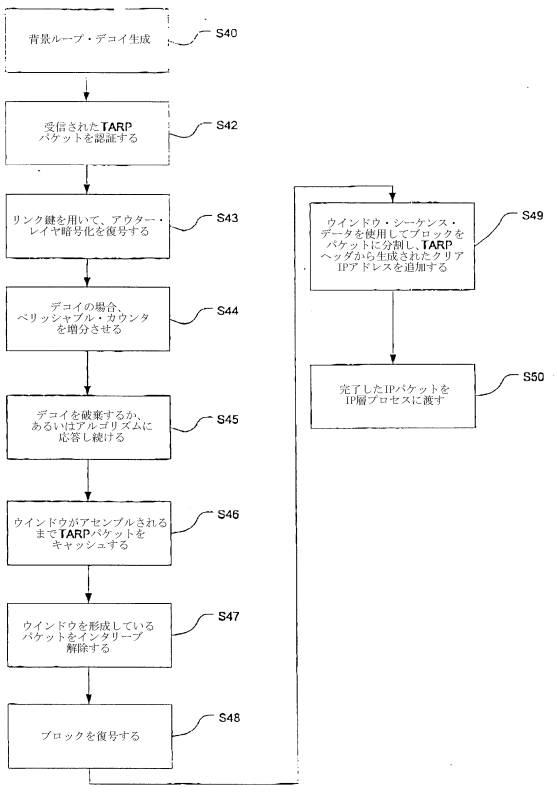
【図５】


背景ループ・デコイ生成
S0
TARPパケットを認証する
S2
リンク鍵を使用して、TARPパケットのアウター・レイヤを復号する
S3
デコイをダンプする
S6
デコイを確認し、必要に応じてペリッシャブル・デコイ・カウンタを増分させる
S4
いいえ
デコイを送信するか
S5
はい
いいえ
TTLを減分させる TTL > 0?
S7
はい
S9
着信先TARPアドレスを判定し、リンク鍵およびIPアドレスを記憶する
次のホップTARPアドレスを生成し、リンク鍵およびIPアドレスを記憶する
S8
次のホップTARPアドレスを生成し、リンク鍵およびIPアドレスを記憶する
S10
IPヘッダを生成し送信する
S11

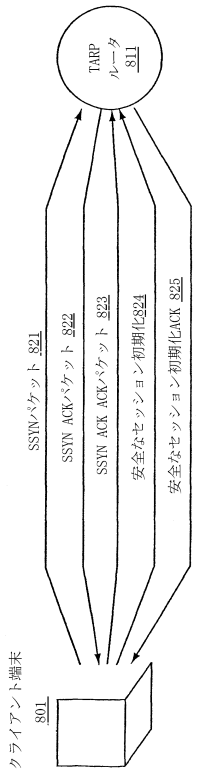
【図６】


背景ループ・デコイ生成
S20
受信されたIPパケットをインタリーブ・ウインドウに対してグループ分けする
S21
着信先TARPアドレスを判定し、TTLを初期化し、TARPヘッダに記憶する
S22
ウインドウ通し番号およびインタリーブ通し番号をTARPヘッダに記録する
S23
第一のホップTARPルータを選択し、IPアドレスを検索し、アウター・レイヤ暗号化したクリアIPヘッダに記憶する
S24
クリアIPヘッダをインストールし送信する
S25

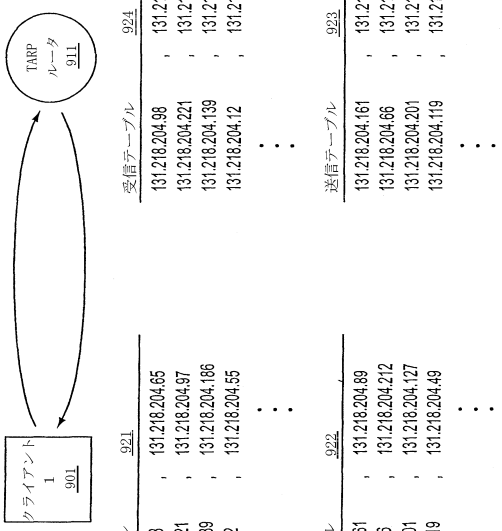
【図７】


背景ループ・デコイ生成
S40
受信されたTARPパケットを認証する
S42
リンク鍵を用いて、アウター・レイヤ暗号化を復号する
S43
デコイの場合、ペリッシャブル・カウンタを増分させる
S44
デコイを破棄するか、あるいはアルゴリズムに応答し続ける
S45
ウインドウがアセンブルされるまでTARPパケットをキャッシュする
S46
ウインドウを形成しているパケットをインタリーブ解除する
S47
ブロックを復号する
S48
ウインドウ・シーケンス・データを使用してブロックをパケットに分割し、TARPヘッダから生成されたクリアIPアドレスを追加する
S49
完了したIPパケットをIP層プロセスに渡す
S50

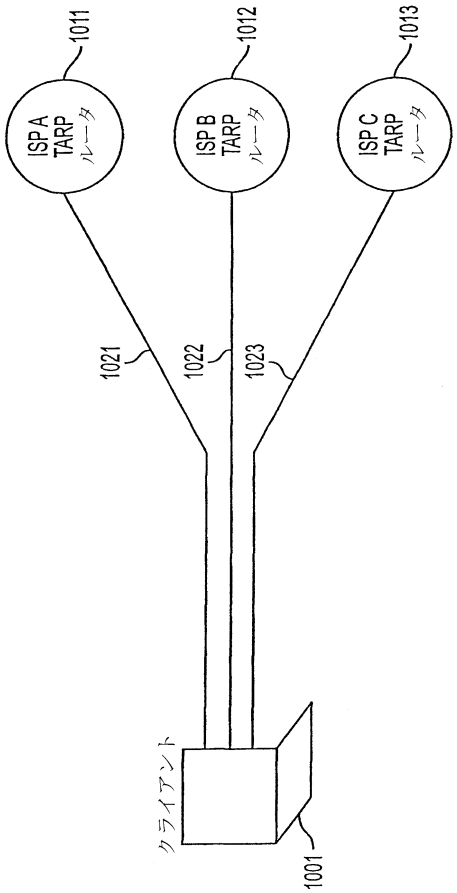
【図８】


TARPルータ 811
SSYNパケット 821
SSYN ACKパケット 822
SSYN ACK ACKパケット 823
安全なセッション初期化824
安全なセッション初期化ACK 825
クライアント端末 801

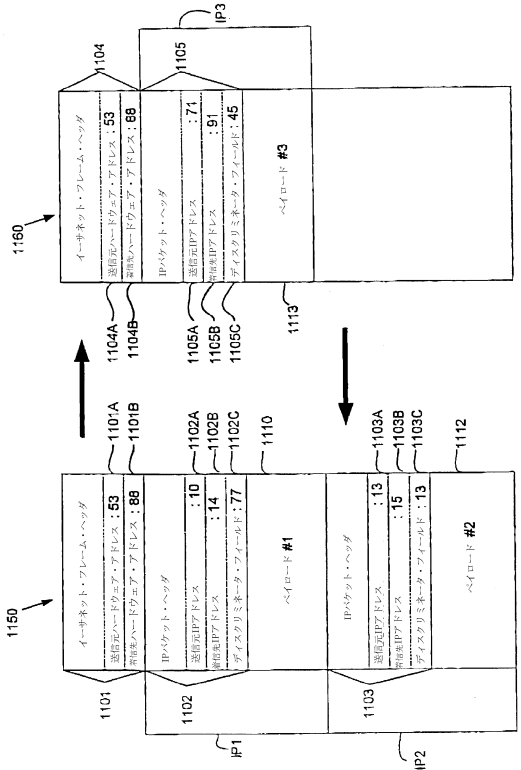
【図９】



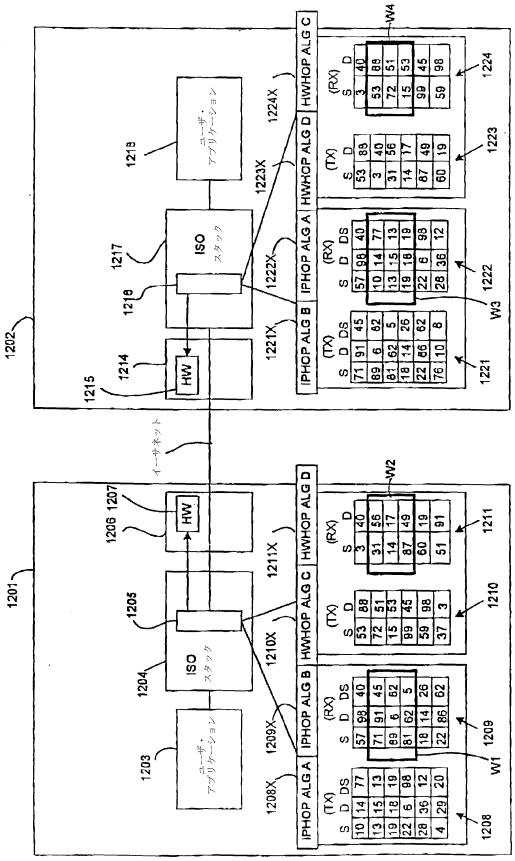
【図１０】



【図１１】



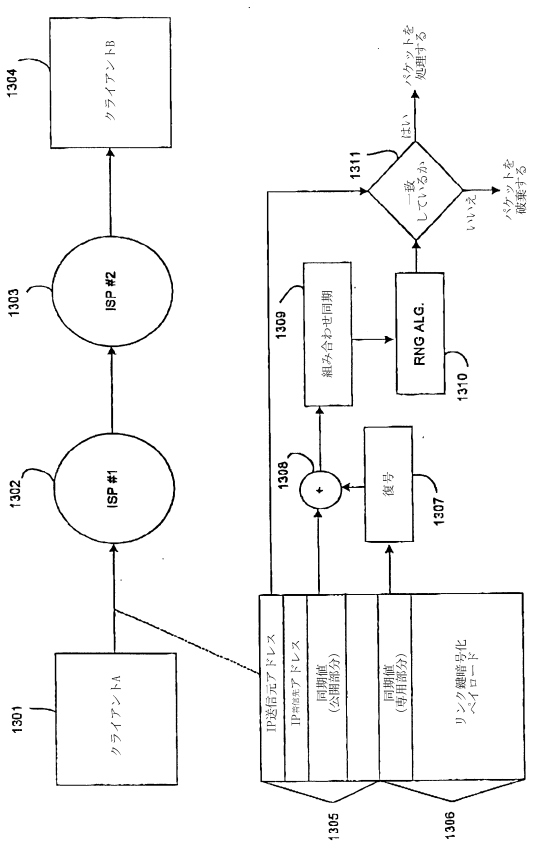
【図１２Ａ】



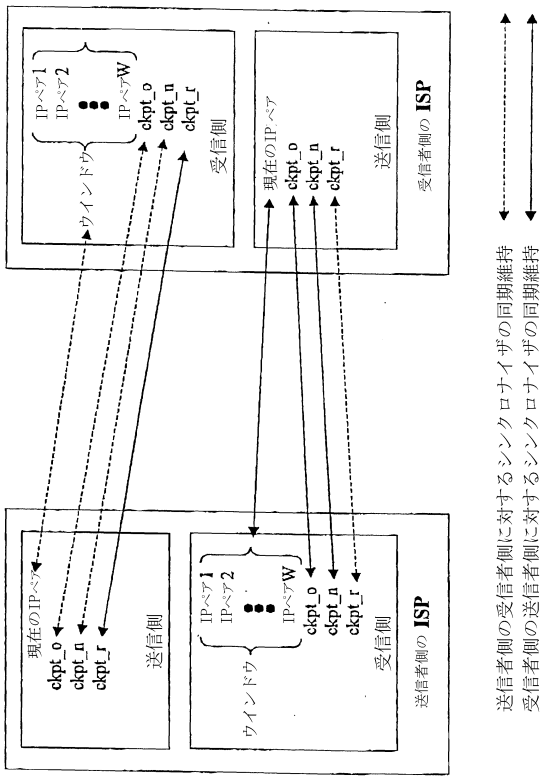
【図１２Ｂ】

モードまたは態様	ハードウェア・アドレス	IPアドレス	ディスクリミネータ・フィールド値
1. 不規則	すべてのノードについて同じ、あるいは完全に無作為	同期的に変更可能	同期的に変更可能
2. VPNごとに不規則	各VPNごとに固定	同期的に変更可能	同期的に変更可能
3. ハードウェア・ホッピング	同期的に変更可能	同期的に変更可能	同期的に変更可能

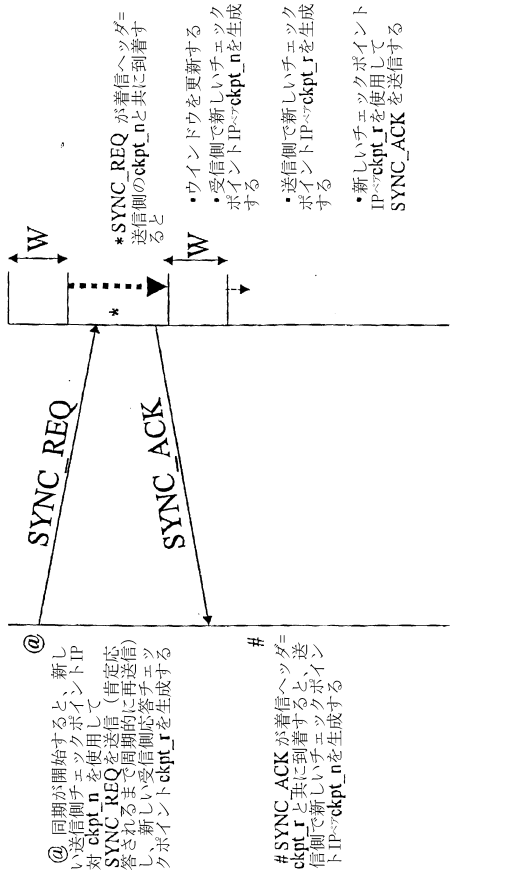
【図１３】



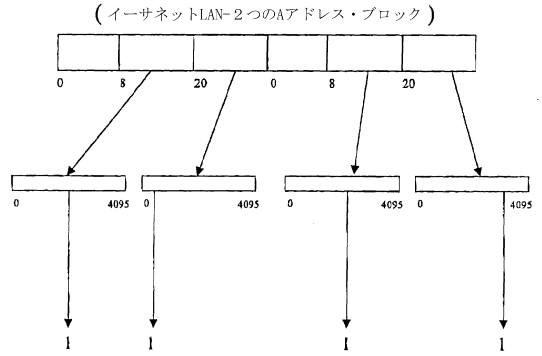
【図１４】



【図１５】



【図１６】


(イーサネットLAN-２つのAアドレス・ブロック)
0
8
20
0
8
20
0
4095
0
4095
0
4095
0
4095
1
1
1
1

フロントページの続き

(74)代理人 100148699
弁理士 佐藤 利光
(74)代理人 100128048
弁理士 新見 浩一
(74)代理人 100129506
弁理士 小林 智彦
(74)代理人 100130845
弁理士 渡邉 伸一
(74)代理人 100114340
弁理士 大関 雅人
(74)代理人 100114889
弁理士 五十嵐 義弘
(74)代理人 100121072
弁理士 川本 和弥
(72)発明者 マンガー エドモンド シー.
アメリカ合衆国 メリーランド州 クロウンズビル オパカ コート １１０１
(72)発明者 サビオ ビンセント ジェイ.
アメリカ合衆国 メリーランド州 コロンビア セッティング サン ウェイ ７４８９
(72)発明者 ショート ロバート ダンハム ザ・サード
アメリカ合衆国 バージニア州 リースバーグ グース クリーク レーン ３８７１０
(72)発明者 グリゴール バージル ディー.
アメリカ合衆国 メリーランド州 シェビー チェイス ブルックサイド ドライブ ６００９
(72)発明者 シャミド ダグラス チャールズ
アメリカ合衆国 メリーランド州 セベルナ パーク オーク コート ２３０

審査官 衣鳩 文彦

(56)参考文献 特開平６－１５２６５５（ＪＰ，Ａ）
特開平９－３４８１６（ＪＰ，Ａ）
特開平１０－２４３０２１（ＪＰ，Ａ）
国際公開第９８／４０９９０（WO，Ａ1）
特開平９－２１４５５６（ＪＰ，Ａ）
特開平９－１８４７３（ＪＰ，Ａ）
特開平７－２９７８１７（ＪＰ，Ａ）
特開平４－３６３９４１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ １２／００～１２／９５５