

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4464963号
(P4464963)

(45) 発行日 平成22年5月19日(2010.5.19)

(24) 登録日 平成22年2月26日(2010.2.26)

(51) Int.Cl.		F I	
H O 4 L 12/56 (2006.01)		H O 4 L 12/56	B
G O 6 F 21/20 (2006.01)		G O 6 F 15/00	3 3 O A

請求項の数 39 (全 49 頁)

(21) 出願番号	特願2006-517182 (P2006-517182)	(73) 特許権者	392026693
(86) (22) 出願日	平成16年6月4日(2004.6.4)		株式会社エヌ・ティ・ティ・ドコモ
(65) 公表番号	特表2007-521749 (P2007-521749A)		東京都千代田区永田町二丁目11番1号
(43) 公表日	平成19年8月2日(2007.8.2)	(74) 代理人	100083806
(86) 国際出願番号	PCT/US2004/017790		弁理士 三好 秀和
(87) 国際公開番号	W02005/006663	(74) 代理人	100095500
(87) 国際公開日	平成17年1月20日(2005.1.20)		弁理士 伊藤 正和
審査請求日	平成19年6月4日(2007.6.4)	(74) 代理人	100101247
(31) 優先権主張番号	60/482,350		弁理士 高橋 俊一
(32) 優先日	平成15年6月24日(2003.6.24)	(74) 代理人	100117064
(33) 優先権主張国	米国 (US)		弁理士 伊藤 市太郎
(31) 優先権主張番号	60/500,175	(72) 発明者	トロッスル、 ジョナサン
(32) 優先日	平成15年9月2日(2003.9.2)		アメリカ合衆国、 カリフォルニア州 9
(33) 優先権主張国	米国 (US)		5110, サンノゼ, スイート300
			, メトロ ドライブ 181
			最終頁に続く

(54) 【発明の名称】 暗号で保護されたプレフィックスを用いたインターネットプロトコルネットワーク用のロケーションプライバシー

(57) 【特許請求の範囲】

【請求項1】

複数の相互接続されたルータを備えるネットワークであって、
データは、前記相互接続されたルータを介して前記ネットワーク上で送信され、
前記データは、I Pアドレスを含み、
前記 I Pアドレスは、前記データの宛先であるルータの論理的な位置を少なくとも識別し、

前記 I Pアドレスは、複数の部分に分割され、

少なくとも前記複数の部分の部分集合は、暗号化され、

前記相互接続されたルータの部分集合は、少なくとも前記 I Pアドレスの一部を解読でき、

前記相互接続されたルータの前記部分集合は、少なくとも前記 I Pアドレスの一部を解読するための鍵を受け取り、

前記相互接続されたルータの前記部分集合は、同じレベルのルータを含み、

前記鍵は、前記同じレベルの前記ルータによって共有されることを特徴とするネットワーク。

【請求項2】

少なくとも前記 I Pアドレスの第1の部分は、前記 I Pアドレスの他の部分とは別に暗号化されることを特徴とする請求項1に記載のネットワーク。

【請求項3】

10

20

前記相互接続されたルータの部分集合は、少なくとも、前記第 1 の部分とは異なる前記 I P アドレスの部分を読解できないことを特徴とする請求項 2 に記載のネットワーク。

【請求項 4】

前記 I P アドレスは、ルーティングドメインツリー内のルータの統合によって、少なくとも部分的に分割され、

前記ルーティングドメインツリーは、前記複数の相互接続されたルータ間の相互接続を表し、

前記統合は、共通の親ルータを共有する複数のルータに対して行われることを特徴とする請求項 3 に記載のネットワーク。

【請求項 5】

第 1 の期間中に、前記相互接続されたルータは、少なくとも前記 I P アドレスの一部を読解することができ、

前記相互接続されたルータは、前記第 1 の期間中に、少なくとも前記 I P アドレスの一部を読解することができ、或いは、第 2 の期間中に、前記 I P アドレスの他の部分を読解することができることを特徴とする請求項 1 に記載のネットワーク。

【請求項 6】

前記 I P アドレスの前記第 1 の部分は、鍵を用いて読解され得ることを特徴とする請求項 3 に記載のネットワーク。

【請求項 7】

前記相互接続されたルータの前記部分集合は、共通リンクを共有するルータを含み、
前記鍵は、前記共通リンクを共有する前記ルータによって共有されることを特徴とする請求項 1 に記載のネットワーク。

【請求項 8】

前記相互接続されたルータの前記部分集合は、少なくともクロスオーバールータを備えることを特徴とする請求項 1 に記載のネットワーク。

【請求項 9】

前記鍵は、前記クロスオーバールータと同じレベルのルータによって共用されることを特徴とする請求項 8 に記載のネットワーク。

【請求項 10】

前記相互接続されたルータの前記部分集合は、論理ツリー内に構成されていることを特徴とする請求項 1 に記載のネットワーク。

【請求項 11】

ネットワークであって、

複数の相互接続されたルータと、

前記ネットワークを介して、I P アドレスを含むデータを送信する手段と、

前記 I P アドレスを複数の部分に分割する分割手段と、

少なくとも前記複数の部分の部分集合を暗号化する暗号化手段とを備え、

前記相互接続されたルータの部分集合は、少なくとも前記 I P アドレスの一部を読解でき、

前記相互接続されたルータの前記部分集合は、少なくとも前記 I P アドレスの一部を読解するための鍵を受け取り、

前記相互接続されたルータの前記部分集合は、同じレベルのルータを含み、

前記鍵は、前記同じレベルの前記ルータによって共有されることを特徴とするネットワーク。

【請求項 12】

複数の相互接続されたルータを含むネットワーク上でデータを共有する方法であって、

前記ネットワーク上で、I P アドレスを含むデータを送信する工程と、

前記 I P アドレスにおいて、少なくとも、前記データの宛先であるルータの論理的な位置を識別する工程と、

10

20

30

40

50

前記 I P アドレスを、複数の部分に分割する工程と、
少なくとも前記複数の部分の部分集合を暗号化する工程と、
前記相互接続されたルータの部分集合に対して、少なくとも前記 I P アドレスの一部を解読するための鍵を与える工程と、
前記相互接続されたルータの前記部分集合を用いて、少なくとも前記 I P アドレスの一部を解読する工程と、
同じレベルのルータとの間で前記鍵を共有する工程とを含み、
前記レベルは、前記相互接続されたルータの前記部分集合で構成されていることを特徴とする記載の方法。

【請求項 1 3】

10

少なくとも前記 I P アドレスの第 1 の部分を、他の部分とは別に暗号化する工程を含むことを特徴とする請求項 1 2 に記載の方法。

【請求項 1 4】

少なくとも、前記第 1 の部分とは異なる前記 I P アドレスの部分の解読できない前記相互接続されたルータの前記部分集合を含むことを特徴とする請求項 1 3 に記載の方法。

【請求項 1 5】

ルーティングドメインツリー内のルータの統合によって、少なくとも部分的に前記 I P アドレスを分割する工程を含み、
前記ルーティングドメインツリーは、前記複数の相互接続されたルータ間の相互接続を表し、前記統合は、共通の親ルータを共有する複数のルータに対して行われることを特徴とする請求項 1 2 に記載の方法。

20

【請求項 1 6】

第 1 の期間中に、少なくとも前記 I P アドレスの一部を解読する工程と、
前記第 1 の期間中に、第 2 の期間のために、少なくとも前記 I P アドレスの一部を解読する鍵を受け取る工程とを含むことを特徴とする請求項 1 2 に記載の方法。

【請求項 1 7】

鍵を用いて、前記 I P アドレスの前記第 1 の部分を解読する工程を含むことを特徴とする請求項 1 4 に記載の方法。

【請求項 1 8】

共通リンクを共有するルータとの間で前記鍵を共有する工程を含み、
前記相互接続されたルータの前記部分集合は、共通リンクを共有するルータを含むことを特徴とする請求項 1 2 に記載の方法。

30

【請求項 1 9】

前記相互接続されたルータの前記部分集合は、少なくともクロスオーバールータを備えることを特徴とする請求項 1 2 に記載の方法。

【請求項 2 0】

前記クロスオーバールータと同じレベルのルータと、前記鍵を共用する工程を有することを特徴とする請求項 1 9 に記載の方法。

【請求項 2 1】

前記相互接続されたルータの部分集合は、同じツリーのルータを備えることを特徴とする請求項 1 2 に記載の方法。

40

【請求項 2 2】

前記 I P アドレスに対するハッシュ関数を演算する工程を含み、
前記 I P アドレスに対する前記ハッシュ関数の結果として、期待値が生成され、
前記期待値は、所望のパス上の前記 I P アドレスの宛先を表すことを特徴とする請求項 1 2 に記載の方法。

【請求項 2 3】

前記期待値を、前記 I P アドレスのサフィックスに対応させる工程を有することを特徴とする請求項 2 2 に記載の方法。

【請求項 2 4】

50

前記期待値は、前記 I P アドレスに含まれていることを特徴とする請求項 2 2 に記載の方法。

【請求項 2 5】

前記 I P アドレスに含まれている修飾子から前記期待値を指し示す工程を含み、
前記期待値は、前記 I P アドレスとは別個のテーブル内にあることを特徴とする請求項 2 2 に記載の方法。

【請求項 2 6】

プロセッサとデータパスとを備えるルータであって、
データは、前記ルータに対して、前記ルータを介して、及び/又は、前記ルータから送信され、前記データは、前記プロセッサによって処理が施されて、前記データパスに沿って送信され、

前記データは、前記プロセッサによって、I P アドレスについての処理を含む処理が施され、前記 I P アドレスは、前記データの宛先である相互接続されたルータ群内のルータの少なくとも論理的な位置を識別し、

前記処理は、前記 I P アドレスを複数の部分に分割することを含み、少なくとも前記複数の部分の部分集合は暗号化され、

前記ルータは、少なくとも前記 I P アドレスの一部を解読することができ、

前記ルータには、少なくとも前記 I P アドレスの一部を解読するための鍵が与えられ

、
前記ルータは、同じレベルのルータを含む前記相互接続されたルータの部分集合の構成員であり、

前記鍵は、前記同じレベルのルータによって共用されることを特徴とするルータ。

【請求項 2 7】

少なくとも前記 I P アドレスの最初の部分は、他の部分とは別に暗号化されることを特徴とする請求項 2 6 に記載のルータ。

【請求項 2 8】

前記ルータは、少なくとも前記最初の部分とは異なる前記 I P アドレスの部分を読解することができないことを特徴とする請求項 2 7 に記載のルータ。

【請求項 2 9】

前記 I P アドレスは、ルーティングドメインツリー内のルータの統合により、少なくとも部分的に分割され、

前記ルーティング領域ツリーは、前記相互接続されたルータ群の間の相互接続を表し、
前記統合は、共通の親ルータを共有する複数のルータに対して行われることを特徴とする請求項 2 6 に記載のルータ。

【請求項 3 0】

前記ルータは、第 1 の期間中に、少なくとも前記 I P アドレスの一部を解読することができ、

前記ルータは、前記第 1 の期間中に、少なくとも前記 I P アドレスの一部を解読することができ、或いは、第 2 の期間中に、前記 I P アドレスの別の部分を解読することができることを特徴とする請求項 2 6 に記載のルータ。

【請求項 3 1】

少なくとも前記 I P アドレスの第 1 の部分は、鍵を用いて解読され得ることを特徴とする請求項 2 7 に記載のルータ。

【請求項 3 2】

前記ルータは、共通リンクを共用するルータを含む前記相互接続されたルータの部分集合の構成員であり、

前記鍵は、前記共通リンクを共用するルータによって共用されることを特徴とする請求項 2 6 に記載のルータ。

【請求項 3 3】

前記ルータは、少なくともクロスオーバールータを含む前記相互接続されたルータの部

10

20

30

40

50

分集合の構成員であることを特徴とする請求項 2 6 に記載のルータ。

【請求項 3 4】

前記鍵は、前記クロスオーバールータと同じレベルのルータによって共用されることを特徴とする請求項 3 3 に記載のルータ。

【請求項 3 5】

前記ルータは、論理ツリー内に構成されている相互接続されたルータの部分集合の構成員であることを特徴とする請求項 2 6 に記載のルータ。

【請求項 3 6】

前記 I P アドレスは、該 I P アドレスに対するハッシュ関数の結果として、期待値が生成される特性を含み、

10

前記期待値は、前記データの宛先への所望のパスを表すことを特徴とする請求項 2 6 に記載のルータ。

【請求項 3 7】

前記期待値は、前記 I P アドレスのサフィックスに対応することを特徴とする請求項 3 6 に記載のルータ。

【請求項 3 8】

前記期待値は、前記 I P アドレスに含まれていることを特徴とする請求項 3 6 に記載のルータ。

【請求項 3 9】

前記期待値は、前記 I P アドレスに明示的に含まれていないが、前記 I P アドレスは、前記期待値に対するポインタを含み、

20

前記期待値は、テーブル内に含まれていることを特徴とする請求項 3 6 に記載のルータ。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

（関連出願の相互参照）

本出願は、（2003年6月24日に出願された）仮出願第60／482，350号及び（2003年9月3日に出願された）仮出願第60／500，157号の利益を主張する。本出願は、これらの仮出願の開示を援用する。

30

【0 0 0 2】

本発明は、コンピュータネットワークセキュリティに関し、具体的には、潜在的な攻撃者に対して、ロケーションアドレス情報を与えない技術に関する。

【背景技術】

【0 0 0 3】

I P v 6（Internet Protocol version 6）等のインターネットプロトコル（I P）のアドレス指定は、ユーザの地理的な位置に関する情報を明らかにすることが可能であり、ハッカーや他の敵対者に対して、ユーザの位置を突き止める及び／又はユーザの行方を追跡する能力を与える。インターネットプロトコルアドレスにおけるトポロジー上の位置は、特定の地理的な位置に対応するため、I P アドレス指定においては、情報が明らかになっている。例えば、I P v 6は、（一般に、64ビット長の）固定されたサブネットプレフィックスを用いる。サブネットプレフィックスは、I P v 6 アドレスにおいて明確なテキストとして現れ、かつ、I P v 6 アドレスは、送信された情報のパケットヘッダに現れるため、敵対者には、トポロジー上の位置と地理的な位置との索引を生成するのに十分なデータが提供される。従って、敵対者は、このようなデータの対応関係を利用して、ネットワークを用いて地理的な位置を特定する可能性がある。

40

【0 0 0 4】

上記の問題を扱うネットワークセキュリティの分野は、「ロケーションプライバシーセキュリティ（location privacy security）」として知られている。ロケーションプライバシーセキュリティを扱おうとするスキームは、様々な理由で

50

不完全である。例えば、「オニオン（Onion）」ルーティングとして知られているものにおいては、システムは、ネットワークの最後のホップオーバーレイルータと、2つの対応するホストとの間のリンク上で、盗聴者に対して無防備である。また、オニオンルーティングは、ホスト上で実行している悪意のあるソフトウェアから、エンドホストの位置を保護しない。さらに、（インターネット技術標準化委員会（Internet Engineering Task Force、すなわち、IETF）により公表されたIPsec規格等の）インターネットプロトコルセキュリティの共通の形式は、オニオンルーティングと共に用いることができない。また、オニオンルーティングは、情報のルーティングに対して著しい遅延を加える。

【0005】

10

また、「フリーダムネットワーク（Freedom Network）（ゼロナレッジ・システムズ社（Zero-Knowledge Systems, Inc.）による製品）」等のアプリケーションも、同様の理由で不完全である。例えば、フリーダムネットワークは、ホスト上で実行している悪意のあるソフトウェアから、エンドホストの位置を保護しない。さらに、フリーダムネットワークを用いたネットワークフィルタは、発信元IPアドレスを変えることができないため、フリーダムネットワークは、IPsecを利用するホストのロケーションプライバシーを保護することができない。また、フリーダムネットワークは、ルーティングに対して著しい遅延を加える。

【発明の開示】

【発明が解決しようとする課題】

20

【0006】

ロケーションプライバシーセキュリティを実現できる最近の試みは、単一の秘密鍵を用いたIPアドレスの暗号化を含む（このようなシステムは、例えば、同時係属中の、共同で譲渡された米国特許出願第10／284,739号明細書に開示されている）。しかし、単一の秘密鍵の漏洩は、敵対者が、プライバシードメイン全体にアクセスできる可能性を残したままにする可能性がある。

【課題を解決するための手段】

【0007】

従って、本発明の全般的な目的は、暗号で保護されたプレフィックスを用いたインターネットプロトコルのアドレス指定の利用を通じて、ロケーションプライバシーをネットワークに提供することである。本発明は、秘密鍵が漏洩しても、及び、ルータに障害が生じて、ホスト上で実行している悪意のあるソフトウェアから、効果的にロケーションプライバシーを維持する。さらに、本発明は、盗聴者に対して、IPアドレス情報を用いてユーザの地理的な位置を特定する能力を与えない。さらに、本発明は、セキュリティの耐性力及びネットワークの頑強性を提供し、IPsec及び他のセキュリティプロトコルと共に実行する場合、例えば、NAT（Network Address translators）が、エンドツーエンドのセキュリティスキームと共にうまく作動しない場合、トンネリングを伴うことなく、かつ有害な反応を伴うことなく、セキュリティの耐性力及びネットワークの頑強性を提供する。

30

【0008】

40

本発明は、その構成及び動作の方法の両方に関して、以下の詳細な説明と共に解釈すれば、図1～図21を含む図面を参照して、より深く理解することができる。

【発明を実施するための最良の形態】

【0009】

本発明の非限定的な実施形態に関する以下の説明は、特定の構成及び構成要素を開示する。しかし、本実施形態は、本発明の単なる例に過ぎず、従って、以下に記載した特定の特徴は、単に、このような実施形態を説明し、かつ本発明の全体的な理解をもたらすために用いられるに過ぎない。

【0010】

従って、当業者は、本発明が、以下に記載された特定の実施形態に限定されないことを

50

容易に理解する。さらに、当業者に知られている本発明の様々な構成及び構成要素の説明は、明確さ及び簡潔さのために省略する。

【0011】

本願明細書において、IPv6は、インターネットプロトコルのアドレス指定の一例として用いられているが、本発明は、IPv4や他のプロトコルにも適用できる。

【0012】

本発明の例示的な実施形態は、「暗号で保護されたプレフィクス(CPP)」を用いてIPアドレスを生成する。従って、CPP IPアドレスと、ホストの地理的な位置との間の対応関係は、容易に使用可能とならない。CPPを用いる場合、IPアドレスは、多数のセグメントのアドレスのプレフィクスに細別される。各セグメントは、アクセスネットワークドメイン(又は、プライバシードメイン(PD))におけるルータの部分集合にのみ知られている暗号鍵を用いて暗号化される。そのため、各ルータは、情報のパケットを転送するのに必要な情報を取得するが、追加情報については取得しない。

【0013】

上記ルータは、レベルと呼ばれる部分集合に細分化される。様々な実施形態において、所定のレベルに属するルータの集合は、全て、単一の暗号化秘密鍵を共用する。加えて、各ルータは、直接付加されているピアと(なお、ピアは、2つのルータがレイヤ2リンクを共用するとき、直接付加されていると考えることができる)、ユニークで動的な秘密セッション鍵を共用してもよい。第2の集合の鍵は、「リンク」鍵と呼ばれる。

【0014】

インバウンドパケットが、CPPプライバシードメイン内のCPP境界ルータに到達すると、CPP境界ルータは、当該CPP境界ルータのレベル鍵を使用して、プレフィクスの第1の部分を読み解する。この読み解されたプレフィクス成分に基づいて、境界ルータは、当該パケットを、PD内の次のホップルータへ転送する。また、境界ルータは、「ホップバイホップオプション」に、上述の共有リンク鍵で暗号化された最初の読み解されたプレフィクス成分を含ませる。当該次のホップルータは、当該共有リンク鍵を用いて、ホップバイホップオプションから、最初のプレフィクス成分を読み解する。次いで、当該次のホップルータは、当該次のホップルータのレベルの鍵を使用して、CPP IP宛先アドレスから、第2のプレフィクス成分を読み解する。当該次のホップルータは、ホップバイホップオプションからの第1のプレフィクス成分と、レベル鍵からの第2のプレフィクス成分とを連結し、連結したプレフィクス成分を、転送するテーブルエントリに対して一致検索(マッチング)するアルゴリズムに基づいて、当該パケットを転送する。

【0015】

転送されたパケットも、新たな次のホップルータと共用されるリンク鍵を用いて暗号化され、連結されたプレフィクス成分を含む新たなホップバイホップオプションを含む。このようにして、当該パケットが、宛先ホストが接続されているアクセスルータへ受け渡されるまで、宛先IPアドレスのプレフィクス成分が読み解される。プレフィクス全体は、アクセスルータに対して使用可能であるため、転送テーブル内のエントリとの的確な一致結果を得ることができる。従って、CPPは、CPPアドレスを生成して発行するために、及び、ルータ上の鍵を管理するのを手助けするために(これらは、本願明細書に詳細に説明されている)、アドレスサーバ及び鍵生成サーバ等のいくつかの追加サーバを必要とする。

【0016】

図1は、例示的なネットワークにおけるCPPを用いた「インバウンドルーティング」を示す。かかる例示的なネットワークにおいて、IPアドレスは、6ビットのプレフィクスを有している(図では、6ビットが使用されているが、当業者には、どのようなビット数を有するプレフィクスも用いることができることは容易に理解でき、例えば、IPv6ネットワークは、64ビットのプレフィクスを用いてもよい)。各リンクには、ルートから最も遠いルータR1が、ルートに最も近いルータに通知するプレフィクスがラベル付けされている。例えば、R2は、プレフィクス「10100」をR1へ通知する。ユーザホ

ストは、アクセスルータ R 5 に接続されている。R 5 用の 6 ビットのプレフィクスは、「1 0 1 0 0 1」である。

【0 0 1 7】

ユーザホストは、次のように I P アドレスを得る。すなわち、プレフィクス「1 0 1 0 0 1」は、3 つのプレフィクス成分、すなわち、

「 $P_0 = 1 0 1 0$ 」、

「 $P_1 = 0$ 」、

「 $P_2 = 1$ 」、

で構成されている。

【0 0 1 8】

「 P_0 」は、外界へ通知されるプレフィクス成分である。I P アドレスのサフィックスは、「M」であり、「M」は、ランダムに生成される、さらに、

「 $X_1 = H(K_1, M) \text{ XOR } P_1$ 」、

「 $X_2 = H(K_2, M) \text{ XOR } P_2$ 」

【0 0 1 9】

「 $H(K_i, M)$ 」の最初のビット ($i = 1, 2$) は、排他的論理和関数 (すなわち、XOR) として用いられる。鍵 K_1 は、R 1 に知られている秘密鍵であり、鍵「 K_2 」は、R 2 及び R 3 に知られている秘密鍵である。「 K_1 」及び「 K_2 」は、レベル鍵である。ユーザの I P アドレスは、「 $Addr = P_0, X_1, X_2, M$ 」である (ただし、「A, B」は、ストリング A とストリング B との連結を示す)。

【0 0 2 0】

図 1 において、R 1 は、ユーザのホストに宛先が指定されているパケットを受け取る境界ルータである。R 1 は、鍵「 K_1 」を用いて、「 $P_1 = X_1 \text{ XOR } H(K_1, M) = 0$ 」を計算する。次に、R 1 は、(「 $P_0 = 1 0 1 0$ 」と「 $P_1 = 0$ 」とを連結することにより) プレフィクス「1 0 1 0 0」を形成して、R 1 の転送テーブルの 1 0 エントリ (すなわち、「1 0 1 0 0」及び「1 0 1 0 1」) に対する「最長マッチ (longest match)」を行う。「1 0 1 0 0」に対する一致検索に成功したため、R 1 は、当該パケットを R 2 へ転送する。R 1 は、ホップバイホップオプションに、R 2 と共用する鍵で暗号化されたプレフィクス「1 0 1 0 0」を含ませる。R 2 は、ホップバイホップオプションを解読することにより「1 0 1 0 0」を得る。

【0 0 2 1】

R 2 は、鍵「 K_2 」を用いて、「 $P_2 = X_2 \text{ XOR } H(K_2, M) = 1$ 」を計算する。次に、R 2 は、(ホップバイホップのプレフィクス「1 0 1 0 0」と「 $P_2 = 1$ 」とを連結することにより、) プレフィクス「1 0 1 0 0 1」を形成して、R 2 の転送テーブルの 1 2 エントリ (すなわち、「1 0 1 0 0 0」及び「1 0 1 0 0 1」) に対する「最長マッチ」を行う。

【0 0 2 2】

「最長マッチ」は、「1 0 1 0 0 1」を生成し、R 2 は、当該パケットを、ホップバイホップオプションにおける暗号化されたプレフィクス「1 0 1 0 0 1」と共に R 5 へ転送する。R 5 は、的確な一致結果を得たので、「1 0 1 0 0 1」を解読して、当該パケットをローカルインタフェースの外のユーザホストへ転送する。

【0 0 2 3】

C P P によってもたらされる 1 つの利益は、各ルータに対して使用可能な情報の量が最少化されることである。例えば、R 1 は、当該パケットが、ルータ R 2 をルートとするツリーの配下のホストに宛先が指定されていることを学習するのみである。そのため、R 1 に知られる秘密鍵の漏洩は、例えば、米国特許出願第 1 0 / 2 8 4, 7 3 9 号明細書に記載されている I P アドレスのスクランブルスキームで生じる可能性のある損失と比べて、ロケーションプライバシーの部分的な損失を発生させるのみである。

【0 0 2 4】

追加実施形態において、本発明は、C P P と共に、「統合 (aggregation)

10

20

30

40

50

」と呼ばれるものも用いる。ルータ転送テーブルにおけるエントリ数を低減するために、ルータは、いくつかの入力プレフィクスが、単一の出力プレフィクス通知に統合されるように、入力ルータ通知を統合するように構成してもよい。例えば、図1において、R2は、入力通知「101000」及び「101001」を取得して、かかる入力通知を単一の出力通知「10100」に統合する。他のルータは、「10100」で始まるいずれのプレフィクスもR2へ転送する。

【0025】

統合においては、プレフィクス「P」は、2つのストリング、すなわち、「P₀」と「P_s」とで構成されている。ここで、「P₀」は、PDからインターネットへ通知されるグローバルルーティングプレフィクスであり、「P_s」は、パケットをPD内へ転送するの

10

【0026】

一般的なルーティングの例は、「G=(V, E)」で例示される。ここで、頂点集合「V」は、PD内のルータによって構成される。エッジ集合「E」は、PDルータ間の全てのレイヤ2リンクを含む。境界ルータの1つは、PDルーティンググラフのルートとして選択され、他の境界ルータは、新たなグラフ「H」を得るために、グラフから排除される。Dijkstra著の「A Note on Two Problems in Connection with Graphs, Numerische Mathematik, 1, pp. 269-271 (1969)」に開示されているもの等のSPF (Shortest Path First) アルゴリズムが、「H」からスパニングツリー「T」を計算するのに用いられる。

20

【0027】

統合は、既存のネットワークにおいて、転送テーブル内のエントリ数を低減するのに用いられるが、本発明の実施形態は、暗号化のためにIPアドレスを細分化するために、統合を用いる。ツリー「T」内のエッジの各々には、ネットワークにおいて対応するリンクに沿って流れるルーティングプロトコルのプレフィクスがラベル付けされている。図1は、このようなラベル付けの一例を示す。頂点に流れるプレフィクスの集合を仮定すると、頂点は、同じプレフィクスの集合と、当該プレフィクスの集合に属するいずれのプレフィクスも送出することができる。

30

【0028】

別法として、頂点「v」は、出力プレフィクスのうちの少なくとも1つを切り捨てることができる。例示的な実施形態において、プレフィクスは、以下の場合に切り捨てられ得る。すなわち、

(1) 切り捨てられたプレフィクスが、入力プレフィクスの集合における少なくとも2つの元のプレフィクスのプレフィクス（初期サブストリング）である場合、及び

(2) 切り捨てられたプレフィクスに一致する（IPアドレスのプレフィクスのいくつかの初期サブストリングは、切り捨てられたストリングに的確に一致する）IPアドレスにおけるプレフィクスが、頂点「v」をルートとするサブツリーの外部のルータ（すなわち、他の頂点）に属していない場合。このような場合、元の集合における2つ以上の一致したプレフィクスは、出力通知プレフィクスの集合における切り捨てられたプレフィクスと置換される。この切り捨てを、統合と呼ぶ。頂点は、統合が実行される場合及び場合のみ、ツリー「T」内のレベル鍵（及び、レベルルータ）に割り当てられる。

40

【0029】

ツリーの頂点には、次のようにレベル付けがなされている。すなわち、ツリーにおける頂点が、統合を行う場合で、かつ、ルートへのユニークなパスにおいて、当該頂点の上方で第1の統合を行っている頂点が、レベルiを有する場合、当該頂点は、レベル(i-1)を受け取る。このことは、ツリーにおけるルートからリーフまでの最長パスにおいて、アクセスルータの真上のレベルのルータが、レベル1としてラベル付けされるように取り決められている。（このことは、ルートが、レベル1を有し、かつ、番号が小さくなるに

50

つれてレベルが増加するように、逆順で、第1の番号付けを行うことによって、決められる。ツリーのリーフに到達した場合で、仮に、dレベルが、当該ルートからリーフまでの最長パス上に割り当てられているとすると、レベルdを有するルートにおいて再番号付けが始まる。)

【0030】

そして、他の境界ルータは、レベルdを有する頂点としてグラフに付加される。これらの他の境界ルータが、どのようにグラフに接続しているかにより、当該他の境界ルータは、それらに対する他のレベルの関係により、(レベルdの鍵に加えて)追加のレベル鍵及び追加の構成を必要とする可能性がある。最良のケースにおいては、当該他の境界ルータは、元のルート境界ルータに必要とされる以上のどのような追加の鍵又は構成も必要としないであろう。

10

【0031】

レベル鍵を有するルータは「レベルルータ」とみなされる。各ルータRは、(適切な鍵設定プロトコルを介して、または、構成を介して)、当該ルータRの「直接のピア(immediate peer)」との間で共有秘密鍵を設定する。直接のピアは、レイヤ2リンクを介してルータRに接続されているピアである。各レベルルータは、当該レベルルータのレベルに対する鍵のコピーを有する。

【0032】

かかるレベルルータは、プレフィクス成分も形成する。プレフィクス「 P_S 」は、上述の境界ルータと上述のアクセスルータとの間で行われる統合から得られるプレフィクス成分によって構成されている。

20

【0033】

例えば、図2に示すように、「 $P_S = P_1, \dots, P_K$ 」である。ルータ24及び26は、統合を行うレベルルータである。24より下の低いレベルのルータは、リンク32を介してルータ24まで、「 $P_0, P_1, \dots, P_i, P_{(j+1, 1)}$ 」、「 $P_0, P_1, \dots, P_i, P_{(j+1, 2)}$ 」及び「 $P_0, P_1, \dots, P_i, P_{(j+1, r)}$ 」を通知する。そのため、ルータ24は、通知を、その後リンク30を介してルータ26へ通知される「 $P_0, P_1, \dots, P_i$ 」に統合することができる。

【0034】

「 $P_{(j+1, 1)}, \dots, P_{(j+1, r)}$ 」は、プレフィクス成分として定義される。そして、ルータ26も、「 $P_0, P_1, \dots, P_{(i-1)}$ 」を統合して、リンク28を介して上方へ通知する。例えば、レベルルータ24は、プレフィクス「 $P_1, \dots, P_i$ 」に対するターゲットであり、プレフィクス成分「 $P_{(j+1, 1)}, \dots, P_{(j+1, r)}$ 」の全てを格納できると考える。ここで、「 $P_{(j+1, 1)}, \dots, P_{(j+1, r)}$ 」は、全て、考えられ得るプレフィクス成分であり、「 $P_0, P_1, \dots, P_j, P_{(j+1, 1)}$ 」、「 $P_0, P_1, \dots, P_j, P_{(j+1, 2)}$ 」及び「 $P_0, P_1, \dots, P_i, P_{(j+1, r)}$ 」は、プレフィクスである。また、ルータは、当該ルータに転送してもよい他のプレフィックス用の追加のプレフィクス成分も格納しなければならない(上述のプライバシードメイン内の他のどこかにおけるルータ障害又はリンク障害によるものを含む)。

30

40

【0035】

次の実施形態においては、レベルルータは、実際のプレフィクス成分を格納する必要はなく、代わりに、より少ない情報を格納してもよい。ルータは、解読する最長のプレフィクス成分に対応する最大長を格納してもよい。したがって、統合されたルートは、統合している頂点のサブツリーに属さなければならないので、統合の定義から考えると、それ自体が他のプレフィクス成分のプレフィクスであるプレフィクス成分がないということになる。それ故、全てのプレフィクス成分は、最大数のビットを解読し、かつ、プレフィクス成分の格納リストに対して一致検索を行うことによって、識別可能であってもよい。

【0036】

全てのプレフィクス成分が、同じ長さである場合で、かつ、かかる長さについての考え

50

られ得るストリングの全てが、プレフィクス成分である場合には、プレフィクス成分を実際に格納する必要はないが、その代わり、プレフィクス成分の長さを格納するのが適切である。かかる長さを格納することにより、当該ルータは、どのくらいのビットを解読するのかを知ることとなる。

【0037】

例えば、「H」を、一方向性ハッシュ関数とする。一方向性のため、「x」ではなく、「H(x)」が、「H」のドメイン内のいずれかの「x」に対して与えられた場合、「x」を計算するのが困難であることを意味する。連結は、「|」又は「コンマ(,)」のいずれかによって表される。dレベルを有する上述のルーティンググラフ「G」が与えられ、ただし、dレベル鍵は、「 $K_1, \dots, K_d$ 」である。鍵「 K_1 」は、レベルdのルータによって共用される鍵であり、鍵「 K_2 」は、レベル(d-1)のルータによって共用される鍵であり、…、鍵「 K_d 」は、レベル1のルータによって共用される鍵である(レベルdのルータは、境界ルータであり、レベル1のルータは、アクセスルータとレイヤ2リンクを共用する)。

10

【0038】

図3は、IPアドレスのプレフィクス40からのCPP IPアドレス54の構造を示す。ここで、要素41は、要素40から要素54への変換を示す。すなわち、プレフィクス「 $P = P_0, P_1, \dots, P_k$ 」とすると、CPPアドレスは、同じグローバルルーティングプレフィクス「 $P_0(42)$ 」を維持する。さらに、「 $X_j = \text{Trunc}(H(K_j, M)) \text{ XOR } P_j (1 \leq j \leq k)$ 」である。ここで、「 $\text{Trunc}(H(K_j, M))$ 」は、「 P_j 」に含まれるビット数に対して切り捨てられた「 $H(K_j, M)$ 」を表す。

20

【0039】

混同のおそれがない場合、単純に「 $H(K_j, M)$ 」と書き、「 $\text{Trunc}(H(K_j, M))$ 」を意味するものとする。図3において、要素44、46及び48は、それぞれ、「 X_1 」、「 X_2 」及び「 X_k 」を示す。そして、図3の要素54によって示されるCPPアドレスは、「 $A = P_0, X_1, X_2, \dots, X_k, Y, M$ 」である。ここで、「52(M)」は、アドレスのサフィックスであり、「42(P_0)」は、PD用のグローバルルーティングプレフィクスであり、「50(Y)」は、内部ルーティングを最適化するのに用いられる成分である。(IPv6の場合、「 P_0 」は、最初の3ビットが「0」に設定されるように取得されることができ、アドレスのプレフィクスとホスト識別子との間の境界が、64ビット境界に該当しないことを表す。)(「Y」については、以下で論じる。)

30

【0040】

2つの追加ビット56(「l」及び「v」)は、上述のアドレスについてのロケーションプライバシーが保護されているか否かについて、現在の鍵のバージョンとを表すのに用いられる。代わりに、ルーティングドメインが、ロケーションプライバシーが保護されていないIPアドレスに対応する一方の「 P_0 」を得て、ロケーションプライバシーが保護されているアドレスに対応する他方の「 P_0 」を得ることを可能にしてもよい。このような代替例の場合においては、別々の1ビットは必要ではない。例えば、2つの「 P_0 」が、ビット0～ビット46に渡って同一であり、かつ、右端の(47番目の)ビットにおいてのみ異なる場合には、プライバシードメインルータは、単に、ビット47を検査して、IPアドレスについてのロケーションプライバシーが保護されているか否かについて識別すればよい。

40

【0041】

上述のアドレスについてのロケーションプライバシーが保護されているか否かについて識別するのに「 P_0 」を用いることによって、サブネット識別子の全16ビットを、非ロケーションプライバシーIPアドレス用のサブネットの識別に用いることが可能になる。そうではなく、上述したような1ビットの場合には、非ロケーションプライバシーIPアドレスは、(64ビット未満が、ホスト識別子に用いられない限り、)サブネットの識別

50

用に使用可能な15ビットを有するのみである。鍵のバージョンのビット「v」は、どの鍵が、C P P I Pアドレスを暗号化するのに用いられたかについて示すのに用いられる。

【0042】

各ルータは、3鍵期間まで、鍵を維持してもよい。すなわち、各ルータは、「現在の鍵期間」、「直前の鍵期間」、ルータが次の鍵期間用の鍵を取得している場合に必要であれば「次の鍵期間」において、鍵を維持してもよい。図4は、鍵更新の動作と、アドレス割り当て及び鍵更新の相互作用とを示す。要素150は、等しい長さの別々の期間に分割されたタイムラインを示す。一つの可能性は、各鍵期間が1日続くことである。より短いまたはより長い他の期間も、セキュリティに係る必要性及び他の理由により可能である。各ルータは、次の鍵期間154が始まるまで、次の鍵期間用の鍵を取得しなければならない。例えば、各ルータは、現在の鍵期間152の後半中に、次の鍵期間用の鍵を取得してもよい。期間i中に取得されたアドレスは、期間i+2(154)の開始前に期限が切れなければならない。期間i中に発行されたアドレスは、期間iの鍵を用いて生成される。そのため、ルータは、鍵を有していない期間中は、有効なアドレスに出会うことはない。

【0043】

上述の構造の場合、「M」は、上記レベル鍵の生存期間中、ユニークであるべきである。換言すれば、「M」は、同じレベル鍵の集合と共に、異なるI Pアドレスの部分として再使用されるべきではない。

【0044】

図5は、C P Pが可能なプライバシードメインルータ(又は、単にC P Pルータ)が従う処理手順を示す。C P Pルータは、機能160において開始する。C P Pルータは、機能162において、C P PルータがC P Pプライバシー保護されたパケットを処理できるようにする暗号鍵を、(鍵サーバから)取得する。C P Pルータは、機能164において、かかる暗号鍵を使用して、C P Pパケットを転送する(164における機能は、さらに、図7に関連して本願明細書に説明されている)。C P Pルータは、機能152において、鍵サーバから未使用の暗号鍵を得る。機能164において、C P Pルータは、C P Pパケットを転送するのを継続する。

【0045】

C P Pを利用した他の実施形態では、I S Pルータが、インバウンドパケットを処理する。レベル(d-j)のルータであるルータR1が、「宛先I Pアドレス:A=P₀, X₁, X₂, ..., X_K, Y, M」を有するパケットを受信すると仮定する。

【0046】

図6に示すように、パケットも、共用秘密リンク鍵を使用して暗号化されている「P₁, ..., P_j」を含むホップバイホップオプション60を含む。図6に示すパケットの「種別」は、判断されることになっている。「長さ」は、(例えば、I P v6によって指定された)オプションの長さから得られる。パケットの「オフセット」は、次の解読を始めるビット位置によって示されている(このフィールドは、4ビットに縮められてもよい)。「プレフィクス長」は、「暗号化されたプレフィクス」内のビット数によって示されている(「暗号化されたプレフィクス」フィールドは、考えられ得る最長のプレフィクスを収容するために、固定長であるが、当該フィールド内の全てのビットが、必ずしも送信される「暗号化プレフィクス」の一部である必要はなく、このフィールドもまた、4ビットに縮められてもよい)。従って、「暗号化プレフィクス:=H(K, M) XOR (P₁, ..., P_j)」である。ここで、「K」は、共用秘密リンク鍵である。「ターゲットレベル」は、解読すべき次のレベルである。「ターゲットレベル」が「0」の場合、プレフィクスの全てのレベルが既に解読されている。

【0047】

上述の共用秘密リンク鍵は、R1と直前のホップとの間で共用される。例えば、「K」を、共用秘密リンク鍵とする。そして、「H(K, M) XOR (P₁, ..., P_j)」は、ホップバイホップオプションの「暗号化プレフィクス」フィールド内に含まれる。

【0048】

R1は、第1に、「ターゲットレベル」フィールド内に含まれるレベルが、R1のレベルに等しいか否かについて検査する。等しくない場合、R1は、ホップバイホップオプションからプレフィクスを得て、転送アルゴリズムに当該プレフィクスを用いる。等しい場合は、R1は、当該レベル鍵を用いて、考えられ得る最長のプレフィクス「 $X_{(j+1)}$ 」を解説する。したがって、この長さは、上述したように、予め設定されている。R1は、「 $X_{(j+1)}$ 」から「 $P_{(j+1)}$ 」を得る（すなわち、「 $P_{(j+1)} = H(K_{(d-j)}, M) \oplus X_{(j+1)}$ 」）。

【0049】

R1は、ホップバイホップオプションの「オフセット」フィールドによって識別されたビットで解説を開始する。ホップバイホップオプションから「 $P_1, \dots, P_j$ 」を得た後、R1は、「 $P_1, \dots, P_j, P_{(j+1)}$ 」を入力プレフィクスとして、転送アルゴリズムを用いる。例示的な実施形態においては、転送アルゴリズムから次のホップ識別を得た後に、R1が、新たなホップバイホップオプション(1)～(4)を生成する。

(1)「暗号化されたプレフィクス」フィールドに含まれる「 $H(K_n, M) \oplus (P_1, \dots, P_j, P_{(j+1)})$ 」。

ここで、「 K_n 」は、R1が次のホップと共用する鍵である。

(2)「暗号化されたプレフィクス」の長さは、「プレフィクス長」フィールドに含まれる。「暗号化されたプレフィクス」フィールドの実際の長さは、常に16ビットであるが、リンクを横断して送信されるプレフィクスは、より短くてもよいことに注意する。「IPsec AH」を用いることが可能なため、オプション全体の長さは、一定に保持しなければならない。

(3)解説を開始する次のビットは、「オフセット」フィールドに含まれる。

(4)現在のレベル+1は、「ターゲットレベル」フィールドに含まれる。

【0050】

本質的に、各ルータは、当該ルータのレベルに関連するプレフィクス情報を得て、かかるプレフィクス情報を用いてパケットを転送する。また、ルータは、(より上位のレベルからのプレフィクス情報と連結された)プレフィクス情報と、転送パス上の次のルータとの共用鍵で暗号化されたホップバイホップオプションにおけるオフセットとを含む。そのため、次のルータは、かかる共用鍵を用いて、より上位のレベルのルータに知られているプレフィクス部を得て、当該次のルータのレベル鍵を用いて、当該次のルータのレベルに対応するプレフィクス部を得る。これら2つのプレフィクス部の連結は、次のルータが、当該次のルータの「最長マッチ転送アルゴリズム」で用いるものである。

【0051】

オプションデータは、転送中に変更してもよいので、オプション種別における3番目に高いビットは「1」である。「IPsec AH」が用いられる場合、IPv6は、ビットがなくなる以外は、ホップバイホップオプションが、計算に含まれることを要する。そのため、ホップバイホップオプションのサイズは、プレフィクスのサイズに関係なく一定である(「暗号化されたプレフィクス」フィールドには、16ビットを用いることができる)。

【0052】

CPPは、モバイルIPやHMIPv6等の処理スキームを行う場合、オーバーヘッドIPTONネリングを必要としない。そのため、(モバイルIPやHMIPv6を用いた場合等のように、)追加のIPヘッダに関連する40バイトが節約される。

【0053】

以下、本発明の転送アルゴリズムについて論じる。「FWalg()」が、既存の「最長プレフィクスマッチ転送アルゴリズム(longest prefix match forwarding algorithm)」を示すものとする。IPアドレスのプレフィクスの最初のサブstringの一致検索は、転送テーブルエントリに対して行われる(IPアドレスのプレフィクスの残りは、暗号化される)。従って、いくつかの一致検索

10

20

30

40

50

が存在する可能性があり、それに伴ってアルゴリズムは、わずかな変更を必要とする。

【0054】

パケットからの最初のサブストリングプレフィクス S が、プレフィクスを、より多くのプレフィクス成分に対して一致検索することができないことを除いて、「FWal g*()」は、「最長プレフィクスマッチングアルゴリズム」を示すものとする。換言すれば、「 S 」は、「 S 」と同数の成分を有するプレフィクス又は「 S 」未満の成分を有するプレフィクスのいずれかに対する一致検索に成功することができる。ここで、「 $S = P_0, \dots, P_{(i-1)}, P_i$ 」であり、「 P_i 」は、プレフィクス成分である。従って、「 S 」の一致検索結果は、(最良の一致検索結果から最悪の一致検索結果の順で、)、以下の通りである。

10

S

$P_0, \dots, P_{(i-1)}$

$P_0, \dots, P_{(i-2)}$

$P_0, \dots, P_{(i-3)}$

...

P_0

$P_0, \dots, P_{(i-1)}, Q_1$

$P_0, \dots, P_{(i-1)}, Q_2$

...

$P_0, \dots, P_{(i-1)}, Q_k$

$P_0, \dots, P_{(i-2)}, Q_1$

$P_0, \dots, P_{(i-2)}, Q_2$

...

$P_0, \dots, P_{(i-2)}, Q_k$

...

ここで、「 $|Q_1| \leq |Q_2| \leq \dots \leq |Q_k|$ 」であり、「 Q_i 」は、プレフィクス成分である。また、「 $P_0, \dots, Q_i$ 」は、全て、レベルルータによって支配される。

20

【0055】

さらに、ローカルインタフェースに対しては、厳密な一致検索が必要である(一致検索されるプレフィクスは、入力プレフィクスと等しくなければならない)。そのため、「 S 」が、転送テーブル内のローカルインタフェースのプレフィクスよりも長い場合には、「 S 」は、かかるプレフィクスと一致しない。また、パケットも、解読が行われない限り、当該パケットが到着したインタフェースから転送されない。より長いプレフィクスに対して一致検索することができないとしたのは、より長いプレフィクスが、次のプレフィクス成分を解読できないツリー内のより低いレベルのルータに対応するという考えからである。そのため、次のプレフィクス成分が解読される前に、パケットをより低いレベルのルータへ転送することは役に立たない。

30

【0056】

図7は、ルータが、変形された(CCP)転送アルゴリズムを用いてパケットを転送する本発明の実施形態を示す。プレフィクスが完全に解読されている場合、「FWal g*() = FWal g()」であることに注意する。図7に示すように、

40

【数 1】

```

    decrypt(Prefix, starting_bit, target_level)
    {
        if (hop-by-hop option present AND (hop-by-hop option type == CPP))

            decrypt hop-by-hop option to get the hop-by-hop prefix Q, target level, starting bit for
            decryption; // Figure 7: function 174

            if (we have a level AND target_level == our level) // Figure 7: function 176
                10
                K = our level key;
                S = H(K,M) XOR [starting bit in Prefix up to maximum length for prefix
                component];
                search S for the correct prefix component C (match against stored prefix
                components);
                update starting_bit, target_level;
                return (Q appended with C);
            else // Figure 7: function 182
                20
                return (Q);
            else if (our level == d) // we are a border router. // Figure 7: function 184
                K = our level key;
                S = H(K,M) XOR [starting bit in Prefix up to maximum length for prefix component];
                search S for the correct prefix component C (match against stored prefix
                components);
                update starting_bit, target_level;
                30
                return (C);
            else
                return error; // we shouldn't be here
        }
        CPP_inbound_route (Packet)
        {
            obtain prefix P from Packet;
            P_new = decrypt(P, &starting_bit, &target_level); // Figure 7: function 174
            40
            interface = FWalg*(P_new);
            if (interface == NULL) // no match
                discard P;
            else // Figure 7: functions 176, 182
                create a new hop-by-hop option for P with P_new, starting_bit, target_level,
                encrypt

```

```

with shared link key;
forward Packet out interface;
}
main()
{
while (1)
{
obtain destination IP address A for incoming Packet; // Figure 7: function 192
if (IP address A is not location privacy protected) // Figure 7: function 178
FWalg();

else
{
obtain prefix P from A; // Figure 7: function 186
if (P does not start with P0) // Figure 7: function 188
forward Packet up the default route; // Figure 7: function 180
else if ((P starts with P0) && (hop-by-hop option not present || types of hop-by-hop
options !=
CPP) && (we are not a border router) ) // Figure 7: function 170
optimal_route(Packet); // described in optimal routing section below, Figure
7: function 194
else
CPP_inbound_route (Packet);
}
}
}

```

である。

【 0 0 5 7 】

C P P 転送アルゴリズムの一例は、図 7 の要素 1 6 4 によって示される。図 7 の機能 1 9 2 に示すように、プライバシードメインルータは、入ってくるパケットから宛先 I P アドレス A を得る。機能 1 9 0 において、アドレス A は、それが C P P アドレスであるか否かについて判断するために評価される。C P P アドレスでない場合、機能 1 7 8 は進行して、従来の転送アルゴリズムを用いてパケットが転送される。アドレス A が C P P アドレスである場合には、機能 1 8 6 は、A からプレフィクス P を得る。次いで、機能 1 8 8 に

において、プレフィクス「P」が、「P₀」で始まっているか否かについて判断される。プレフィクス「P」が、「P₀」で始まっていない場合、パケットは、機能180において、境界ルータへ向けてデフォルトルートの上方向へ転送される。プレフィクス「P」が、「P₀」で始まる場合には、170の機能が判断される。

【0058】

機能170においては、CPPオプションがあるか否かについて判断される。CPPオプションがない場合、機能194は、最適化されたルーティングを利用することを要求する。CPPオプションがある場合には、機能172は、パケットが、現在境界ルータにあるか否かについて問い合わせる。パケットがある場合には、機能184は、レベル鍵を用いて最初のPCを解読し、PCを「FWalg*（）」に入力し、それに続いて、次のホップRが得られ、（Rとの）共用秘密リンク鍵を用いて、ホップバイホップオプション内でPCが暗号化される。そして、ホップバイホップオプションは、当該パケット内で転送される。

10

【0059】

機能172において、CPPオプションがない場合、機能174は、共用リンク鍵を用いてホップバイホップオプションを解読し、パケットが、ホップバイホップからのターゲットレベルと等しいレベルを有するレベルルータにあるか否かについて尋ねる。かかるパケットがある場合には、機能176は、「X」の開始ビットにおいて開始することにより（この開始ビットは、ホップバイホップオプションから得られる）、アドレスのX成分から次のPCを解読する。次に、機能176は、ホップバイホップオプションのプレフィクスと新たなPCとを連結して、「Pnew（NextHop=FWalg*（Pnew）」を得る。次いで、機能176は、次のホップへ転送する前に、パケットに挿入されるホップバイホップオプションにおいて共用リンク鍵を用いて「Pnew」を暗号化する。最後に、機能176は、開始ビット及びホップバイホップオプションのターゲットレベルを更新する。

20

【0060】

機能174の判断が、パケットが、ホップバイホップからのターゲットレベルに等しいレベルを有するレベルルータにないことを示す場合、機能182が用いられる。機能182は、ホップバイホップオプションからの解読されたプレフィクス（「FWalg*（）」への入力プレフィクス）を用いて当該パケットを転送し、その後、ホップバイホップオプションを用いて、次のホップとの共用リンク鍵を用いて当該プレフィクスを再暗号化する。

30

【0061】

本発明の別の実施形態は、障害がある場合の転送アルゴリズムの復元力に対して提供する。すなわち、ルーティングリンクに障害が生じた場合には、プレフィクスが統合されるように変化する可能性がある。そのため、より低いレベルのルータは、同じレベル又はより高いレベルのルータを統合することが許されていない。この要求に対する動機は、特定のレベルのプレフィクス成分の解読が必要な場合に、より高いレベルのルータが、他のルータの転送テーブルで認識できるようにすることである。以下の証明のため、非レベルルータは、ルーティングリンクに障害が生じた場合でも、統合を行わない。

40

【0062】

図8は、IPアドレスのプレフィクスが、6ビットを有し（6ビットのうちの2ビットが、サブネット識別子用に用いられる）、転送アルゴリズムが、ある程度の数の障害がある場合を含み、パケットを当該パケットの宛先へ転送することに成功することができるCPPアクセスネットワークの一例を示す。各リンクには、当該リンク上の下位のルータによって通知されるプレフィクスがラベル付けされている。例えば、ルータR3（要素60）は、プレフィクス64（「10101」）を62（ルータR1）に対する上方に通知する。図8においては、R3が故障することが想定される。そして、さらに、プレフィクス「101010」を有するパケットを想定する。R1は、最初のビットを解読して「10101」を得る。R3が故障失敗した場合、R1の転送テーブルは、

50

「1 0 1 0 0 R 2」
 「1 0 1 0 1 0 R 7」
 「1 0 1 0 1 1 R 7」
 「1 0 1 0 ローカル」

になる。

【0 0 6 3】

これらのプレフィクスは、「1 0 1 0 1」よりも長いため、「1 0 1 0 1」は、「1 0 1 0 1 0」又は「1 0 1 0 1 1」に対して一致検索することができない。（このような一致検索が許されていると、パケットは、次のビットを解読できないグラフ内の下位のレベルのルータへ送られることになる。そのため、発明者らは、上述の転送アルゴリズムにおけるこのような一致検索を認めない。）

10

【0 0 6 4】

その代わりに、「1 0 1 0 1」は、（最後の「0」が除去された場合、）「1 0 1 0 0」に対して一致検索する。かかるパケットは、後に解読してプレフィクス「1 0 1 0 1 0」を得るルータ R 2 に転送される。そして、このプレフィクスは、R 2 の転送テーブル：

「1 0 1 0 0 ローカル」
 「1 0 1 0 1 0 R 1」
 「1 0 1 0 1 1 R 1」
 「1 0 1 0 R 1」
 「1 0 1 0 0 0 R 4」
 「1 0 1 0 0 1 R 5」

20

において、当該プレフィクス自体に一致する。

【0 0 6 5】

そして、当該パケットは、その後、R 1 へ転送される。次いで、R 1 は、プレフィクスを R 7 に対して一致検索して、当該パケットを R 7 へ転送する。R 7 は、当該パケットをローカルインタフェースで転送する。こうして、当該パケットは、ルータ R 3 の故障にも関わらず配信される。

【0 0 6 6】

本発明は、一定条件下における次の提案（Lemma 4. 1）によって立証されるような転送アルゴリズムの頑強性も提供する。パケット P が、境界ルータ又はアクセスルータ（エントリポイントルータ）のいずれかを介して、プライバシードメインに到着すると仮定する。また、パケット P が、レベル（ $i + 1$ ）ルータ Q を通過し、当該パケットを処理する次のレベルルータ R が下方にある場合（R が、Q と同じ接続成分にはない場合）、R は、Q と同じ接続成分において兄弟関係のルータを有すると仮定する。ここで、兄弟関係は、Q の子孫でもあるレベル i を有する別のルータとして定義される。そして、当該パケットは、転送プロセス中に、完全に解読される（全てのレベル鍵が適用され、プレフィクス全体がルータに対して使用可能になる）。

30

【0 0 6 7】

立証として、パケット P が、境界ルータに到達すると仮定する。帰納法を用いて、プレフィクス成分「 $P_1, \dots, P_i$ 」が解読されていると仮定する。P は、レベル（ $d - i$ ）のルータへ転送されることが分かっている。R _{i} は、「 P_i 」を解読したレベル（ $d - i + 1$ ）のルータとする。プレフィクス「 $P_1, \dots, P_i$ 」を有するルータである元のスパニングツリーにおいて、R を選択する。R は、レベル（ $d - i$ ）を有する。

40

【0 0 6 8】

ケース a：R が、上方にあり、R _{i} と同じ接続成分にある。R _{i} から R へのパスがあり（統合が行われない場合）、最少のコストのパスを選択し、かかるパスを A と呼ぶ。次に、「 $P_1, \dots, P_i$ 」は、かかるパスに沿った各ルータ用の転送アルゴリズムにおいて的確に一致する。そのため、P は、要望通りに、R へ転送されることになる。

【0 0 6 9】

さもなければ、R _{i} から R への最少の統合及びコストを伴うパス A を選ぶ。R _{i} から R

50

へのパスにおいては、「 $T_1, \dots, T_k$ 」を統合を行うルータとする。そして、「 T_i 」により通知されるプレフィクスは、「 $T_{(i-1)}$ ($T_0 = R_i$)」用の転送アルゴリズムにおいて、一致検索に成功することになる。この最後の言及は、最少の統合を伴うパスを選んだということから当然の結果である。そのため、 P は、要望通りに、 R へ転送される。

【0070】

ケースb： R は、異なる接続成分にある。仮定により、 R_i は、 R と同じレベルを有するルータ T と同じ接続成分にある。そして、「 T 」は、「 $P_1 \dots P_{(i-1)}, Q$ 」を通知する。「 $P_1 \dots P_{(i-1)}, Q$ 」は、(R_i の転送テーブル内の) R_i に対して認識できるか、または、統合されたプレフィクス「 $P_1, \dots, P_j$ 」は、認識できる。最初の
10 ケースの場合、認識できる統合されたプレフィクスがない場合、「 $P_1, \dots, P_{(i-1)}, Q$ 」が、一致検索結果する。このような一致検索は、 R_i と「 T 」との間のルータのつながりにおいて継続し、その結果、「 T 」へ転送されるパケット P を生じる。そして、「 T 」は、要望通りに、当該パケットを解読する。

【0071】

ここで、統合されたプレフィクス「($P_1, \dots, P_j$)」が、 R_i に認識できると仮定する。この最良の一致検索は、解読が行われるまで（及び、立証が完了するまで）、又は、より長いプレフィクス成分シーケンス（「 $P_1 \dots P_k$ 」、ここで、「 $k > j$ 」）が登場するまで、続けられる。さらに、最良の一致検索は、解読が行われるまで（及び、立証が完了するまで）、又は、より長いプレフィクス成分シーケンスが登場するまで、続けられ
20 る。最後に、あるプレフィクス成分「 Q_1 」に対して、「 $P_1, \dots, P_{(i-1)}$ 」とならなければならない。以前に対する最良の一致検索は、このプレフィクスを有するルータが、上述の転送アルゴリズムについての言及によるレベルルータであるルータに到達されるまで続けられる（このルータは、適切なレベル鍵を有する兄弟ルータである）。そのため、次の成分が解読され、立証が完了する。

【0072】

さらに、パケットが、境界ルータ又はアクセスルータ（エントリポイントルータ）のいずれかを介して、上記プライバシードメインに到達すると仮定し、かつ、「（上述した）
Lemma 4. 1」の条件を保持すると仮定する。そのため、当該パケットは、宛先のアクセスルータへ配信される。立証として、「Lemma 4. 1」及び既存の最長プレフィ
30 クスの特性が、ルーティングアルゴリズムに一致すると考える（プレフィクスが、一旦完全に解読されると、「FWalg*() = Fwalg()」）。

【0073】

別の実施形態においては、統合に対する要求の1つを緩和することが可能であり、それにより、本願明細書において正規化と呼ぶプロセスにおいて、追加の統合が可能になる。追加の統合の利点には、プライバシーの向上及びルータ転送テーブルにおけるエントリの数のさらなる低減が含まれる。本実施形態において、プレフィクスの所有権についての必要条件は緩和される（この場合、所有するルータは、当該プレフィクスに対する統合を行
40 っているルータ、或いは、当該プレフィクスと等しいIPアドレスを有するルータである）。すなわち、統合を行っているルータのサブツリーにおけるルータは、プレフィクスの所有権を所有する必要はない。それに伴って、統合されたプレフィクスは、スパニングツリー「 T 」内にないリンクを横断して通知されるルートを含んでもよく、その代わり、当該リンクは、ルーティンググラフ「 G 」に含まれてもよい。

【0074】

図9は、プライバシードメインルーティンググラフ「 G 」の一部を示す正規化の一例を示す。この例もまた、プレフィクス内において限定された数のビットを用いる。しかし、リンク80が、元のスパニングツリー「 T 」内に存在しないリンクであることに注意する。そのため、要素76（ルータ R_2 ）が、リンク80を受け入れた場合には、上述したような厳密な統合は起こらない。すなわち、ルータ R_2 が、統合用のプレフィクス「10100」及び「101011」の通知を受け取るのみである場合、通常の統合である。しか
50

し、「101010」は、ルータR2をルートとするサブツリーの外にあるので、ルータR2は、ルータR8からのリンク「101010」の受け入れのため、統合の通常の規則に従わないことになる。

【0075】

統合の必要条件を緩和することにより、ルータR2は、(80によって通知された「101010」と共に、)プレフィクス「10100」及び「101011」をルータR1における「1010」と統合できるようになる。そのため、要素72(ルータR1)では、R1の転送テーブルにおいて、1つエントリが減ることになる。プライバシーの観点から、R1は、もはや、パケットが、82(ルータ7)に向けられているか否かについて検出することはできなくなる。従って、ルータR2(要素76)が、(要素80に通知された「10101」と共に、)プレフィクス「10100」及び「101011」を「1010」に正規化できるようにした場合、プライバシーは向上する。

10

【0076】

本発明者等は、ルータR2が「1010」に正規化して通知できるようにすることに関して問題がある可能性があることに注目した。要素70は、リンク74によって「101010」を72(ルータR1)に対する上方へ通知する。要素72は、「1010」が意図されたプレフィクスであるかどうか、又は、「101010」を生じる2つの追加ビットの解釈が意図されたプレフィクスであるかどうかについて検知することができなくてもよい。すなわち、意図されたプレフィクスは、「101011」であってもよいし、「101010」であってもよい。「101010」の解釈結果は、最後の2ビットが、ルータR2の鍵に従って暗号化された「101011」であってもよい。この問題は、「1010」を「10100」として符号化することにより解決することができる。従って、この符号化を用いて、ルータR1は、2つの追加ビットを解釈する(ルータR1は、「1010」をホップバイホップオプションで受け取る)。

20

【0077】

このように、ルータR1が上述の解釈から得るビットパターンは、「0x」又は「10」のいずれかである。ここで、「x」は、「0」又は「1」とすることができる。第1のケース(「x=0」)においては、パケットは、「1010」をプレフィクスとして含むホップバイホップオプションを用いて、要素76(ルータR2)へ転送される。ここで、符号化されたプレフィクス(「10100」)が送信されるのではなく、転送テーブルのプレフィクスに対応する実際のプレフィクス「1010」が送信される。従って、「10100」は、「1010」に対してマッピングされる。第2のケース(「x=1」)においては、パケットは、「101010」のホップバイホッププレフィクスを用いて、要素70(ルータR8)へ転送される。

30

【0078】

統合前、要素76(ルータR2)は、レベルを伴わないルータであり、要素72(ルータR1)は、レベル2のルータである。要素76は、統合を行うことができるようになった後、レベル2のルータとなり、要素72は、レベル3ルータとなる。72より上位の全てのルータは、当該ルータのレベルを1だけ増す。要素78(ルータR4)は、統合の前後において、レベル1を有している。プレフィクス成分(PC)の符号化を用いて、2つの異なるPCの集合が導入される。第1の集合は、上述の転送テーブル内のプレフィクスの一部である実際のPCである。第2の集合は、C P Pアドレスにおいて暗号化されている集合である。第1の集合は、「C__R」と設定され、第2の集合は、「N__R」と設定されており、この符号化方法は、「正規化」と呼ばれている。

40

【0079】

追加の統合が導入された場合、統合を行っている頂点をルートとするスパニングツリーのサブツリー内の統合を行っているルートのみに関して、統合の制約を緩和することにより、レベルが、ルーティンググラフ「G」内で生じる。レベルルータではないルータRが、ここで1つになると仮定する。Rをルートとするサブツリー内のルータは、同じレベルを保持する。Rは、(スパニングツリー内の)Rの子よりも大きいレベルを有する。当該

50

グラフ「G」内の他のレベルのルータは、全て、1だけ増加したレベルを有する。当然、統合を行っている頂点が、既にレベルルータである場合には、ルータのレベルは変化しないことに注意する。

【0080】

PCの正規化された集合は、各ルータに対するプレフィクスにおいて符号化されることが要求される。PCの正規化された集合を用いて、PCは、他のPCのプレフィクスとすることができない。そのため、PCを解説する場合、曖昧さはない。正規化されたPCを解説した後、(集合「N_R」に属する)正規化されたPCは、ルータR用のPCの元の集合「C_R」内のPCにマッピングされ得る。

【0081】

ここで、「C_R」を、ルータRにおけるPCの集合とする。ユーザホスト用のサブネットワークを識別するプレフィクスを「P」とすると仮定する。「P」を、「 $P = P_1 | P_2 | P_3 | \dots | P_k$ 」と書くことができる。ここで、「|」は、連結を示し、「 $P_1 | P_2 | \dots | P_i$ 」は、(d-i)番目のレベルのルータ用の一致検索を行うプレフィクスである。(d-i)番目のレベル用の鍵を「 K_i 」で表す。

【0082】

ルータRにおける可能性の有る「 P_i 」の集合は、「C_R」である。この場合、「C_R」の正規化されたバージョンが必要である。「N_R」は、以下に示す「Definition 6. 2」の特性を満たす集合である。

「Definition 6. 1」：ストリングSが、ストリングTのプレフィクスである場合、(S, T)が、プレフィクス対である。

「Definition 6. 2」：「N_R」は、集合であり、全単射f（全単射は、1対1及び上への関数である）は、次の特性を満たす。

- 全単射 $f : N_R \supseteq C_R$
- 「N_R」の構成員である任意の2つのストリング「S」及び「T」に対して、「S」は、「T」のプレフィクスではない。

【0083】

「C_R」が、「S」が「T」のプレフィクスであるようなストリング「S」及び「T」を有していない場合、「 $N_R = C_R$ 」である。さもなければ、あるストリング「T」のプレフィクスである「C_R」におけるストリング「S」を想定する。ここで、「T」は、「S」がプレフィクスである「C_R」における全てのストリングのうちの最短のものである。

【0084】

集合「 $E = \{SR : R \text{ は、ストリングであり、} |SR| = |T| \text{ である} \}$ 」は、「C_R」に含まれていない。さもなければ、「最長プレフィクスマッチルーティングアルゴリズムは」、ストリング「S」を選択しない。「 $|SY| = |T|$ 」となるように、「E」内にはない「Y」を選択する。集合「C_R」から「S」を取り除いて、「SY」を新たな集合「N1_R」に加える。

【0085】

それに応じて、第1の命題は、「N1_R」が、「C_R」よりも厳密に少ないプレフィクス対を含むことを提示する。第2の命題は、「S」に対する「SY」のマッピングを除いて「N1_R」上の識別であるマッピング「 $f1 : N1_R \rightarrow C_R$ 」が全単射であることを提示する。第2の命題は、明白である。第1の命題は、「SY」が「S」よりも厳密に少ないストリングのプレフィクスである（「SY」は、「T」のプレフィクスではない）ため、理解される。また、「SY」のプレフィクスである任意のストリングは、「S」である、「S」のプレフィクスである、又は、「S」よりも大きい「Y」よりも小さい長さを有する。最後のケースは、「T」が、「S」が「T」のプレフィクスである最短のストリングであるため、不可能である。最初のケースは、「S」が「N1_R」内にはないため、不可能である。

【0086】

(新たに生成された各集合は、直前のものよりも少ないプレフィクス対を有するので、終わらせなければならない) 直前の構造は継続される。したがって、「 $C_R \geq N1_R \geq \dots \geq Nk_R = N_R$ 」が得られ、「 N_R 」が、「 N_R 」から「 C_R 」への全単射を伴う所望の集合となる。

【0087】

「 N_R 」は、暗号化されたプレフィクス成分のコンパクトな符号化を提供する。具体的には、各ルータは、考えられ得る最長のプレフィクスを解読することができ、その後、どのような曖昧さに対する心配も伴うことなく、一致するものを探索することができる。

「 N_R 」は、プレフィクス成分の符号化に用いられる。ルータは、「 N_R 」のストリングを「 C_R 」のストリングにマッピングして、上述の転送アルゴリズム用の適切な一致するストリングを得る(そのため、転送テーブルのエントリは、このようなマッピング及び符号化に影響を及ぼされない)。しかし、プレフィクス成分の符号化は、実際のプレフィクスよりも多少長くてもよいことに注意する。ルーティングトポロジー及びプレフィクスが、統合の量を最大化するように設定されている場合、サイズの増加の程度は、最少にされ得る、或いは、無視され得る。

【0088】

本発明の別の実施形態は、最適化されたルーティングを提供する。パケットが、上記プライベートメイン内で、アクセスルータに接続されたホストから異なるアクセスルータに接続された別のホストへ送られる場合、最適な転送パスは、境界ルータを通過するパケットを必要としない場合が多い。C P P用の1つのパフォーマンス尺度は、C P Pが用いられない場合に存在する最適な転送パスが、どの程度、C P Pを用いてなお使用されているかということである。上述したアルゴリズムの場合、入ってくるパケットは、常に、境界ルータのうちの1つを通過する。

【0089】

本発明においては、(1)境界ルータ上のトラヒック負荷を低減すること、及び、(2)C P Pで保護された宛先アドレスを有するパケットが通過するルータの数を低減することが好ましい。上述のロケーションプライバシーが保護されたI Pアドレスが、「 $A = (P_0, Y, X, M)$ 」であることを思い起こす。ここで、「 Y 」は、イントラドメイン通信を最適化するのに利用可能である。「 Y 」は、およそ16~36ビットになると予想される(「 P_0 」は、32又は48ビットのいずれかであり、「 X 」は、およそ16ビットである。「 X 」は、正規化が行われる場合、16ビットよりも少し長くすることができることに注意する。また、「 Y 」も、ツリー上に多数のルートがある場合には、少し長くすることができる。従って、「 M 」は、およそ18~24ビットである)。

【0090】

本発明の様々な実施形態による最適化されたルーティング用の例は、基本的な例、高度化された基本的な例、最適化された例、最適化された例の特別なサブケース、及び最適化された例の鍵ハッシュ変形例を含む。以下、これらについて論じる。

【0091】

上述の基本的な例においては、「 Y 」は、ホスト用のアクセスルータと、当該アクセスルータの親ルータとの間の共用鍵で暗号化されたI Pアドレスのプレフィクスによって構成されている。ここで、アクセスルータは、(通常、発信元アドレスのプレフィクスを解読して調べることにより、)「イングレスフィルタリング」を行う。親ルータは、宛先アドレスを解読して、当該親ルータの子アクセスルータのうちの1つへ転送すべきか否かについてチェックすることにより、当該宛先アドレスをチェックして、(そのアクセスルータが、共通の親ルータを共用するホスト用に)限定された最適化ルーティングを可能にする。

【0092】

上述の高度化された基本的な例においては、「 Y 」は、アクセスルータの2レベル上にあるルータと、当該ルータの子との間で共用される鍵で暗号化されたI Pアドレスのプレフィクスによって構成されている。ここで、アクセスルータの親は、アクセスルータのフ

フィルタを直接用いて、或いは、（より粗い）統合フィルタを用いて、「イングレスフィルタリング」を行う。これらのルータの両方は、それらのサブツリー内のルーティングを最適化することができる。上述の高度化された基本的な例に関しては、親又は祖父のいずれかの障害は、祖父の下全てのホストに対するロケーションプライバシーの潜在的な損失をもたらすため、安全性は低下する。

【0093】

上述の基本的な例及び高度化された基本的な例の利点は、「Y」が16ビットを要するのみなので、2つの例とも実装が容易であることである。加えて、以下にさらに説明する完全に最適化されたルーティングの例は、より大きな解読演算を必要とする。

【0094】

上述の基本的な例及び高度化された基本的な例に関して、本発明者等が注目した潜在的な問題は、間違った暗号鍵が用いられた場合で、かつ、解読されたプレフィクスが、偶然に、上述の転送テーブル内のプレフィクスに一致した場合に、パケットが、間違ったルートで転送されかねないことである。このような可能性は、転送テーブル内のプレフィクスの数に依存し、プレフィクスは、単に16ビットであるため、この可能性は、いくつかのネットワークにとってはかなり高くなり得る。

【0095】

図10は、最適化されたルーティングの例を示す。本実施形態において、内部のホストから送られたパケットは、宛先アドレスに対する最適なパスの一部であるルータに到達するまで、ルーティンググラフの上方に転送される。このルータは、「クロスオーバールータ」と呼ばれている。パケットは、このポイントで下方へルーティングされるべきである。クロスオーバールータRは、「Y」成分のセグメント（実際には、「 $W(j, k) = U(j, k) \oplus H(N_j, M, X_j)$ 」）に対して、（上述したように、当該クロスオーバールータRのレベル鍵を用いて、）標準的なXORを実行して、ローカルシークレット「 $U(j, k)$ 」を得ることにより、このパケット用のクロスオーバールータであることを識別する。結果が、当該シークレットと一致する場合、Rは、それがクロスオーバールータであることを知る。

【0096】

その後、クロスオーバールータRは、「Z」の最初の部分（ここで、「 $Y = (W, Z)$ 」）を解読して、最初のプレフィクス部分を得る。次いで、Rは、インバウンド転送用の上述のアルゴリズムを用いて、パケットを転送する。図10は、上述のプライバシールーティングドメイン内の90に接続されたホストによって、同じドメイン内の92に接続された他のホストへ送信されるパケットPkt用の最適な転送パスを示す。パケットPktは、94を含むいくつかのルータにより、（境界ルータ98へ向けて）デフォルトルートの上方へ転送される。ルータ96は、「 $W(3, k_3)$ 」及び「 $U(3, k_3)$ 」を調べることで、ルータ96より下のサブツリー内のホストに、Pktの宛先が指定されていることを判断した後、Pktをシークレット「 $U(4, k_4)$ 」と共に、当該ルータへ転送する。従って、96は、パケットPkt用のクロスオーバールータである。その後、Pktは、92に到達するまで、連続するルータにより、ツリーの下方へ転送される。

【0097】

（安全性の低下という犠牲を払った）さらなる最適化として、クロスオーバーシークレットは、共用リンク（ルータグラフ内のクロスリンク）上で同じレベルの他の少数のルータと共用され得る。この共用によって、これらの隣接するルータが、パケットを迅速にクロスオーバールータへ転送することができる。

【0098】

上述したように、最適化されたルーティング用のC P Pアドレスの成分は、追加の必要条件を、I Pアドレス内の使用可能なビットに対して課す。図11は、使用可能なビットに対して課された、考えられ得る必要条件を含む例示的なI Pアドレス指定を示す。

【0099】

「M」に必要とされるビット数を低減することが好ましいと考える。以下の例は、「M

10

20

30

40

50

」が、所定のサブネット内でユニークであることを必要とするのみである。換言すれば、同じプレフィクスを有する2つのC P P I Pアドレスは、「M」に対して異なる値を有していなければならない、そうでない場合、2つのアドレスは、「M」に対して同じ値を有することができる。それに伴って、この以下のアルゴリズムは、「M」のサイズを約16～22ビットに低減すると予想される。

上述のように、プレフィクス成分「 $P_0, P_1, \dots, P_k$ 」を有するI Pアドレスを仮定すると、拡張されたC P P I Pアドレスは、「 $P_0, v, X_1, X_2, \dots, X_k, Y, M$ 」である。ここで、「 $X_k = H(K_k, M, P_1, \dots, P_{(k-1)}) \text{ XOR } P_k$ 」、 $X_{(k-1)} = H(K_{k-1}, M, X_k) \text{ XOR } P_{(k-1)}$ 」、 $\dots$ 、「 $X_1 = H(K_1, M, X_2) \text{ XOR } P_1$ 」となる。

10

【0100】

図11は、要素101を用いて、I P v 6アドレスのプレフィクス100からC P Pアドレス114への構造について示す。図11において、「 $Z = P_1 \text{ XOR } H(T_2, M, X_1) \mid P_2 \text{ XOR } H(T_3, M, X_2) \mid \dots \mid P_{k-1} \text{ XOR } H(T_k, M, X_{k-1})$ 」である。ここで、「 $T_i = \text{HMAC}(L_i, L_1 \mid L_2 \mid \dots \mid L_{(i-1)})$ 」、 $1 \leq i \leq k$ 」であり、例えば、「 L_1 」は、境界ルータによって配信された鍵であり、「 L_2 」は、次のホップルータ等により配信される。さらに、「 $W = H(B(1, k_1), X_1, M) \text{ XOR } V_1, H(B(2, k_2), X_2, M) \text{ XOR } V_2, \dots, H(B(j, k_j), X_j, M) \text{ XOR } V_j$ （完全に最適化されたスキームの鍵ハッシュ変形の場合）」であり、「 $V_1 = H(N_1, M, X_1)$ 」、 $\dots$ 、「 $V_{(k-1)} = H(N_{(k-1)}, M, X_{k-1})$ 」、 $V_k = H(N_k, M, X_k)$ 」である（又は、「 $W = U(1, k_1) \text{ XOR } V_1, U(2, k_2) \text{ XOR } V_2, \dots, U(j, k_j) \text{ XOR } V_j$ 」）（完全に最適化されたスキームの場合）である。

20

【0101】

図11において、要素100は、真の、即ち、暗号化されていないI P v 6アドレスを示す。グローバルルーティングプレフィクスは102である。要素104及び106は、「 X_1 」及び「 X_k 」を示す。上述の「 X_i 」の式は、異なるプレフィクス「P」を有するホストが、それらのC P Pアドレスにおいて同じ「M」の値を共用していても、当該ホストに関連する「X」を持たせない。

【0102】

30

「 $N_1, \dots, N_d$ 」を、レベル鍵の第2の集合とする。これらは、対の別個の秘密鍵である。当該レベルに属する全てのルータは、各レベル鍵を共用する。次のハッシュ式は、以下のように短縮される。

$$V_1 = H(N_1, M, X_1)$$

...

$$V_{(k-1)} = H(N_{(k-1)}, M, X_{k-1})$$

$$V_k = H(N_k, M, X_k)$$

【0103】

レベル $(d+1-j)$ のルータの番号は、「1」で始まる。クロスオーバーシークレット「 $U(j, k)$ 」は、レベル $(d+1-j)$ のk番目のルータ及び鍵サーバのみに知られているシークレットである。「(レベルjにおける) $U(j, k)$ 」は、対で別個であるため、必要なビット数は、レベル $(d+1-j)$ のルータの数の底2の対数に等しい。アクセスルータの真上のレベルで始めると、必要なビット数が確保される。「 $U(j, k)$ 」に必要な総ビット数は、「j」を越えるレベル $(d+1-j)$ の番号のルータの底2の対数の合計に等しい。典型的には、ルータの最大数は、アクセスルータのレベルの真上にある。

40

【0104】

インターネットからのパケットP用の最適なパスが、シークレット「 $U(1, k_1), U(2, k_2), \dots, U(j, k_j)$ 」を有するレベルルータを含む。そこで、I Pアドレスの「W」成分は、「 $W(i, k) = U(i, k) \text{ XOR } V_i$ 」（ここで、「M」は

50

、上述したサフィックスであり、「 V_i 」は、上記で定義した鍵であり、最適なパスは、レベル $(d+1-j)$ の k 番目のルータを含む)。IPアドレスは、「 $A = (P_0, X, Y, M)$ 」である。ここで、「 P_0 」、「 X 」及び「 M 」は、上述したようなものであり、「 $Y = (W, Z)$ 」であり、「 $W = U(1, k_1) \text{ XOR } V_1, U(2, k_2) \text{ XOR } V_2, \dots, U(j, k_j) \text{ XOR } V_j$ 」である。

【0105】

図11の要素108は、「 W 」を示す。鍵バージョンビット「 v 」は、要素116として示されている（例えば、ルーティングドメインが、ロケーションプライバシーが保護されたアドレス及びロケーションアドレスが保護されていないアドレス用の別々のグローバルルーティング識別子を得ている場合、必要ではないため、かかる1ビットは、示されていない）。レベル $(d+1-j)$ の k 番目のルータは、IPアドレス A と等しい宛先アドレスを有するパケットが一般に通過するルータである（当然、そうでない場合、この場合の最適化に影響を及ぼす可能性がある）。

10

【0106】

図11は、さらに、拡張されたCPP IPアドレスを示す。上述の「 W 」についての式において、境界ルータと、ホストの最初のホップルータとの間のレベルルータの数は、認識され得る（各レベルのクロスオーバーシークレットは、レベルにより固定長を有する）。スパニングツリーにおける境界ルータからアクセスルータへの全てのパスが等しい長さであれば、このことは問題ではない。さもなければ、下位のレベルに対するどのルータにも対応しない追加のクロスオーバーシークレットを用いることが好ましい。それらを「 W 」

20

$$「V_1 = H(N_1, M, X_1)」$$

...

$$「V_{(k-1)} = H(N_{(k-1)}, M, X_{k-1})」$$

$$「V_k = H(N_k, M, X_k)」$$

$$「V_{(k+1)} = H(N_{(k+1)}, M, X_1)」$$

...

$$「V_d = H(N_d, M, X_{(d \bmod k)})」$$

$$「W = U(1, k_1) \text{ XOR } V_1, U(2, k_2) \text{ XOR } V_2, \dots, U(d, k_d) \text{ XOR } V_d」$$

30

【0107】

全てのレベルルータは、全てのレベルに対するクロスオーバーシークレットの長さをもって構成されている。各レベルには、正確な1つの長さがある。別法として、「 U 」が、ルーティンググラフにおける少数のレベル用のルーティングを最適化することにのみ用いられる場合には、特別のシークレットに対応する特別の「 $W_{(j, k)}$ 」に加える必要はない。一実施形態において、特別の「 $W_{(j, k)}$ 」は、結果として生じるツリーが完全にバランスが取れるように、すなわち、境界ルータのルートからアクセスルータへの全ての経路が（「 $W_{(j, k)}$ 」によってモデル化されたスパニングツリー用の）同じ長さを有する。このアプローチの場合、ルータにとっては、当該ルータのレベルに対応する「 $W_{(j, k)}$ 」成分を見つけるのは簡単なことでもある、すなわち、当該ルータのレベルは、「 W 」成分中の同じ位置に常に現れる。

40

【0108】

本発明におけるIPアドレスの「 Y 」成分が、「 Z 」成分を含んでもよいことを思い起こしてほしい。「 Z 」成分を伴う実施形態の説明を続ける（「 Z 」は、図11において、要素110として示されている）。各レベルルータは、サブツリー内の当該レベルルータ配下の全ルータとの間で、秘密鍵 L を共用する。ここで、当該サブツリーは、当該共用リンク上のルータをルートとしている。元のスパニングツリーにおいて、レベルルータは、子ルータの集合を有する。レベルルータは、各子ルータをルートとするサブツリーの各々との間で、別々の秘密鍵を共用する。成分「 Z 」は、14～16ビットであり、上述したようにプレフィクス成分に分割されるサブネットプレフィクスからなる。この場合、各成

50

分は、秘密鍵「T」を用いて暗号化される。

【0109】

各秘密鍵「T」は、現在のレベルルータ及び当該現在のレベルルータの上方にあるレベルルータからの秘密鍵「L」から得られる。そのため、所定のレベルのルータRは、より高いレベルに対応する「Z」におけるプレフィクス成分を解読することにより、所定のレベルのルータRの上方にあるレベルルータからプレフィクスを得ることができ、各レベルルータは、当該レベルルータ自体の直接上のレベルからの全ての鍵「L」を有する。しかし、このことは、Rが、パケット用の最適なパス上にある場合にのみ可能であり、それによってセキュリティが維持される（「Z」成分は、ツリー内のいくつかの他のルータに、IPアドレスのロケーションに関する情報を漏らしてもよいことに注意する）。

10

【0110】

プレフィクス成分「 P_i 」は、鍵「 $T_i = \text{HMAC}(L_i, L_1 | L_2 | \dots | L_{(i-1)}) : Z_i = P_i \text{ XOR } H(T_{(i+1)}, M, X_i), (2 \leq i \leq k)$ 」を用いて暗号化される。ここで、例えば、「 L_1 」は、境界ルータによって配信される鍵であり、「 L_2 」は、次のホップルータによって配信される。「HMAC」の式において、「HMAC秘密鍵」として用いられる「 L_i 」及び「 $L_1 | L_2 | \dots | L_{(i-1)}$ 」は、「テキスト」である。

【0111】

「 K_i 」に対しては、他の構造が考えられ得る。「TLS RFC」や様々な暗号の論文は、いくつかの鍵を1つの鍵に結合する方法を記述している。

20

「 $Z = P_1 \text{ XOR } H(T_2, M, X_1) | P_2 \text{ XOR } H(T_3, M, X_2) | \dots | P_{k-1} \text{ XOR } H(T_k, M, X_{k-1})$ 」

レベル（ $d-k+1$ ）のルータが、「X」から「 P_k 」を得ることができるため、「Z」の最後の成分は必要ではない。

【0112】

ルータは、一般に、次の方法で、入ってくるパケットを転送する。第1に、当該ルータは、宛先アドレスの「 P_0 」プレフィクスビットをチェックして、当該パケットが、内部アドレス向けのものであるか否かについて判断する。内部アドレス向けのものでない場合、当該ルータは、当該パケットを、デフォルトルートを用いて、トップレベルのルータへ向けてルーティングする。内部アドレス向けのものである場合、及び、CPPホップバイホップオプションがない場合には、当該ルータは、当該ルータのレベルに従って、「W」成分をチェックする。この情報は、他の転送テーブル情報と共に予め構成される。当該ルータは、クロスオーバールータではない場合、当該パケットを、デフォルトルートを用いて、境界ルータへ向けて転送する。

30

【0113】

しかしながら、ルータRが、当該パケット用のクロスオーバールータであると仮定する。このとき、Rは、（境界ルータに対応する）「Z」における最初のレベル成分で開始する。かかる成分は、「 $P_1 \text{ XOR } (T_2, M, X_1)$ 」である。Rは、クロスオーバールータであるため、「 L_1 」及び「 L_2 」に対する正確な値を有するので、鍵「 T_2 」を計算することができる（Rは、直前のインバウンドルーティングアルゴリズムが用いられていなければ、境界ルータではない）。そのため、Rは、「 P_1 」を得ることができる。同様に、Rは、上述のツリーにおいてR自体よりも高いレベルに対応する他のPCを得ることができる。最後に、Rは、インバウンドルーティングアルゴリズムを用いて、R自体のレベルに対するプレフィクス成分を得る。Rは、連結されたプレフィクスを取得して、当該プレフィクスに基づいてパケットを転送する。また、Rは、当該連結されたプレフィクスを含むホップバイホップオプションも含む。Rは、境界ルータに近いツリーにおけるレベルに対するプレフィクス成分を解読するので、他のルータに対する最大PC長、及び、（他のルータに対して）考えられ得るPCを用いて構成される必要がある（また、正規化を用いる場合、Rは、他のルータ用の正規化マップも有することが必要となる）。

40

【0114】

50

従って、このアルゴリズムを用いた場合、上述の最大PC長の全てのストリングが、PC自体である場合には、構成の量を低減することが有利である。理想的には、全てのPCも、所定のルータに対して、最大PC長を有する。所定のレベルにおける全てのルータも同じ最大PC長を有している場合、このことは、必要な構成の量をさらに低減することになる。例えば、レベル1のルータRは、次のようにして、「 $U(1, k_1) \oplus V_1$ 」をチェックする。

- 「 V_1 」及び「 $B = W(1, k_1) \oplus V_1$ 」を計算する。
- そして、ターゲットホストがRより下のアクセスルータに属していれば、「B」は「 $U(1, k_1)$ 」である。

【0115】

10

統合を行わないルータは、入ってくるパケットを、デフォルトルートを用いて、ルーティンググラフの上方へ転送する。（レベル鍵を有する）レベルルータだけが、内部IPアドレスに宛先が指定された最適パケットをルーティングする。

【0116】

図1に関連して論じたような次の例について考える。サブネットのプレフィクス符号化「X」に対して2ビットが要求される。2ビットは、「Z」に対して必要となる。1ビットは、「W」に必要である。インターネットから、R5に接続されたホストへ送信されたパケットは、R1、R2及びR5を通過することになる。R2に属するシークレット「 $U(1, 2)$ 」が、1ビットからなるものとする。そして、R3に属するシークレット「 $U(1, 3)$ 」が反対の値を有するものとする（R2は、この特別なケースにおいて、ルータR3のシークレットの値を導出するが、IPアドレスのロケーションに関する追加の情報は与えないことに注意する。換言すれば、かかるIPアドレスは、R2の配下にない場合、R3の配下になければならない）。従って、この場合、「W」は、単一のビットを有する単一の成分からなる。

20

【0117】

R4からR5へ送信されたパケットは、R2へ転送されることになる。そして、R2は、「Y」をチェックして、R2のシークレットとの一致結果を得る。一致結果を得た後、R2は、当該パケットは、上述のグラフのより下位のレベルへ転送されるべきであることを知る。R2は、宛先アドレス内の「Z」における2つの成分を解釈して、サブネットプレフィクス「01（ $P_1 = 0$, $P_2 = 1$ ）」を得て、このサブネットプレフィクスに基づいて当該パケットをR5へルーティングする。

30

【0118】

本発明の例示的な最適化されたシステムにおけるIPアドレス成分の例示的なサイズは、次の通りである。IPアドレスAは、フォーム「 $A = (P_0, v, X, W, Z, M)$ 」からなる。

【0119】

「W」に必要なビット数は、「X」に必要なビット数よりも多い。これは、dレベルを有するバイナリツリーであるルーティンググラフのケースでは、「W」に必要なビット数は、各レベルにおけるビット数の対数の合計であることから明らかである。従って、

【数 2】

$$\begin{aligned}
 & d-1 \quad d-1 \\
 & (\\
 & |W| = \sum_{j=1}^{d-1} \log_2(d-j) = \sum_{j=1}^{d-1} (d-j) = d(d-1)/2 \\
 & 2 \\
 & j=1 \quad j=1.
 \end{aligned}$$

10

別の表現では、

【数 3】

$$|W| = \sum_{j=1}^{d-1} \log_2(d-j) = \sum_{j=1}^{d-1} (d-j) = d(d-1)/2$$

20

【0 1 2 0】

限定された数のレベル（4－5）の場合、必要な追加ビット数は、それほど多くはない。例えば、レベル 4 に 1 個のルータ、レベル 3 に 8 個のルータ、レベル 2 に 6 4 個のルータ、レベル 1 に 1 2 8 個のルータ及び 5 1 2 個のアクセスルータが存在するルーティングツリーの場合、「W」は、1 6 ビットを必要とするのに対して、「X」は、1 2 ビットを必要とする。「W」のビット数が、（「M」のサイズによって決まる）必要とされる IP アドレスの数により限定されなければならない場合、1 つの選択肢は、上述のグラフ中の最下位のレベルに対してのみ、ルート最適化を用いることである。

【0 1 2 1】

上述したように、「X」は、1 6 ～ 1 8 ビットを有する（正規化を用いる場合、X は、1 6 ビットよりも長い）。「W」は、クロスオーバーシークレットを含む。「W」に対しては、2 6 ビットで十分であると仮定する。1 ビットは、鍵のバージョンを表すのに用いられる。1 ビットは、（独立したグローバルプレフィクス「P₀」が、ロケーションプライバシーアドレスを識別するのに用いられる場合に不要とされ得る）ロケーションプライバシーバージョンを表すのに用いられ得る。「P₀」は、4 8 ビットを要してもよい。「Z」が 1 4 ビットを含むと仮定する。従って、「M」に割り当てるために、アドレス割当サーバには、2 0 ビットが残る。同じプレフィクスを有する 2 つの異なる IP アドレスは、同じ「M」成分を有するべきではない。

30

【0 1 2 2】

完全に最適化されたルーティングの例に関する特別なサブケースは、いくつかのアクセスネットワークに対して、改善されたパフォーマンス及びセキュリティ特性をもたらす。本発明を利用するネットワークは、可能であれば、この特別なサブケースを利用できるようにするように構成することが推奨される。この例においては、「Y=W」である。換言すれば、「Z」成分は、IP アドレス中には存在しない。そのため、（「Z」が用いられないため、）「Z」に関連する約 1 4 ビットが節約される。上述したように、「Z」は、IP アドレスのロケーションに関するいくつかの情報を（プライバシードメイン内のルータへ）漏らすかもしれない。従って、「Z」を取り除くことも、セキュリティの利益をもたらす。

40

【0 1 2 3】

「Y=W」である特別なサブケースにおいて、ルータは、プライバシードメイン内のア

50

ドレスに宛先が指定されたパケットを転送する場合、「W」を用いて、当該ルータがクロスオーバルルータであるか否かについてチェックする。当該ルータは、クロスオーバルルータでない場合、（以下に説明するように、ホップバイホップオプションをつけた後、）当該パケットをデフォルトルートの上方向へ転送する。当該ルータは、クロスオーバルルータである場合には、当該ルータのレベルに応じて、「X」成分を解読する。クロスオーバルルータは、常に、当該クロスオーバルルータのレベルに従って、単一の「X」成分に基づいてパケットを転送することができるため、かかる特別なケースが発生する。

【0124】

ホップバイホップオプションは、解読を終了するビットを表すのに用いられる。各ルータは、「X」成分に関して、ホップバイホップオプションのオフセットフィールドを用いて、次のホップが、解読を終了させるビットを表す。また、他のPCのサフィックスであるPCはないので、ルータは、（最後のビットが終了ビットに等しい）最大サイズのPCを解読した後、当該ルータのPCのリストを探索して正確なPCを得ることができる。（本願明細書中において、以後、「ORホップバイホップオプション」と呼ぶ）ホップバイホップオプションは、「暗号化されたプレフィクス」フィールドを有せず、かつ、「ターゲットレベル」フィールドが、1バイトの「インバウンドルーティングインジケータ」フィールドと置換され、長さが4バイトであることを除いて、上述のCPPホップバイホップオプションと同じフォーマットを有する。上述の特別なサブケースについて、上述のグラフ内の全ルータが、レベルルータであり、また、ルータが、当該ルータのレイヤ2リンクピアの各々のレベルを維持すると仮定する。

【0125】

上述の最適化されたルーティング状態を用いる場合、ルータは、常に、「インバウンドルーティングインジケータ」が「0」に等しい場合、当該ルータ自体のレベルに「1」を加えたレベルを有するルータにパケットを転送し、「インバウンドルーティングインジケータ」が「1」に等しい場合、当該ルータ自体のレベルから「1」を引いたレベルを有するルータにパケットを転送する（図7は、ルータが、（パケットに「ORホップバイホップオプション」がない場合に対応する）最適化されたルーティング状態にどのように入るかについて示す）。また、「ORホップバイホップオプション」を有するパケットを受け取るルータは、最適化されたルーティング状態にある。

【0126】

上述の完全に最適化されたスキーム及び最適化されたルーティングに対する鍵ハッシュ変形スキームとは異なって、クロスオーバルルータが一旦決まると、上述のインバウンドルーティングアルゴリズムは用いられない。これは、パケットが、宛先のアクセスルータに到達するまで、各処理ルータによって、当該パケット内で、「ORホップバイホップオプション」が、新たな「ORホップバイホップオプション」に置換され続けられるためである。違いは、クロスオーバルルータが、自身がクロスオーバルルータであると判断したときに従う「ORホップバイホップオプション」において、「1」に等しい「インバウンドルーティングインジケータ（IRI）」を設定することである。後続の全てのルータも、「IRI」を「1」に設定する。重要なことは、「IRI」フィールドが、「1」に等しい場合、解読用の終了ビットが、解読用の開始ビットになることである。

上記クロスオーバルルータの前に、上記IRIが0に設定される。ルータがクロスオーバルルータではない場合でも、ルータは、ホップバイホップオプションにオフセットフィールドを含んでいなければならない。そのため、ルータは、次のレベルのルータのためのエンディング解読ビットを得るために、当該ルータのレベルに対する対応するX成分を解読しなければならない。

【0127】

上述のクロスオーバルルータの前に、「IRI」が「0」に設定される。ルータが、クロスオーバルルータではない場合であっても、当該ルータは、ホップバイホップオプションに「オフセット」フィールドを含んでいなければならない。そのため、ルータは、次のレベルルータ用の終了解読ビットを得るために、当該ルータのレベルに対して対応する「X」

成分を解釈しなければならない。

【0128】

図12は、最適なルーティングの追加の実施形態を示す。本実施形態において、各ルータは、当該ルータ及びアドレス生成サーバ（アドレス生成については、図14に関連して以下に論じる）にのみ知られている固有鍵「K_r」を有して構成される。

【0129】

（図12において、）最適なルーティングを示す追加の実施形態において、内部ホストから送られたパケットは、宛先アドレス用の最適なパスの一部であるルータに到達するまで、ルーティンググラフの上方へ転送される。かかるルータは、クロスオーバルルータR（要素7b96として示す）と呼ばれる。パケットは、このポイントで下方へルーティングされる。

10

【0130】

クロスオーバルルータRは、パケットのサフィックス「M」、又は、当該サフィックスの一部に対して標準的なハッシュ処理を施して、特定のビットを切り捨てることにより、自身が当該パケット用のクロスオーバルルータであることを識別する。そして、当該ルータは、ハッシュ処理の結果と、「0」等の所定の値とを比較する。別法として、アドレス内の特定のビットは、所定値を含むリストに対するインデックス等の特定の値へのポインタ用に確保されていてもよい。かかる特定の値と、ハッシュ及び切り捨てる結果とが同じである場合、当該ルータは、自身がクロスオーバルルータであることを知る。ハッシュ及び切り捨てる例は、以下の通りである。

20

【数4】

```
RESULT = TRUNCATE (HASH (Kr, M))
VALUE = (pre-specified value) OR (table [index-contained-in-the-address])
If RESULT == VALUE, then cross-over router.
```

【0131】

上述のケースが「真」である場合、ルータは、単に、図13に示すわずかな追加を伴って、図10に関連して説明した最適化されたルーティングプロセスに従う。

30

【0132】

図13は、図12に関連して最初に説明したような最適なルーティングの変形例について更に説明するフローチャートである。初めからツリー内の上方へ向けられている任意のパケットは、予め解釈された（すなわち、より高いレベルのルータによって予め解釈された）プレフィックスを示すホップバイホップオプションを含むことはない。従って、このことは、クロスオーバルルータを、解釈されたより高いレベルのプレフィックスを有しない場合の境界ルータと同様の状況にする。

【0133】

ルータは、当該ルータのレベル鍵を用いて、当該ルータのレベルに応じて、解釈したプレフィックス成分を得て、解釈されたプレフィックス成分を、予め構成されたより高いレベルのプレフィックスに連結する。このより高いレベルのプレフィックスは、全てのより高いレベルの解釈されたプレフィックスの連結である。この連結は、ルータが、パケットが転送されなければならない次のより低いレベルのルータを見つけ出せるように、クロスオーバルルータに対して、当該クロスオーバルルータのルーティングテーブル内で「最長プレフィックスマッチ」を実行するための十分長いプレフィックスを与える。次のホップルータが一旦決定されると、当該クロスオーバルルータは、以下のうちの1つを用いてもよい。

40

【0134】

ケース1：クロスオーバルルータは、パケットを、（図12に示す7b100のような）

50

次のホップルータへ転送し、当該次のホップルータのホップバイホップヘッダに、新たに形成された連結プレフィクスを含ませる。ここで、ホップバイホップヘッダは、次のホップルータと共用する鍵を用いて状況に応じて暗号化されてもよい。この場合、次のホップルータは、ホップバイホップオプションを含む他のパケットのように、このパケットを処理することができる。

【0135】

ケース2：クロスオーバールータは、パケットを次のホップルータへ転送し、ホップバイホップヘッダに、新たに形成された連結プレフィクスを含ませない。この場合、受信ルータは、再び、上述したハッシュ及び切り捨て演算を実行して、当該受信ルータがクロスオーバールータであるか否かについて同様に判断する。この新たなタイプのアドレス指定を用いて、当該ルータは、（当該ルータのレベルに対応するプレフィクス成分を解釈し、より高いレベルのプレフィクスの解釈された連結と連結することにより）プレフィクスを再生し、当該ルータのルーティングテーブルにおける「最長プレフィクスマッチ」を行い、次のホップルータを見つけ出すことができる（ケース2は、図12には示していないことに注意する。また、ルータは、アクセスルータの真上のレベルにあるルータである場合には、ケース2を選んではいけない）。

【0136】

パケットを正確にルーティングすることができる図13の例示的な実施形態のルータについては、宛先アドレスは、次の条件に従うことになる。

【0137】

アクセスルータARへルーティングされるアドレスのサフィックス「M」は、アクセスルータARと境界ルータBRとの間の所望のパス又は最適なパス上の全ルータR用のものであり、

【数5】

$$\begin{aligned} \text{RESULT} &= \text{TRUNCATE}-(\text{HASH}(\text{Kr}, \text{M}), \text{L}) \text{ for all } r \text{ in } R \\ \text{VALUE} &= (\text{pre-specified value}) \text{ OR } (\text{Table-L} [\text{index-contained-in-the-} \\ &\quad \text{address}]) \\ \text{RESULT} &\text{ equals VALUE if and only if } r \text{ in } R. \end{aligned}$$

である。

【0138】

ここで、「L」は、ルータ「r」のレベルである。「TRUNCATE (NUMBER, L)」は、「NUMBER」を、レベル「L」に対応する所望のビット数に切り捨てる。一般的には、「L」に対応するビット数は、レベル「L」におけるルータの数の $\log_2$ について切り上げて整数化された数である。「Table-L」で示される独立したテーブルが、各レベル「L」に対して維持されていてもよい。「K_r」は、ルータ「r」の固有鍵である。この鍵は、一般に、上述したレベル鍵やリンク鍵とは異なる。これらの鍵は、他の鍵のようにバージョン化されていてもよい。本出願の他の箇所でも説明したような「鍵のバージョン化方法」を適用してもよい。

【0139】

次に、例えば、図12に関連して上述したようなネットワークに有用なアドレス生成サーバについて考える。サフィックス「M」のアドレスは、以下の事象が「真」である場合、アクセスルータ7b92に属することができる。

【数 6】

```

TRUNCATE-( (HASH (Key7b96, M), 1) equals Value1
And
TRUNCATE-( (HASH (Key7b100, M), 2) equals Value2
And
TRUNCATE-( (HASH (Key7b102, M), 3) equals Value3
And
TRUNCATE-( (HASH (Key7b104, M), 2) NOT equals Value2
And
TRUNCATE-( (HASH (Key7b94, M), 3) NOT equals Value3

```

10

【0140】

ここで、「Value1」、「Value2」及び「Value3」は、それぞれ、「7b96」がレベル1に属し、「7b104」及び「7b100」がレベル2に属し、「7b94」及び「7b102」がレベル3に属すると仮定した場合のレベル1、レベル2及びレベル3における所望の値である。「Key7b94」、「Key7b96」、「Key7b100」、「Key7b102」及び「Key7b104」は、それぞれ、ルータ7b94、7b96、7b100、7b102及び7b104の固有秘密鍵である。

【0141】

20

「M」についての直前の状況を考慮し、かつ、アドレス生成サーバが、ネットワークポロジ及び当該ネットワークにおけるルータの各々の固有秘密鍵を認識していると仮定すると、当該アドレス生成サーバは、各アクセスルータ用のクロスオーバールータを決めることができ、当該クロスオーバールータと当該アクセスルータとの間の対応するパスを見つけ出すことができ、かつ、上述したように、「M」の集合を計算することができる。次いで、当該アドレス生成サーバは、かかる「M」集合を、暗号化したプレフィクスと「P₀」との連結に連結して、完全なアドレスを形成する。そして、これらのアドレスは、当該アクセスルータの1つ上のレベルのルータに格納される（或いは、別法として、これらのアドレスは、当該アクセスルータに格納されてもよい）。

【0142】

30

図14は、アドレス生成サーバによるアドレス生成の例示的なプロセスを示す。当該アドレスサーバは、（機能2236及び機能2202に示すように、）適切なプレフィクスを見つけ出すことができる前に、アドレス空間集合「S」及びルータRの集合（及び、関連するルータポロジ情報）を知らなければならない。一旦、前の情報が分かると、当該アドレスサーバは、（機能2204に示すように、）一度に、集合「S」から1つの「M」を取り出して、取り出した「M」が、ルータのいずれかにとって適切であるか否かについて判断する。

【0143】

機能2208において、当該アドレス生成サーバは、上述の集合からルータ「r」を取り出して、「r」用のクロスオーバールータの集合を見つけ出す。かかる集合を、集合「P」と呼ぶ。さらに、「r」用のクロスオーバールータとは考えられないルータの集合「Q」は、機能2212において見つけ出される。

40

【0144】

多値変数が用いられるか否かにより、考えられ得る値「v」の集合「V」が、機能2216において計算される。多値変数が用いられない場合、かかる集合は、ちょうど1つの要素を有することになる。さらに、機能2216において、アドレス生成サーバは、一度に、1つの値「v」を取り出して、機能2218、2220及び2226に示すように、「P」における全てのルータに対して、「Truncate(Hash(K_p, M))」が「v」に等しいか否かについて判断する。「P」における全てのルータが、この条件を満たす場合、当該アドレスサーバは、機能2228、2230、2232及び2234に

50

示すように、集合「Q」における各ルータ「q」に対して、「Truncate (Hash (K_p, M))」が「v」に等しくないことを確かめる。また、この条件も満たされる場合、機能 2 2 3 4 及び 2 2 5 0 において、(v, M) 対も、ルータ「R」用の有効なサフィックスとして加えられる。これらのこれまでのステップは、機能 2 2 4 8 及び 2 2 4 6 における「v」の他の考えられ得る値に対して繰り返される。

【0 1 4 5】

機能 2 2 2 4 において、「P」におけるいずれかのルータが、テストに失敗した場合、(v, M) 対を、「r」に対して用いることができない。そして、機能 2 2 4 4 において、「V」の他の全ての値が、全てが試されるまで繰り返される。次に、機能 2 2 5 2 において、次のルータが、「R」における全てのルータが探索されるまで、評価される。所定の「M」に対して、全てのルータが探索された場合、次の「M」が評価される（機能 2 2 5 2、2 2 3 6 及び 2 2 4 0）。機能 2 2 3 2 において「Q」におけるルータ「q」が上述の条件に当てはまらない状況と同様に（かかる場合、(v, M) 対を「r」に対して使用できない）、機能 2 2 4 6 において、ルータ「r」用の「M」を廃棄し、機能 2 2 4 4 及び 2 2 5 2 において、「R」における他のルータに対してテストする前に、集合「V」における残りの値を通じて繰り返す。

【0 1 4 6】

図 1 5 及び図 1 6 は、追加の最適なルーティング、例えば、図 1 2 に関連して説明したルーティング用のアドレスの 2 つの考えられ得る構造を示す。

【0 1 4 7】

図 1 5 において、かかるアドレスは、プライバシードメインプレフィクス 1 6 1 0 と、当該 IP アドレスに対応する鍵のバージョンを表すビット 1 6 1 2 と、暗号化されたプレフィクスの連結 1 6 1 4 と、サフィックス「M」1 6 1 6 とによって構成されている。当該アドレスは、これまでに既に決まっているテーブル内の特定のエントリに対するインデックス等の特定の値へのどのようなポインタも含まない。値がない場合、全てのルータ（又は、少なくとも所定のレベルにおける全てのルータ）は、それまでに合意されている特定の値を使用しなければならない。IPv6 の場合、「M」の長さは、一般的には、6 2 ビットであるが、ネットワークに適した他の値が用いられてもよい。

【0 1 4 8】

図 1 6 は、（要素 1 7 2 1 によって）IP アドレス内の値 1 7 1 8 により示されるインデックステーブル 1 7 2 0 を示す。当該アドレスは、プライバシードメインプレフィクス 1 7 1 0 と、当該 IP アドレスに対応する鍵のバージョンを表すビット 1 7 1 2 と、暗号化されたプレフィクスの連結 1 7 1 4 と、サフィックス「M」1 7 1 6 とによって構成されている。また、当該アドレスは、既に合意されているテーブル内のエントリ、各レベルにおけるルータ用に構成されているテーブル内のエントリ、或いは、特定の値自体に対するインデックス等の特定の値 1 7 2 0 に対するポインタ 1 7 1 8 も含む。ポインタ又は特定の値 1 7 1 8 を有することは、アドレス生成サーバの効率を改善するのに好ましい。ポインタ 1 7 1 8 を用いて、ルータの所定のトポロジーに必要なアドレスを見つけ出すためにアドレスサーバが行わなければならない計算の量は、平均して「2ⁿ 倍」だけ低減される。ここで、「n」は、値のビット数又は値に対するポインタのビット数である。1 7 1 8 がある場合、「M」の長さは、IPv6 が用いられている場合、一般的には、(6 2 - m) ビットに低減される。しかし、当業者は、他の値も使用できることを理解している。

【0 1 4 9】

図 1 7 は、最適化されたルーティング 1 9 4 用の特別なサブケースを示す。ルータが、受信したパケットに対してインバウンドルーティング状態である場合、機能 2 1 6 にいる。OR タイプのホップバイホップオプションを用いずにパケットを受信し、インバウンドルーティング状態に達したとき、当該ルータは、機能 2 1 2 に示すようになる。当該ルータは、機能 2 1 8 に示す「クロスオーバーチェック」を行う。機能 2 1 8 は、当該ルータが、次に示すように、クロスオーバー状態にあるか否かについて判断する。

- 自身のレベルが「j」であるの場合、「W(j, l)」を見つける。

- 「 $W(j, 1)$ 」用の開始ビットは、予め構成されている。
- 「 $V_j = H(N(j), M, X_j)$ 」を計算する。
- 「 $H(B(j, 1), X_j, M) = W(j, 1) \text{ XOR } V_j$ 」であるか。(ルータが、クロスオーバールータである場合のみ「Yes」)。

【0150】

次に、機能212において、「クロスオーバーチェック」を行った後、当該ルータは、右端のビットで始まる「X」を解読し、その解読されたプレフィクス成分(PC)用の終了ビットを得る。当該ルータが、クロスオーバールータである場合、当該ルータは、(PCを「FWalg*()」に入力して次のホップルータを得ることにより、)解読されたPCに基づいて、パケットを転送する。当該ルータが、クロスオーバールータでない場合、当該ルータは、終了ビットを含むORホップバイホップ及び「IRI=0」で、パケットをデフォルトルートの上方向へ転送する。「IRI=1」の場合、「E/Sビット」が開始ビットであり、当該ルータは、開始ビットで解読を開始し、右へ進行する。一方、「IRI=0」の場合には、「E/Sビット」は終了ビットであり、当該ルータは、終了ビットで解読を開始して、左へ進行する。この後者の場合には、PCのサフィックスが、PC自身ではない特性(この特性は、以下で説明する)を必要とする。

10

【0151】

そうでない場合、当該ルータは、受信したパケットがORホップバイホップオプションを有する状態220でなければならない。当該ルータは、ORホップバイホップオプションを解読して、「終了又は開始(E/S)ビット」及び「IRI」を得る。当該ルータは、「E/Sビット」で開始する「X」を解読して、新たな「E/Sビット」(なお、新たなPCが、当該新たな「E/Sビット」で終了する)を得て、PCを解読する。次に、当該ルータは、ホップバイホップオプションからの「IRI」が「0」に等しいか否かについてチェックする。「IRI」が「0」に等しくない場合、(機能210のように、)「IRI=1」であり、当該ルータは、解読したPCに基づいて転送を行い、パケット内で、古いホップバイホップオプションに取って代わる新しいホップバイホップオプションに「IRI=1」を設定する。また、新しいORホップバイホップは、新しい「E/Sビット」を含む。

20

【0152】

「IRI=0」の場合には、状態214である。次いで、「クロスオーバーチェック」が行われる。当該ルータが、クロスオーバールータである場合、解読したPCに基づいて転送が行われ、「ORホップバイホップオプション」に新たな「E/Sビット」と共に「IRI=1」を設定する。当該ルータが、クロスオーバールータでない場合、「IRI=0」を保持し、ORホップバイホップ内の新たなE/Sを用いて、デフォルトルートの上方向への転送を行う。

30

【0153】

プレフィクスは、転送アルゴリズムにおいて、上方へ流れてはいけないことに注意する。プレフィクスがそうなった場合、より高いレベルのルータは、完全に解読されたプレフィクスを得ることができ、CPPのプライバシー特性が無効になる。

【0154】

上記の論考に基づくと、以下の問題は自明である。すなわち、ルータが、当該ルータのレベル鍵で暗号化されたプレフィクス成分に基づいて、インバウンドパケットを転送することができるかと仮定する。さらに、プレフィクス成分のサフィックスが、当該プレフィクス成分自身ではないと仮定する。従って、上述のアルゴリズムは、プライバシーードメイン内で最適なルーティングを提供する。

40

【0155】

以下の実施形態においては、セキュリティ問題及び適合した鍵ハッシュ変形について完全に最適化されたルーティングの例が提供される。

【0156】

「U」が、アドレスのロケーションに関する情報を漏らすことに注意する。それに伴っ

50

て、レベルルータのシークレットを所有する盗聴者は、十分なパケットを入手することにより、「U」を計算することができる。このことは、上述の「W」の式における「U」を置換することにより阻止することができる。

【0157】

例えば、「 $B(j, k)$ 」を、鍵サーバ及びレベル $(d+1-j)$ におけるレベルの k 番目のルータに知られている秘密鍵とする。インターネットからのパケットP用の最適パスが、次の秘密鍵（すなわち、「 $B(1, k_1)$ 」, $B(2, k_2)$ 」, ..., $B(j, k_j)$ 」）を有するレベルルータを含むと仮定する。すると、「W」は、「 $W = H(B(1, k_1), M) \oplus V_1, H(B(2, k_2), M) \oplus V_2, \dots, H(B(j, k_j), M) \oplus V_j$ 」となる。ここで、「 $H(B(j, k), M)$ 」は、「 k 」が「1」に等しくない場合、「 $H(B(j, 1), M)$ 」に等しくなく、「 $\text{Trunc}(H(B(i, k), M))$ 」用に選択されたビット数が、レベル $(d+1-i)$ におけるルータの数の2倍の対数の底に等しい。

【0158】

例えば、上述の例において、レベル1における128のルータの場合、14ビットが、「W」のレベル1成分に用いられる。ビット数を増やすことにより、潜在的な衝突の数も、単一のレベルにおけるハッシュ関数値の間で低減される。当業者は、バースデーパラドックスは、たとえビット数が増加した場合でも、当時の約40～50%、衝突が予想できることを示すことを認めるであろう。この問題を扱うために、上述の鍵サーバは、レベル $(d+1-i)$ における全てのルータに対する「 $\text{Trunc}(H(B(i, k), M))$ 」を計算することにより、各レベルルータに対して所定の「M」を検査しなければならない。「M」は、計算された値に衝突がない場合にのみ用いることができる。レベル $(d+1-i)$ をチェックした後、当該鍵サーバは、レベル $(d-1)$ が良好にチェックされるまで、レベル $(d+2-i)$ 等をチェックする。「M」は、最適化されたルーティング成分が「W」にあるレベルの衝突が全くない場合にのみ用いられる。

【0159】

各レベルにおいて、「M」の半分以下を失うことが予想される。そのため、4つのレベルの場合、「M」の15/16未満が失われる。「M」が、20ビットの場合、 2^{16} 超の考えられ得る「M」がある。このようなアプローチは、各成分のサイズを増加させる。従って、このアプローチを伴う最適化されたルーティングの1～3レベルに対しては、十分なビットがある可能性がある。状況に応じて、「W」成分は、「W」の成分が存在すること、或いは、「W」の成分が存在しないことを示す1ビット又は2ビットを前置きしてもよい。この新しいスキームを用いた別のアプローチは、レベル1のルータ用の最適化したルーティングを省くことである。この場合、レベル2及びレベル3にとって十分なビットであろう、及び、最低でも、レベル4において「 $U(j, k)$ 」に対して用いられるビットをいくつか残す。「U」は、上述のツリーにおけるより高いレベルにおいて、より少ない情報を漏らす。換言すれば、レベル2及びレベル3においては、「 $H(B(j, k), M)$ 」を用いることができ、レベル4においては、「 $U(j, k)$ 」を用いることができる。

【0160】

別法として、レベル2、3において、及び、ビットが残るより高い他のいかなるレベルにおいても、「 $H(B(j, k), M)$ 」のみを用いることができる。これらは、当業者にとって明白な追加の変形例である。

【0161】

レベル4に1個のルータを有し、レベル3に8個のルータを有し、レベル2に64個のルータを有し、レベル1に128個のルータを有し、512個のアクセスルータを有するルーティングツリーを含む上述の例について再び考える。「X」が、16ビットを使用し、「Z」が、14ビットを使用する一方、「 P_0 」が、48ビットを要するものと仮定する。「M」は、20ビットを有する。2ビットが、鍵バージョン用として、及び、ロケーションプライバシーが保護されたアドレスであることを表すために用いられる。そのため

、「W」は、28ビットを有する。レベル1のルータにおけるルート最適化が先行した場合、完全に最適化されたスキームの鍵ハッシュ変形は、レベル2に対して12ビットを使用し、6ビットがレベル3用に必要となる。そのため、「W」は、18ビットを必要とする。

【0162】

別法として、レベル1及びレベル2は、(鍵ハッシュ変形を用いて)最適化され得る。ここで、(14+12)ビット(合計で、26ビット)が使用される。「U」が、レベル3における最適化に用いられる場合、さらに3ビットが「W」に必要である。「W」は、28ビットを有し、26ビットがすでに使用されている。そのため、「M」から1ビット借りて、「M」を19ビットに低減する。「M」は、(衝突によるいくつかの「M」の排除により、)実際には16ビットよりも少し大きいだけである。従って、65,000超のIPアドレスが、サブネットごとに使用可能となる。

【0163】

上述の式においては、「 $H(B(i, k_i), M)$ 」を「 $H(B(i, k_i), X_i, M)$ 」と置換することができることに注意する(ここで、 $1 \leq i \leq j$ である)。この置換の利点は、 $H(B(i, k_i), M)$ を得る攻撃者が、この値と関連付けられたロケーション情報と、同じ「M」の値を共有するホストとを対応付けることを防ぐことである。換言すれば、同じ「M」の値を共有するホストは、「i」のいくつかの値に対して、同じ「 $H(B(i, k_i), M)$ 」を共有してもよい。

【0164】

図18は、本発明の鍵ハッシュルーティング変形例201を示す。図示の実施形態は、「 $H(B(i, k_i), X_i, M)$ 」を使用し、「鍵ハッシュ変形例」とは、この例を指す(ここで、図5の機能160は、鍵ハッシュが実施されることを表してもよい)。図18の機能206において、パケット用の処理ルータは、自身がレベルルータであるか否かについて判断する。レベルルータでない場合、当該ルータは、機能202においてデフォルトルートの上方にパケットを転送する。レベルルータである場合、当該ルータは、機能208を実行する。すなわち、

- 当該処理ルータのレベルが「j」の場合、「 $W(j, 1)$ 」を見つける。
- 「 $W(j, 1)$ 」の開始ビットは、予め構成されている。
- 「 $V_j = H(N(j), M, X_j)$ 」を計算する。
- 「 $H(B(j, 1)X_j, M) = W(j, 1) \text{ XOR } V_j$ 」であるか。

【0165】

当該パケット用の処理ルータが、機能208において、自身がレベルルータではないと判断した場合には、当該処理ルータは、(自身がクロスオーバールータではないため、)機能200に進んで、パケットを、デフォルトルートの上方へ転送する。次いで、当該処理ルータは、機能204へ進んで、次のステップを実行する(かかる場合、当該ルータは、当該パケット用のクロスオーバールータである)。

- 鍵「 $T_i = \text{HMAC}(L_i, L_1 | L_2 | \dots | L_{(i-1)}, (i=2, \dots, j))$ 」は、予め計算されている。
- 「 $P_{(i-1)} = H(T_i, M, X_{(i-1)}) \text{ XOR } P_{(i-1)} \text{ XOR } H(T_i, M, X_{(i-1)})$ 」を計算する。
- 「 P_i 」を連結して、プレフィクス「 P_c 」を取得する。
- レベル鍵を用いて、「j」、「 P_c 」、「 P_d 」を解読し、「 $P_{new} = P_c | P_d$ 」を取得する。
- 「 P_{new} 」を「FWalg*()」に入力することによって、「NextHop(次のホップのルータ)」を取得する。
- 開始ビット(「オフセット=(P_d の右端のビット)+1」)及び新たなターゲットレベル(「新たなターゲットレベル=自身のレベル-1」)と共に、共有リンク鍵を用いて、ホップバイホップオプション内に、「 P_{new} 」を暗号化する。
- パケットに対して当該ホップバイホップオプションを追加して、「NextHop」

10

20

30

40

50

に転送する。

【0166】

C P P は、ルーティングを最適化しているため、トラヒック集中ポイントが回避され、モバイル I P や H M I P v 6 に関連するようなトンネリングオーバーヘッドも回避される。

【0167】

図 19 は、本発明の一実施形態における例示的なアーキテクチャ構成要素を示す。ユーザホスト 144 は、アクセスルータと同じ場所に配置された D H C P v 6 サーバ 146 から、当該ユーザホスト 144 の I P アドレスを得る。この通信は、120 を介して行われる（ローカルリンク上の盗聴者に対する追加の保護の場合、I P s e c を用いることができる。120 を介した通信について暗号化すべきである。）

10

【0168】

要素 146 は、通信パス 122 を介して、126 に配置されたアドレスサーバから、C P P I P アドレス（基本 C P P I P アドレス又は拡張 C P P I P アドレスのいずれか）の集合を定期的に得る。ここでも、この通信について、（例えば、A A A (R a d i u s / D i a m e t e r) 又は I P s e c 等の）暗号化プロトコルを用いて保護することができる。A A A サーバは、要素 138 によって示される。

【0169】

いくつかの鍵の集合がある。基本 C P P I P アドレスにおいては、レベル鍵及び共用リンク鍵のみが必要となる。共用リンク鍵は、（例えば、I K E、ケルベロス、R a d i u s / D i a m e t e r 又は T L S 等の）任意の適当な鍵管理／配布プロトコルを用いて、レイヤ 2 リンクを介して設定される。当該プロトコルは、相互認証及びセッション鍵の設定を提供すべきである。拡張 C P P I P アドレスの場合、クロスオーバレベル鍵 (N_i)、クロスオーバーシークレット ($U(j, k)$ 及び／又は $B(j, k)$)、及び、ツリー鍵 (L_i) が、基本レベル (K_i) 鍵に付加される。これらの鍵は、まとめて「C P P 鍵」と呼ばれる。

20

【0170】

C P P 鍵は、耐タンパ型スマートカード内のマスター鍵生成器上で生成され得る。鍵サーバホストのスマートカードの各々の公開鍵は、マスター鍵生成器のスマートカードに手動で入れられていてもよい。当該スマートカードは、キーボードパッド及びコマンドラインディスプレイインタフェースを有してもよい。（公開鍵暗号化アルゴリズムにおける公開鍵に対応する）プライベート鍵及び秘密鍵は、スマートカードのどれからも出ない。この規則の例外は、上述の C P P 鍵が、スマートカードにのみ知られている鍵、或いは、ターゲットのスマートカードと共用する T L S セッション鍵にいずれかによって暗号化されて、当該スマートカードからホストに出力されることである。

30

【0171】

逆に、マスター鍵生成器 134 及びバックアップ鍵生成器 136 の公開鍵は、アドレスサーバ／鍵サーバホストのスマートカードに、手動で入れられる。この手動処理手続きは、スマートカードの寿命中に一度だけ行われる。後に、新たな公開鍵が生成された場合、かかる公開鍵は、ターゲットホストのスマートカードの T L S セッション鍵で暗号化されて、当該スマートカードから出力される。当該 T L S セッション鍵も、当該スマートカードに保持される。当該スマートカード用の最初の公開鍵は、適切なコマンドが入力されたときに、コマンドラインインタフェース上に表示される。

40

【0172】

本発明のアーキテクチャは、スマートカードに対する物理的攻撃が成功しない限り、T L S プロトコルが破壊されない限り、或いは、鍵が C P P I P アドレス又はプライバシードメインルータから得られない限り、秘密鍵及びプライベート鍵が敵対者／攻撃者に知られないことを保証する。T L S プロトコルは、上述のホストで実施される（鍵サーバホスト、及び、マスター鍵／バックアップ鍵生成器ホスト）。しかし、当該ホストは、暗号化及び鍵操作のために、スマートカードとのインタフェースである A P I を用いる。当該ホストは、どのプライベート鍵又は秘密鍵にもアクセスしない。また、当該スマートカー

50

ドは、T L S プロトコルの認証状況についての処理を行ってもよい。

【0173】

スマートカードが、互いに公開鍵を共用するので、認証は、全て自己承認される。そのため、認証動作は、名前検索と、及び、認証における公開鍵が、当該名前に関連付けられており当該スマートカードに記憶されているものと一致するかについてのチェックとに限定される。マスター鍵生成器及び鍵サーバは、共に、以下の要素、すなわち、「ルータ I D 1」、「クロスオーバーシークレット I D 1」、「(ツリー鍵 1 1、ルータ I D 1 1)」、「(ツリー鍵 1 2、ルータ I D 1 2)」、…、「レベルルータ I D 2」、「クロスオーバーシークレット I D 2」、「(ツリー鍵 2 1、第 1 のルータ I D 2 1)」、「(ツリー鍵 2 2、第 2 のルータ I D 2 2)」、…、レベル…、を含む鍵キャッシュを構築する。

10

【0174】

最初の列は、全てのレベルルータをリストする。当該ルータ用の I D は、D N S ネーム、D N S ネームのハッシュ、或いは、可能ならば I P アドレスのいずれかとすることができる。第 2 の列は、当該ルータ用のクロスオーバーシークレット (U (j , k)) を含む。次のエントリは、当該ルータが、当該ルータの子のルータに配布するツリー鍵である。最後の列は、当該ルータ用のレベルを含む。加えて、鍵キャッシュは、レベル鍵について二重にリンクされたリストと、クロスオーバレベル鍵について二重にリンクされたリストとを含む。

【0175】

これらのシークレット及び鍵は、マスター鍵生成器のスマートカードに生成される。マスター鍵生成器サーバは、このキャッシュをホスト上に構築する。当該シークレット及び鍵は、暗号化形式でキャッシュ内に存在する。当該シークレット又は鍵用の暗号化形式の例は、次の通りである。

20

[鍵バージョン、{ランダムな 8 バイト (コンファウンダー)、鍵の長さ、鍵、タイムスタンプ (4 バイト)} スマートカード鍵] スマートカード鍵 2

【0176】

「{…} K」という記述は、秘密鍵「K」を用いた括弧内に含まれるバイトの暗号化を表す。「[…] K 2」という記述は、「…」が、鍵「K 2」及び H M A C 等の鍵を用いた完全性関数を用いた完全性が保護されたバイトであることを表す。従って、ホストは、秘密鍵にアクセスしない。暗号化された秘密鍵を鍵サーバホストへ送信する前に、マスター鍵生成器サーバは、T L S プロトコルを用いて、暗号化された鍵を宛先ホスト名と共に、スマートカードへ送信する。

30

【0177】

当該スマートカードは、宛先ホストのスマートカードと共用する T L S 秘密セッション鍵 (チャンネル暗号化鍵) を用いて、鍵を再び暗号化する。次いで、当該マスター鍵生成器サーバは、鍵キャッシュデータを鍵サーバホストへ送信する。上述のフィールド内のタイムスタンプは、ホスト上の悪意のあるソフトウェアからのリプレイを検知するのに用いられる。当該スマートカードは、自身のタイムスタンプを維持する (すなわち、暗号化された鍵の全てのタイムスタンプ値は、当該タイムスタンプよりも後でなければならない)。鍵サーバホスト上の各鍵サーバは、同じ暗号化された鍵プロトコルを用いて、ローカル鍵キャッシュを構築する。このローカル鍵キャッシュは、マスター鍵サーバホスト上のもののコピーである。

40

【0178】

別法として、鍵キャッシュは、マスター鍵サーバホスト上の鍵キャッシュの部分集合とすることができる。鍵サーバホストスマートカードは、(このキャッシュが、マスター鍵サーバホスト上の鍵キャッシュのコピーであっても、マスター鍵サーバホスト上の鍵キャッシュの部分集合であっても、) アドレスサーバからの要求を処理する場合、鍵キャッシュにアクセスすることになる。

【0179】

次に、鍵サーバとマスター鍵生成器サーバとの間のプロトコルについて説明する。要求

50

メッセージに関して、以下に示す非排他的な例は、上述のアドレスサーバによって送信されてもよい。

- (1) クロスオーバーシークレット要求メッセージ *i d*、長さ、ルータ *I D*；
- (2) ツリー鍵要求メッセージ *i d*、長さ、親ルータ *I D*、子ルータ *I D*；
- (3) レベル鍵要求メッセージ *i d*、長さ、レベル；
- (4) クロスオーバーレベル鍵要求メッセージ *i d*、長さ、レベル；
- (5) 公開鍵更新要求メッセージ *i d*、長さ；
- (6) 最適化されたルーティング鍵更新メッセージ、メッセージ *i d*、ルータ *I D*

【0180】

直前の要求メッセージに対応する応答メッセージに関して、以下に示す非排他的な例が返されてもよい。

- (1) クロスオーバーシークレット応答メッセージ *i d*、長さ、ルータ *I D*、暗号化された鍵；
- (2) ツリー鍵応答メッセージ *i d*、長さ、親ルータ *I D*、子ルータ *I D*、暗号化された鍵；
- (3) レベル鍵応答メッセージ *i d*、長さ、レベル、暗号化された鍵；
- (4) クロスオーバーレベル鍵応答メッセージ *i d*、長さ、レベル、暗号化された鍵；
- (5) 公開鍵更新応答メッセージ *i d*、長さ、暗号化された鍵；
- (6) 最適化されたルーティング鍵応答メッセージ *i d*、長さ、暗号化された鍵及び他のパラメータ

【0181】

上述のプライバシーードメイン内のルータに関する情報を有するデータベースが存在する。典型的なデータベースのエントリは、「当該ルータのDNSネーム」、「他のルータネーム又は識別子」、「当該ルータにより所有されるプレフィクス」、「レベル」、「当該ルータの親」、「当該ルータの子」、「当該ルータの他のレイヤ2ピア」及び「直接接続されたリンク用のリンクコスト」を含む。C P P及び／又は拡張C P Pアドレスの構造も、ここに格納される。例えば、種々の成分「M」、「X」、「W」、「Z」に用いられるビット数が格納される。また、「W」のいずれかの成分が（最適化されたルーティングのために）失われた場合、かかる情報も同様に格納される。

【0182】

上述のデータベースエントリ内の情報は、当該データベース内で、及び、ネットワークを通過する際に、保護される。同様に、その他の情報があってもよい。各アドレスサーバは、当該データベースにアクセスして、ルータ及びアドレスに関する情報を構築する。当該アドレスサーバは、

「プレフィクス1→ルータの最適なルートリスト； $P_{11}, \dots, P_{1k}$ 」

「プレフィクス2→ルータの最適なルートリスト； $P_{21}, \dots, P_{2k}$ 」

の形式のアドレスサーバキャッシュを構築する。

【0183】

「 P_i 」は、プレフィクスに対応するプレフィクス成分である。最適なルータのリストは、プレフィクスについて、境界ルータから宛先アクセスルータへの最低のコストのパスを含む。また、アドレスサーバは、マスター鍵生成器サーバに対して、「M」のブロックを定期的に要求する。これらの「M」は、C P P I Pアドレスを構築するのに必要である。これらの「M」は、T L Sを用いて暗号化され、完全に保護される。「M」は、暗号化されていない形式では、上述のホストに対して利用可能ではない。

【0184】

アドレスサーバとスマートカードとの間のインタフェースは、次の通りである。当該アドレスサーバは、「 $P_1, \dots, P_k$ 」と、DNSネーム又は最適なルートのルータ用の他の識別子のリストとを含む要求を生成することができる。当該スマートカードは、かかる情報から、C P P I Pアドレスを生成して返すことになる。当該スマートカードは、C P P I Pアドレスを構築するために、ルータ鍵情報について、鍵サーバキャッシュに問

10

20

30

40

50

い合わせる。これらの問い合わせは、レベル鍵及びクロスオーバーレベル鍵を要求したときに、ルータID又はレベルのいずれかで索引付けされる。

【0185】

また、スマートカードは、「M」のブロックについて、アドレスサーバに問い合わせることもできる。これらの「M」は、当該スマートカードに配布されるまでは、暗号化されて完全に保護されたままである。信頼の置けないソフトウェアは、「M」を変更することができないことが重要である。完全に最適化されたルーティングの鍵ハッシュ変形を用いる場合、当該スマートカードは、レベル $(d+1-j)$ におけるルータに対応する各「k」に対して「 $\text{Trunc}(H(B(j, k), M))$ 」を計算することにより、各「M」をテストしなければならない。ここで、「Trunc」は、 $(d+1-j)$ ルータの数の2倍の対数に切り捨てることを示す。これらの計算値に衝突がなければ、「M」は、 $(d+1-j)$ レベルに対して許容できる。各連続するレベルも、テストされ、「M」は、全てのレベルにおいて衝突がない場合にのみ用いられる。

10

【0186】

図20は、完全に最適化されたルーティングの鍵ハッシュ変形が用いられる場合に、拡張C P P I Pアドレスを割り当てる例の機能を示す。(可能であれば、アドレスサーバホスト上のスマートカード内で実行する) 割り当てエンティティは、機能230で始まる。入力は、機能232にリストされており、アドレスプレフィクス成分「 $P_0, P_1, \dots, P_k$ 」と、(ネットワーク障害がない場合に、) 通常、パケットが通過するルータのリストとを含む(ネットワーク障害は、パケットが、生成されるアドレスへ配信されることを妨げない、或いは、アドレスから配信されることを妨げないが、いくつかのパケットは、ネットワーク障害がある場合、特別なホップを要求してもよいことに注意する)。

20

【0187】

機能234において、「M」が選択される。かかる機能は、(リストからのものである、或いは、任意に生成され得る) 次の「M」を得る。加えて、「M」は、既に「 $P_0, P_1, \dots, P_k$ 」に関連付けられているサブネットに対して用いられている場合には、捨てられる。「M」が捨てられた場合、機能234を再度行うことにより、別の「M」が選択される。機能236において、最適化ルーティングが用いられている各レベル「 $j = d-1, \dots, 2$ 」について、レベル $(d+1-j)$ の各ルータ「k」に対する「 $\text{Trunc}(H(B(j, k), X_j, M))$ 」を計算する。いずれか2つの値が一致した場合には、「M」を捨てる。「M」が、レベル $(d+1-j)$ に許容できる場合、「j」を減らして、テストを繰り返す。「M」が捨てられた場合、機能234に戻って、新たな「M」を選択する。そうでない場合には、機能238へ進む。

30

【0188】

機能238においては、拡張C P PアドレスAが生成され、機能234及び機能236におけるテストを通過した「M」がある。そこで、「 P_0 」をグローバルルーティングプレフィクスとして有する新たなI Pアドレスが生成される。(このアドレスを生成するのに用いられる鍵バージョンを示す) 鍵バージョンビット「v」、「 $X = P_1 \text{ XOR } H(K_1, M, X_2) \mid \dots \mid P_k \text{ XOR } H(K_k, M, X_{k-1})$ 」、「 $W = H(B(2, K_2), X_2, M) \text{ XOR } V_2, \dots, H(B(d-1, K_{(d-1)}), X_d, M) \text{ XOR } V_d$ 」、「 $Z = P_1 \text{ XOR } H(T_2, M, X_1) \mid P_2 \text{ XOR } H(T_3, M, X_2) \mid \dots \mid P_{k-1} \text{ XOR } H(T_k, M, X_{k-1})$ 」、「M」

40

【0189】

機能240においては、上述のC P PアドレスAは、アドレスサーバに受け渡され、当該アドレスサーバは、アドレス有効期間及び「 $P_0, \dots, P_k$ 」と共に、Aを、「 $P_0, P_1, \dots, P_k$ 」に対応するサブネット内のアクセスルータ(AR)上のDHCPv6へ受け渡す。(図19に示す) C P Pレベルルータ142は、(図19に示す) 通信パス124を介して、鍵サーバから鍵を得る。ルータは、これらの鍵を、新たな鍵期間が始まる前に得なければならない。

【0190】

50

例えば、鍵期間は、1日の長さを有してもよい。これらの鍵は、（例えば、IPsec、TLS、ケルベロス、又は、AAAが導いた暗号化プロトコル等の）暗号化プロトコルを用いて暗号化される。鍵を配信するプロトコルは、相互認証も提供しなければならない。ルータと鍵サーバとの間のプロトコルは、次のフィールド（すなわち、「ルータ（DNSネーム又は他の識別子）」、「鍵使用（レベル鍵、クロスオーバーシークレット、クロスオーバーレベル、又は、ツリー鍵）」、「鍵種別（アルゴリズム）」、「鍵」及び「バージョン」）を含む。スマートカードは、それらの鍵がホストに対して使用可能になるのを防ぐために、暗号化プロトコル用のセッション鍵を管理する。CPP鍵は、スマートカード上で、暗号化プロトコルのセッション鍵で暗号化される。プロトコルコード自体は、スマートカード上に実装してもよい。

10

【0191】

マスター鍵生成器は、（図19に示す）130を介して、バックアップ鍵生成器136に対して、ステータスデータを定期的に転送する。このようなステータスデータは、当該マスターが「M」ブロックを配布したサーバかについての情報、及び、当該「M」ブロックの値に関する情報を含んでもよい。当該バックアップは、上述の鍵サーバと同様のローカル鍵キャッシュを構築することもできる。当該バックアップは、当該マスターが故障した場合に、当該マスターの機能を引き継ぐ。当該バックアップ及び当該マスターは、（図19にも示す）ファイアウォール148を用いて、互いに、及び、アドレス／鍵サーバホストを除いて、いかなるホストとも通信することが防止されている。

【0192】

20

次に、本発明の「ICMP（Internet Control Message Protocol）」との相互作用について説明する。「traceroute」等のパケットによって取られたパスを判断する標準的なIPツールは、修正しない限り、又は、IPsecを用いない限り、ロケーションプライバシーを危うくする。「traceroute」は、接続の他端の通信相手に到達するまで、徐々に増加する「TTL（time to live）」値を伴ってパケットを送出することにより、クライアントのトポロジーにおけるロケーションを決める。

【0193】

通信相手に到達するのに要されるより小さな「TTL」は、減少して「0」になる。TTL値が減少して「0」になると、「減少」を行うルータは、「ICMP時間超過メッセージ（ICMP Time Exceeded message）」を発信元へ返信する。このパケットにおける発信元アドレスは、「TTL=0」が検知されたルータである。攻撃者は、このような情報を利用して、ネットワークのトポロジーと地理とを対応させるマップを形成することができる。

30

【0194】

無許可のロケーション情報の決定を防ぐため、ルータは、送信者が、トポロジー情報を要求するのを許可されているか否かについて判断することができなければ、「時間超過メッセージ」の送信をやめるべきである。他のICMPエラーメッセージも同じ分類である。ルータ又はホストは、受信者に、メッセージを受信することが許可されていることが分かっている限り、CPPアドレスを用いて、ホストに対して、かかるメッセージのうちの1つを送信してはいけない。具体的には、受信者は、CPPアドレスと当該受信者のロケーションとの間のマッピングによって信用されなければならない。IPsecは、このシナリオにおいて認証を提供するための候補プロトコルである。

40

【0195】

本発明は、本願明細書において説明したように、ローカルリンク上での盗聴者に対する防御も提供する。ローカルリンク上での盗聴者に対する防御を所望する場合、CPP IPアドレスは、当該ローカルリンク上で暗号化されなければならない。例えば、IPsecトンネルを、ホストからアクセスルータへのパケットをトンネリングするのに用いることができる。内部のIPsecトンネルアドレスは、CPPアドレスである。外部のトンネルアドレスについては、

50

(1) 「トポロジー的に正しい I P アドレス」 — かかるアドレスは、企業内ルーティングを良好に機能させることができる（一方、かかるアドレスは、ホストのロケーションを当該ホスト上の悪意のあるソフトウェアにさらす）、又は、

(2) 「任意に生成された I P アドレス」 — かかるアドレスは、悪意のあるソフトウェアが、(1) のようなトポロジー的に正しい I P アドレスにアクセスするのを防ぐ（この場合、ホストルートを用いなければならない）、のいずれかとすることができる。

【0196】

追加の防御について、C P P ルータは、当該 C P P ルータのレイヤ 2 リンクを介して、I P s e c を用いることもできる。このようなことは、当該リンク上での盗聴者に対する追加の防御を提供する。一般に、盗聴者は、転送パス上で、ホストのローカルリンクからのアドレスを追跡する必要がある、かつ、当該リンクのうちの 1 つで、C P P I P アドレスを明瞭に見ようとする。I P s e c 暗号化は、この種の攻撃アタックを防ぐ。

【0197】

本発明には、多くの展開オプションがある。例えば、C P P ルータは、オーバレイルータとして実施することができ、トンネルを用いることができる。このため、全てのルータを修正することを防ぐ。別のオプションは、境界ルータのみをレベルルータにすることである。そして、かかるルータは、同じ秘密鍵を共用する（かかる場合、C P P プレフィクスは、1 つのみのセグメントを有する）。このようなルータは、パケットを受信したときに、プレフィクスを解読して、真の宛先 I P アドレスを得る。そして、かかるルータは、プレフィクスをホップバイホップオプションに封じ込めて、当該パケットを次のホップへ転送する。他の展開オプションも、当業者には明白である。

【0198】

図 21 は、ソフトウェアをベースとするルータの考えられ得る実装を示す。かかるルータは、バス 2421 に接続された多数のネットワークインタフェース（この場合、2402…2412 の 6 個）によって構成されており、少なくとも 1 つの中央処理装置（C P U ）2414 と、揮発性メモリ 2418 と、（プログラマブル R O M 等の）不揮発性メモリ 2420 と、セキュリティ及び鍵関連機能のためのスマートカードインタフェース 2416 とを含む。かかる実装は、キーボード、マウス及びディスプレイ等のいくつかの他の I / O インタフェースを必要に応じて有してもよい。このようなシステムは、本願明細書で言及したロジック及びアルゴリズムを実装し、このような実装は、ソフトウェアを用いて（全体的にまたは部分的に）行われてもよい。

【0199】

本発明の有利な特徴は、以下の内容を含む。

(1) ホスト上の悪意のあるソフトウェアからの攻撃に対する防御。この場合、I P アドレスは、未許可で信頼の置けないソフトウェアにさらされる；

(2) 障害が発生したルータに対する防御。この場合、I P アドレス空間のスクランプリングは無防備である；

(3) 盗聴者に対する保護；

(4) 最適なルーティング、及び、トラヒック集中ポイント／部分的に最適なルーティングの防止；

(5) ネットワークの構成要素が故障した場合の安定性、パケットを配信し続けることを保証すること；

(6) 基本的なインバウンドルーティングアルゴリズムによって示したような安全性；

(7) 追加の統合をスパニングツリーの外でできるようにする正規化、従ってルーティングを改善；

(8) トンネリングの排除（唯一の追加のオーバヘッドは、ホップバイホップオプション用の 6～8 バイトであり、これは、例えば、H M I P v 6 及びモバイル I P v 6 に必要なトンネリングオーバヘッドの僅かな部分である）；

(9) I P s e c 及び他のセキュリティプロトコルとのネガティブな相互作用がないこと

10

20

30

40

50

。

【0200】

上述の実施形態の説明は、当業者が本発明を実行しかつ利用できるように記載されている。これらの実施形態に対する様々な変更は、当業者には容易に理解でき、本願明細書で定義された包括的な原理及び特定の例は、発明の機能を用いていない他の実施形態に適用されてもよい。例えば、上述の異なる実施形態の特徴のうちのいくつか又は全てについて、かかる実施形態から削除してもよい。従って、本発明は、本願明細書に記載した実施形態に限定しようとするものではなく、特許請求の範囲及びそれらの均等物によってのみ定義される最も広い範囲を認容すべきである。

【図面の簡単な説明】

10

【0201】

【図1】本発明の実施形態に係る暗号で保護されたプレフィクス（CPP）を用いた例示的なネットワークを示す。

【図2】本発明の実施形態に係る統合からプレフィクス要素を得ることの実施例である。

【図3】本発明の実施形態に係る（IPv6を例として用いた）CPPインターネットプロトコルアドレスの構造を示す。

【図4】鍵のバージョンを示すビットを含み、かつ、鍵のロールオーバーをサポートする本発明の例示的な実施形態を示す。

【図5】CPPプライバシードメインルータの処理機能を含む本発明の例示的な実施形態を示す。

20

【図6】ホップバイホップオプションを含む本発明の例示的な実施形態を示す。

【図7】CPP転送アルゴリズムを含む本発明の例示的な実施形態を示す。

【図8】ルーティング障害を考慮したCPP転送アルゴリズムを含む本発明の例示的な実施形態を示す。

【図9】正規化の例を含む本発明の例示的な実施形態を示す。

【図10】最適化されたルーティングスキームを含む本発明の例示的な実施形態を示す。

【図11】（例示的な形式においてIPv6を用いた）プレフィクスにより識別されるサブネット内にCPP拡張アドレスを含む本発明の例示的な実施形態を示す。

【図12】最適化されたルーティングを含む本発明の例示的な実施形態を示す。

【図13】CPPルータ機能を含む本発明の例示的な実施形態を示す。

30

【図14】ルーティングアドレス生成を含む本発明の例示的な実施形態を示す。

【図15】最適化されたルーティング用のアドレス構造を含む本発明の例示的な実施形態を示す。

【図16】最適化されたルーティング用の追加のアドレス構造を含む本発明の例示的な実施形態を示す。

【図17】最適化されたルーティング用の特別な部分的なケースを含む本発明の例示的な実施形態を示す。

【図18】鍵ハッシュで最適化されたルーティング機能の例を含む本発明の例示的な実施形態を示す。

【図19】アーキテクチャ要素の一例を含む本発明の例示的な実施形態を示す。

40

【図20】最適化されたルーティングの鍵ハッシュの変形が用いられたときに、拡張CPPを生成するのに用いられる機能の一例を含む本発明の例示的な実施形態を示す。

【図21】本発明の例示的な実施形態をソフトウェアで実施するための考えられ得るハードウェアを示す。

【図 1】

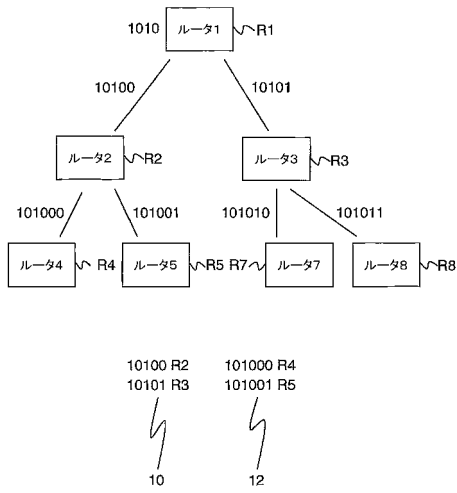


Figure 1

【図 2】

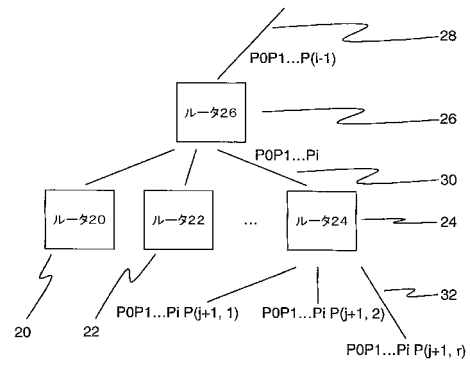


Figure 2

【図 3】

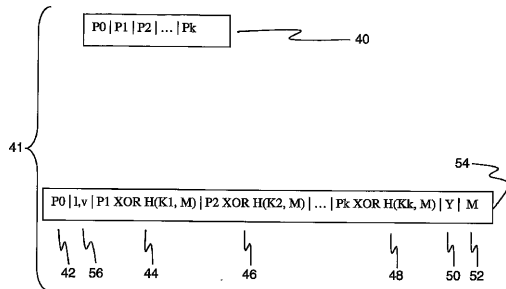


Figure 3

【図 4】

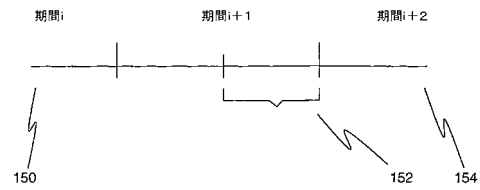


Figure 4

【図 5】

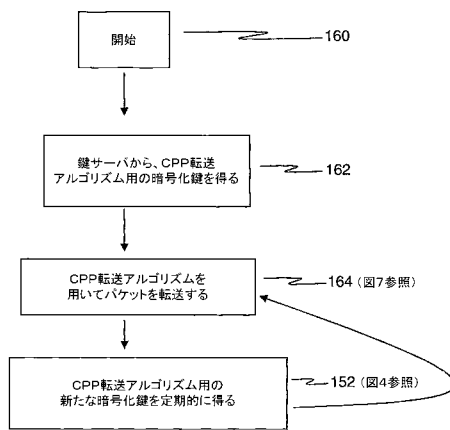


Figure 5

【図 6】

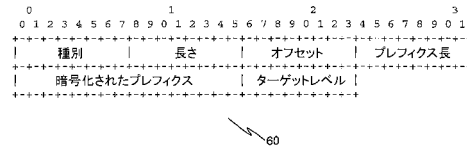


Figure 6

【図 7】

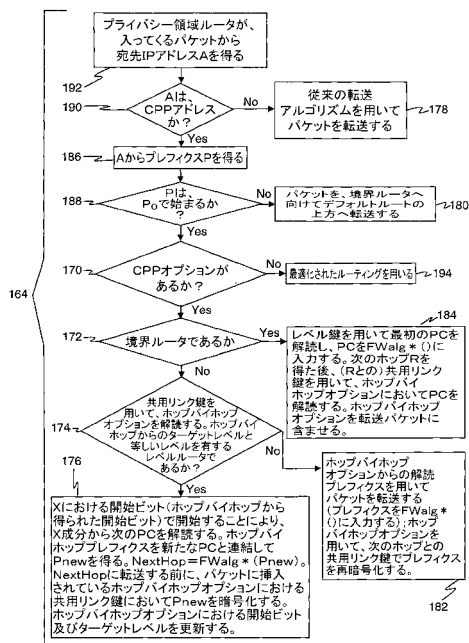


Figure 7

【図 8】

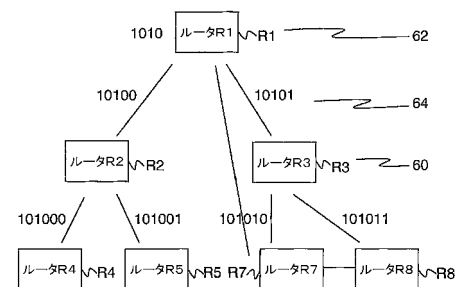


Figure 8

【図 9】

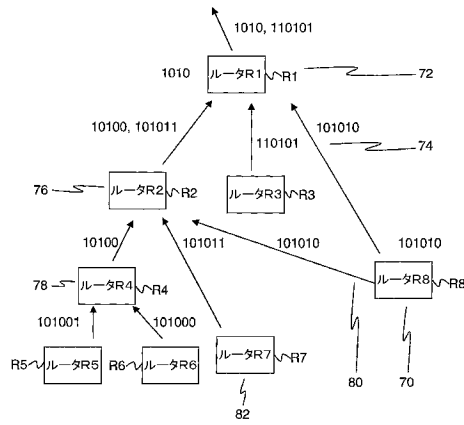


Figure 9

【図 10】

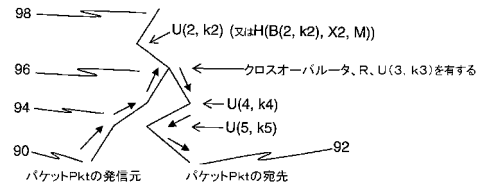


Figure 10

【図 11】

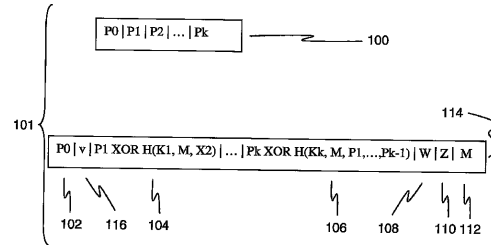


Figure 11

【図 12】

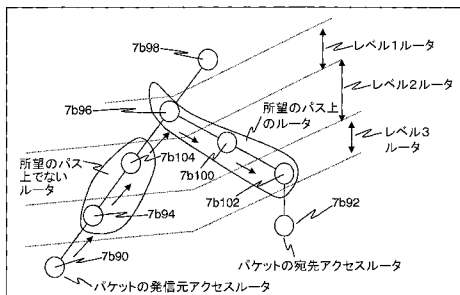


Figure 12

【図 13】

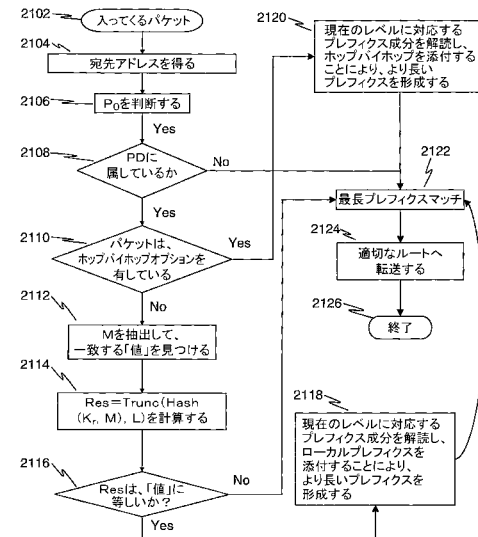


Figure 13

【図 14】

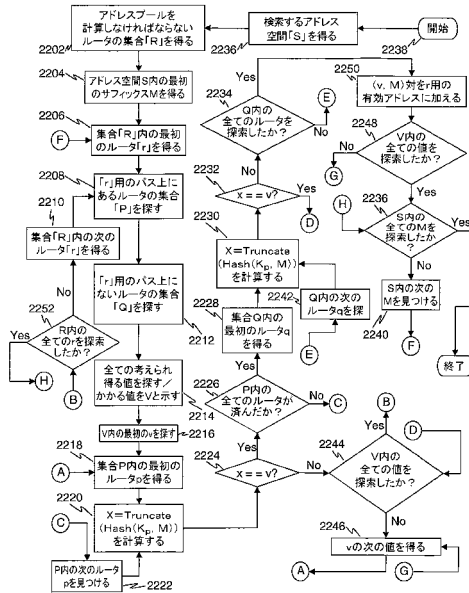


Figure 14

【図 15】

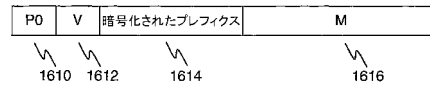


Figure 15

【図 16】

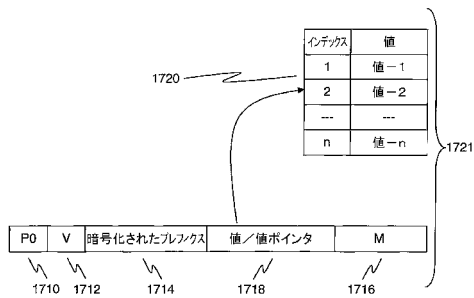


Figure 16

【図 17】

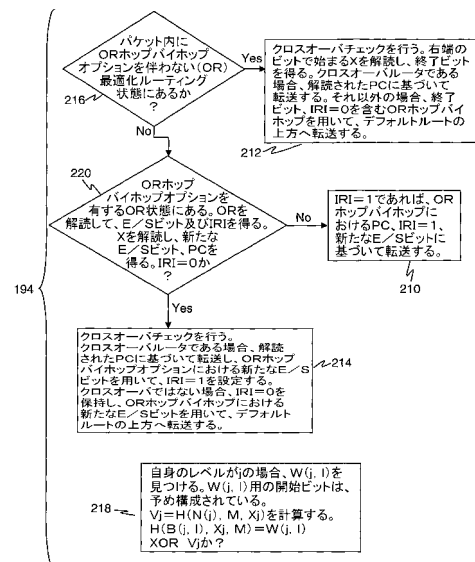


Figure 17

【図 18】

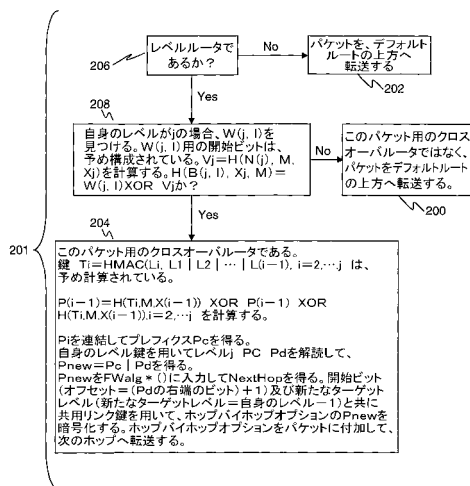


Figure 18

【図 19】

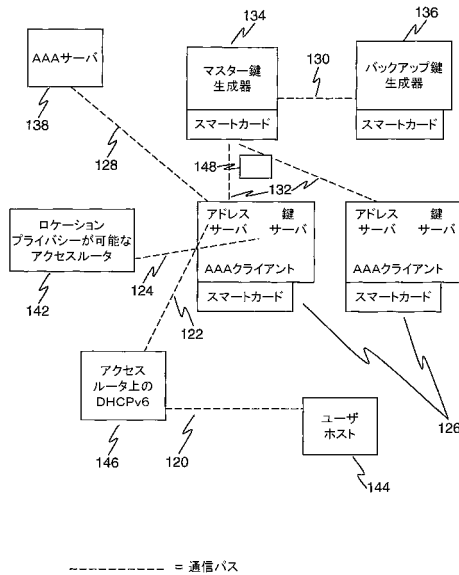


Figure 19

【図 20】

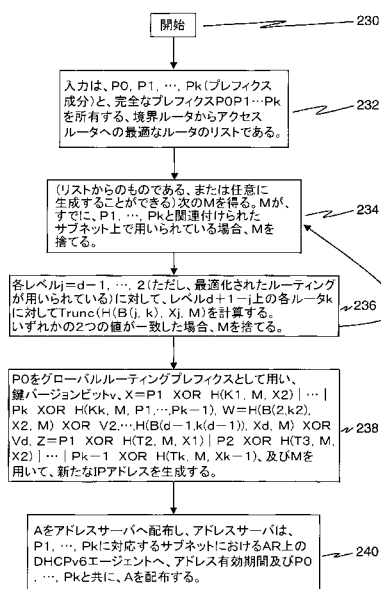


Figure 20

【図 2 1】

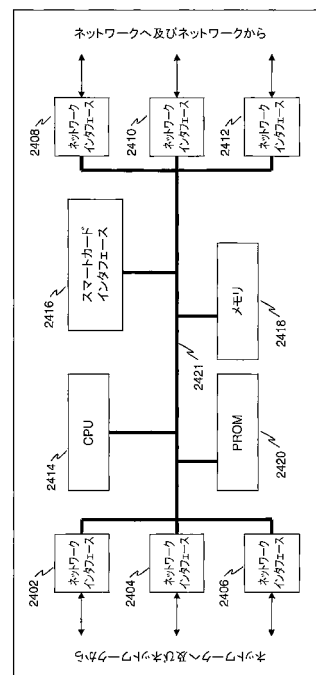


Figure 21

フロントページの続き

(31)優先権主張番号 10/851,846

(32)優先日 平成16年5月20日(2004.5.20)

(33)優先権主張国 米国(US)

(72)発明者 ビン タリク、 ムハマッド ムカラム

アメリカ合衆国, カリフォルニア州 95110, サンノゼ, スイート300, メトロ
ドライブ 181

(72)発明者 ケンプ、 ジェームス

アメリカ合衆国, カリフォルニア州 95110, サンノゼ, スイート300, メトロ
ドライブ 181

(72)発明者 松岡 保静

アメリカ合衆国, カリフォルニア州 95110, サンノゼ, スイート300, メトロ
ドライブ 181

(72)発明者 ジェイン、 ラビ クマー

アメリカ合衆国, カリフォルニア州 95110, サンノゼ, スイート300, メトロ
ドライブ 181

(72)発明者 河原 敏朗

アメリカ合衆国, カリフォルニア州 95110, サンノゼ, スイート300, メトロ
ドライブ 181

審査官 矢頭 尚之

(56)参考文献 特開平11-331152 (JP, A)

特開2001-237896 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H04L 12/56

G06F 21/20