Vol. 45 No. 2 情報処理学会論文誌 Feb. 2004

RSIPサーバにおけるポート番号による パケットフィルタリングの提案とその性能検証

石川大法[†] 木村成伴[††] 海老原義彦[††]

IPSecではIPのデータ領域を暗号化するため，TCPやUDPのポート番号を使用したアドレス変換を受けられない．この問題を解決するため，RSIP（Realm Specific IP）ではゲートウェイに通信相手のIPアドレスやポート番号などを登録しておき，送信元がポート番号を使用したアドレス変換を実行する．しかし，本方式を用いても途中経路上でポート番号が参照できないため，ファイアウォールでセキュリティポリシを強制することができない．そこで本論文では，RSIPクライアントが外部ホストにアクセスする場合を対象とし，RSIPサーバと外部ホストが協力することで暗号化されたデータグラムのポート番号によるパケットフィルタリングを実現する．このため，本方式では暗号化したデータグラムにポート番号などの情報のハッシュ値と同情報を暗号化した値を追加する．これらの情報から，RSIPサーバにポート番号を確認する手段を与えるとともに，このポート番号を外部ホストがデータグラムに上書きすることによって，RSIPサーバに登録した以外のポート番号を利用することを妨げる．次に，本方式におけるRSIPクライアント，RSIPサーバ，外部ホストをそれぞれFreeBSD上で実装し，本実装システムで機能検証と性能検証を行うことで，提案方式により十分なスループットを確保しつつ，セキュリティポリシを強制できることを示す．

A Proposal of Packet Filtering Methods Based on Port Numbers at RSIP Servers and the Performance Evaluation

Hironori Ishikawa,[†] Shigetomo Kimura[††] and Yoshihiko Ebihara[††]

In IPSec, since IP data may be encrypted, NAPT (network port address translation) cannot be functioned. To solve this problem, in RSIP (Realm Specific IP), a source host registers the IP address and port number of the destination host to the gateway, and then executes NAPT by itself. Even if RSIP is used, however, the port numbers cannot be referred between the source host and destination host, and thus the firewall cannot enforce the security policy. This paper subjects the cases of accessing a RSIP client to outside host, and realizes filtering of encrypted datagrams based on the port numbers by cooperating the RSIP server and outside host. From this purpose, each datagram has the hashed value of the informations such as the port numbers, and the encrypted data of the informations. These values provide the method to confirm the port numbers for the RSIP server. The outside host overwrites the port numbers on the decrypted datagrams to prevent that the RSIP client uses unregistered port numbers. Finally, the proposal system is implemented on FreeBSD. The function validation and performance evaluation for the implementation system conclude that our system can provide the sufficient throughput and also enforce the security policy.

1. はじめに

近年，インターネットの急速な普及により，インターネットを介した電子商取引などへの需要が増加してきている．しかし，これらのサービスにおいては，金銭や個人情報のやりとりが必要となる場合が多いため，通信内容の改竄や盗聴に対する対応策を講じる必要がある．このため，SSHやSSL，IPSec[1]などの暗号化通信プロトコルが開発されている．

SSHとSSLはTCPのデータ領域を暗号化するプロトコルであるが，各アプリケーションがこれらのプロトコルを利用するように実装する必要がある．これに対し，IPSecでは，IPデータグラムのデータ領域の暗号化や送信者の署名を加える機能を提供しており，

† 筑波大学大学院理工学研究科
Master's Program in Science and Engineering, University of Tsukuba
†† 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba

これらを用いることで，アプリケーションが意識することなくIPデータグラムの盗聴，改竄を防ぐことができる．

IPSecでは，暗号化を行うための暗号化アルゴリズムやその鍵をまとめたものをSA（Security Association），そのインデックス値をSPI（Security Parameter Index），暗号化された領域をESP（Encapsulating Security Payload）[2]と呼ぶ．IPSecで暗号化されたIPデータグラムにはこのSPIが含まれ，受信側はこの値から復号に必要なSAを特定している．したがって，SAは送信側と受信側で共有される必要があり，またデータグラムの行きと帰りとで別々に設定する必要がある．

また，IPSecでの暗号化の際は，IPデータグラムのデータ領域を暗号化し，IPヘッダの後ろにESPヘッダが挿入される．このESPヘッダには，暗号化されたデータ領域を復号するために必要なSPIなどが納められている．受信したホストは，このSPIを基に，復号に必要なSAを特定し，復号する．ただし，SAはホストの管理者が設定するものであり，第三者がSPIの値によって使用された暗号化アルゴリズムやキーを知ることはできない．さらに，データグラムが改竄されていないことを保証するために，AH（Authentication Header）[3]を加えることもできる．AHによる認証の範囲はIPデータグラム全体であり，データ領域だけでなくIPヘッダも認証の範囲である．

ところで，IPSecではTCPやUDPのヘッダも暗号化してしまうため，そのポート番号を途中経路上で読み取ることが不可能である．しかし，ファイアウォールやNAPT（Network Address Port Translator）[4]はその処理のためにポート番号を使用するため，前者はポート番号によるフィルタリングが，後者は変換テーブルの作成ができなくなる．ほかにも，AHを加えて認証を行う場合，途中経路上にあるNAPTやファイアウォールでデータグラムに変更を加えると，受信時に改竄がなされていると判断され，データグラムが破棄されてしまうという問題も生じる[5]．

この問題のうち，NAPTのサービスを受けられない問題とAHを使用できない問題を解決するため，現在，RSIP（Realm Specific IP）[6)~8)]が提案されている．この方式では，**図1**に示すネットワーク構成において，RSIPクライアントが属する内部ネットワークと外部ネットワークの境界にゲートウェイであるRSIPサーバを設置する．NAPTではゲートウェイにおいてポート番号を用いたアドレス変換を行うが，RSIPではRSIPクライアントがIPSecを用いて外部ホストに通信する際に，あらかじめRSIPサーバに通信相手のIPアドレスやポート番号などを通知し，RSIPクライアント自身がポート番号を使用したアドレス変換を行う[9),10)]．しかし，このRSIPを使用しても，ゲートウェイ上のファイアウォールでセキュリティポリシを強制することができないという問題は解決されていない．

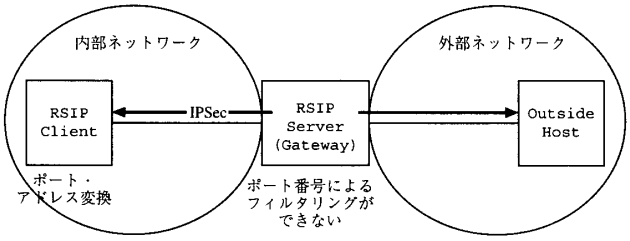


図1 対象とするネットワーク
Fig. 1 Target network.

そこで本論文では，図1に示すネットワーク構成を対象とし，RSIPクライアントは外部ホストにアクセス可能であるが，その逆はできない状況を仮定する．この状況は，内部ネットワークのマシンにプライベートアドレスが付与されており，外部ネットワークのサーバと接続する際にはグローバルアドレスに変換する必要があるというNAPTもしくはRSIPが用いられる典型的なネットワーク構成に相当する．この条件の下で，RSIPとIPSecのアルゴリズムを改良し，暗号化されたデータグラムのポート番号によるパケットフィルタリングを実現する[11)]．

本方式によるフィルタリングが可能になると，内から外へ通信できるプロトコルを制限することができる．これにより，内部情報の漏洩やネットワークの私的利用などを防ぐ目的から，内部ネットワークから対外接続のために利用できるアプリケーションプログラムをある程度限定することが可能となり，内部ネットワークの管理コストを低減させることが期待される．また，内部ネットワークにおけるアプリケーションプログラムにセキュリティホールがあった場合，外部ネットワークへの攻撃手段を制約することができる効果も期待される．このように，内から外への通信を無制限に許可したくない場合は，これらの通信に対してセキュリティポリシを適用する必要があり，本方式によるフィルタリングには意義がある．

本論文の構成は以下のとおりである．まず，2章においてRSIPについて概説する．3章では提案方式について述べ，4章で同方式の実装について説明する．最後に，5章で本論文のまとめと今後の課題について述べる．

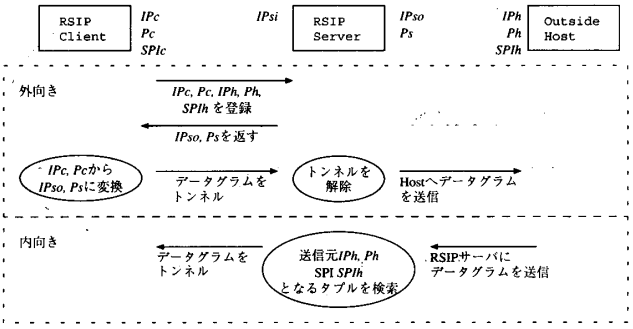


図 2 RSIP のフレームワーク
Fig. 2 Framework of RSIP.

2. RSIP（Realm Specific IP）

RSIP（Realm Specific IP）[5),6)] では，あらかじめ RSIP サーバ（ゲートウェイ）に通信相手の IP アドレスやポート番号，SPI などを通知しておき，送信元がこれらの変換を行うことで NAPT の機能を実現する．なお，使用する SPI が重複しないように調整することで，RSIP サーバはデータグラムをどのローカルホストに返したらよいのかを判別することができる．

2.1 RSIP のフレームワーク

以下では RSIP の動作を，RSIP クライアントから外部ホストにデータグラムを送る場合と，外部ホストから RSIP クライアントにデータグラムを送る場合に分けて説明する．

2.1.1 外部ホストに対してデータグラムを送る場合

RSIP は**図 2** のように，RSIP クライアントと外部ホストが 2 つのネットワークの間に位置する RSIP サーバを通じて通信を行うことを想定している．ここで RSIP クライアントの IP アドレスを IP_c，発信ポート番号を P_c，RSIP サーバの内部アドレスを IP_{si}，外部アドレスを IP_{so}，RSIP クライアントのために RSIP サーバが外部ホストにデータグラムを送る際の発信ポート番号を P_s，外部ホストの IP アドレスを IP_h，ポート番号を P_h，外部ホストから RSIP クライアントへデータグラムを送る際の暗号化に使用する SPI を SPI_h とする．

(1) RSIP クライアントは RSIP サーバに自分を RSIP クライアントとして登録した後，RSIP サーバにアサインリクエストを送る．これは RSIP クライアントがどの外部ホストと通信を行うかを宣言し，RSIP サーバに IP_c，P_c，IP_h，P_h，SPI_h の登録を要求するものであり，RSIP プロトコルによって行われる．

(2) アサインリクエストを受けた RSIP サーバは，要求内容に問題がなければそれをアサインリストに登録し，IP_{so}，P_s を RSIP クライアントに返す．これらは RSIP クライアントがアドレス変換を行う際に必要な情報である．

(3) 次に RSIP クライアントは，IP_c，P_c を RSIP サーバから渡された IP_{so}，P_s に書き換えたデータグラムを用意し，これを IPSec で暗号化する．

(4) 暗号化されたデータグラムはカプセル化されて RSIP サーバに送られる．カプセル化には IP-IP Encapsulation[12)]，GRE[13)]，L2TP[14)] のどれかが使用される．

(5) これを受け取った RSIP サーバは，暗号化されたデータグラムの宛先となっているホストにデータグラムを送信する．

2.1.2 外部ホストからデータグラムを受け取る場合

次に，RSIP クライアントからのデータグラムを受け取った外部ホストが，その応答を返す場合について，図 2 を用いて説明する．外部ホストが受け取ったデータグラムに書かれている送信元は RSIP サーバになっているため，応答に用いるデータグラムの宛先は RSIP サーバとなる．

(1) 外部ホストは RSIP サーバ宛のデータグラムを用意し，これを暗号化したものを送信する．

(2) データグラムが RSIP サーバに届くと，RSIP サーバは登録されたアサインリストを検索し，外部ホストの IP アドレスが IP_h，ポート番号が P_h，SPI が SPI_h であるものを探す．

(3) 該当するものを検出できた場合，RSIP サーバは受け取ったデータグラムを検出したアサインリストの RSIP クライアントにトンネリングする．

以上のように，途中経路上ではなく送信元でアドレス変換を施すことによって，ゲートウェイがポート番号を読み取る必要性をなくし，さらにゲートウェイがデータグラムを書き換えてしまうことによって AH を使用できなくなるという問題を解決している．

2.2 RSIP プロトコルのメッセージフォーマット

RSIP クライアントが RSIP サーバにアサインリクエストを出し，その結果を受け取るために RSIP プロトコル[15)] が使用される．この RSIP プロトコルは TCP 上で動作する．

図 3 に RSIP プロトコルのメッセージフォーマットを示す．Version フィールドにはプロトコルのバージョンを指定する．現在のバージョンは 1 である．Message Type フィールドには，メッセージタイプに関連づけられた値を指定する．このメッセージタイプによって，

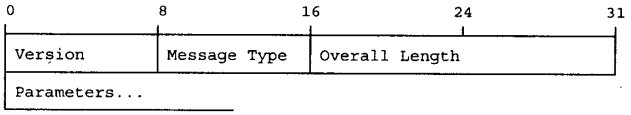


図 3 RSIP プロトコルのメッセージフォーマット
Fig. 3 Message format for RSIP protocol.

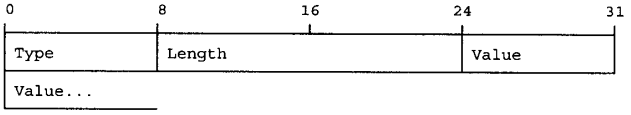


図 4 RSIP プロトコルのパラメータフォーマット
Fig. 4 Parameter format for RSIP protocol.

アサインリクエストやその応答，エラーメッセージなどを設定している．Overall Length フィールドは，送信するメッセージの全体長を指定する．つまり，Version フィールドからパラメータの最後までの長さをバイト単位で示す．

Parameters フィールドには，Message Type フィールドで指定したメッセージタイプに必要なパラメータが設定される．パラメータは**図 4** に示すフォーマットとなっている．Type フィールドには，パラメータのタイプを指定する．たとえばアドレスは 1，ポート番号は 2 などの ID がふられており，該当する ID を Type フィールドに指定する．Length フィールドは Value フィールドの長さをバイト単位で指定する．

最後に，RFC3103[15] と RFC3104[9] で定義されたメッセージタイプを**表 1** に示す．たとえば，RSIP クライアントの登録を要求する場合は，RSIP クライアントは RSIP サーバに対して REGISTER_REQUEST を送る．これに成功した場合は，RSIP サーバから RSIP_RESPONSE が返される．一方，なんらかのエラーが発生した場合は ERROR_RESPONSE が返される．その後，RSIP クライアントが外部ホストと IPSec による通信をする場合は，ASSIGN_REQUEST_RSIPSEC を RSIP サーバに送り，要求が受理されると ASSIGN_RESPONSE_RSIPSEC が返される．

3. RSIP サーバにおけるポート番号によるパケットフィルタリングの提案

前章で述べた RSIP によって，IPSec で暗号化されたデータグラムを RSIP サーバ（ゲートウェイ）を超えて送ることができるようになった．しかし，途中経路上でポート番号を読み取れないため，依然としてファイアウォールのサービスを受けられないという問題が残っている．たとえ RSIP サーバが受け取ったアサインリクエストで通過が許可されるポート番号が登録されていたとしても，実際に外部ホストとの間でやりとりされるデータグラムで登録どおりのポート番号が使用されているかどうかは，データグラムを復号しない限り確認することができない．

表 1 RSIP プロトコルのメッセージタイプ
Table 1 Message types for RSIP protocol.

Value	Message
1	ERROR_RESPONSE
2	REGISTER_REQUEST
3	REGISTER_RESPONSE
4	DE-REGISTER_REQUEST
5	DE-REGISTER_RESPONSE
6	ASSIGN_REQUEST_RSA_IP
7	ASSIGN_RESPONSE_RSA_IP
8	ASSIGN_REQUEST_RSAP_IP
9	ASSIGN_RESPONSE_RSAP_IP
10	EXTEND_REQUEST
11	EXTEND_RESPONSE
12	FREE_REQUEST
13	FREE_RESPONSE
14	QUERY_REQUEST
15	QUERY_RESPONSE
16	LISTEN_REQUEST
17	LISTEN_RESPONSE
22	ASSIGN_REQUEST_RSIPSEC
23	ASSIGN_RESPONSE_RSIPSEC

そこで本章では RSIP のプロトコルを改良することで，この問題を解決する．従来の RSIP と提案方式の違いは，登録されたポート番号であるか否かを識別する 2 つの値をデータグラムに添付することにある．そして，これらの値を用い，RSIP サーバと外部ホストが協力することで，RSIP サーバに登録されていないポート番号を持つデータグラムのフィルタリングを実現するものである．

以下では，図 1 のネットワーク環境において，内部ネットワークにある RSIP クライアントから外部ホストに対してセッションの開設が要求される場合に限定して議論を進める．

3.1 外部ホストに対してデータグラムを送る場合

まず，**図 5** を用いて，RSIP クライアントから外部ホストへデータグラムを送信する手順について説明する．ここで，r_c，r_h を乱数値，h_c を (r_c, P_s, P_h, r_h) のハッシュ値とする．また，k_c を (r_c, r_h, P_s, P_h) を RSIP クライアントと外部ホストの共通鍵で暗号化した値とする．その他の値は，図 2 と同じである．

なお，r_c と r_h は h_c と k_c を作成する際に用いる nonce として使用している．次節で述べるように，これらの値は外部ホストから RSIP クライアントへデータグラムが送られる場合にも使われている．このため，第三者に通信内容を推測されにくくする目的から，本

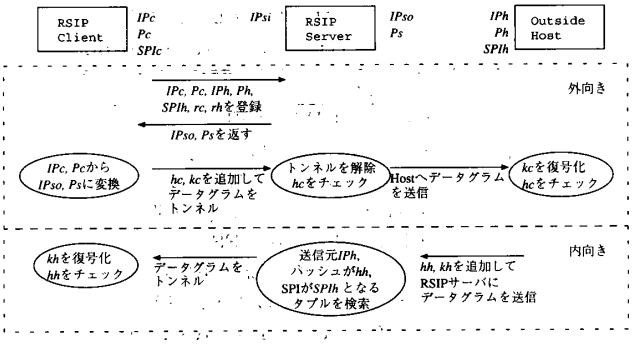


図 5 RSIP サーバにおけるポート番号によるパケットフィルタリング

Fig. 5 Packet filtering based on port numbers for RSIP servers.

方式では乱数を 2 つ用い，さらに，送る方向によって乱数を用いる順序を入れ換えている．また，RSIP クライアントと外部ホストの共通鍵の鍵管理ポリシについては，本提案方式では特に制約はなく，IKE などの鍵管理プロトコルを使用することが可能である．

(1) RSIP クライアントは，RSIP サーバにアサインリクエストを送信し，IP_c，P_c，IP_h，P_h，SPI_h，r_c，r_h の登録を要求する．この情報がセキュリティポリシの要件を満たすならば，RSIP サーバはこれらをアサインリストに追加する．

(2) RSIP サーバは，IP_{so} と P_s を返す．これを使い，RSIP クライアントはデータグラムのアドレス変換を行う．

(3) RSIP クライアントは，送信するデータグラムに h_c と k_c を追加して RSIP サーバにトンネルする．

(4) トンネルされたデータグラムを受け取った RSIP サーバは，登録された情報からハッシュ値を計算し，データグラムの h_c と一致することを確認する．正しければ，このデータグラムを宛先のホストに送信する．

(5) データグラムを受け取った外部ホストは，k_c を復号することでポート番号の対 P_s，P_h と乱数値 r_c，r_h を取得する．これらの値を用いて，h_c が偽造されていないことを確認するとともに，ここで得られたポート番号の対をデータグラムのポート番号フィールドに上書きする．

以上のようにプロトコルを改良することで，データグラムに含まれるポート番号を RSIP サーバに登録されたポート番号に強制することができる．このプロトコルの中で RSIP クライアントがポート番号を詐称を試みるには以下の方法が考えられる．

(1) k_c に含まれる P_s，P_h を書き換える．

(2) 暗号化された TCP ヘッダや UDP ヘッダに含まれるポート番号を書き換える．

どちらの方法においても，h_c を書き換えれば RSIP サーバにおいてデータグラムが破棄される．したがって，RSIP サーバを通過させるためには，h_c を RSIP サーバに登録された値から算出できる正規の値にする必要がある．

ここで，(1) の手法をとってデータグラムが外部ホストに到着した場合，外部ホストはまず k_c を復号し，ポート番号 P_s，P_h と r_c，r_h を取得する．これらからハッシュ値を生成し，h_c と比較する．この際，h_c は正規の値であるが，生成されたハッシュ値は RSIP サーバに登録されたポート番号とは異なる値から生成されており，h_c とは一致しない．そのため，データグラムは破棄される．

(2) の手法をとってデータグラムが外部ホストに到着した場合，前述の外部ホストによるハッシュ値のチェックには通過するが，外部ホストは取得したポート番号 P_s，P_h を復号した TCP ヘッダや UDP ヘッダに上書きする．そのため，データグラムは RSIP サーバに登録された宛先ポート番号 P_h に届くことになる．

以上のように，ポート番号を詐称したデータグラムは，破棄されるか，RSIP サーバに登録されたポート番号に届くかのどちらかになる．ただし，外部ホストの協力がなければ，ポート番号の詐称を許すことになる．外部サーバが詐称を許しているかどうかを判別することは困難であるが，RSIP サーバは IP アドレスによるフィルタリングが可能であり，信頼できる外部ホスト以外の IPSec 接続を認めないというセキュリティポリシをとることはできる．このような状況では，外部ホストの協力が得られると仮定できる．

3.2 外部ホストからデータグラムを受け取る場合

次に，外部ホストから RSIP クライアントへデータグラムを送信する手順について図 5 に基づき説明する．ここで，h_h を (r_h, P_h, P_s, r_c) のハッシュ値，k_h を (r_h, r_c, P_h, P_s) を外部ホストとの共通鍵で暗号化したものとする．

(1) まず外部ホストは，RSIP サーバ宛に h_h と k_h を付加したデータグラムを送信する．

(2) このデータグラムを受け取った RSIP サーバは，アサインリストの中から，外側から内側へのデータグラムの送信元アドレスが IP_h，ハッシュ値が h_h，SPI が SPI_h となるタプルを探す．

(3) 該当するタプルが見つかった場合，RSIP サーバは関連づけられた RSIP クライアントにデータグラムをトンネルする．

(4) データグラムを受け取った RSIP クライアン

トは k_h を復号することで r_c, r_h とポート番号 P_h, P_s を取得し，さらに h_h が偽造されていないことを確認する．そして，ここで得られたポート番号の対をデータグラムのポート番号フィールドに上書きする．

このアルゴリズムによりポート番号の詐称ができないのは，3.1 節で述べた理由と同じである．以上の手順を用いることで，経路上の RSIP サーバ以外のホストにポート番号を知られることなく，登録されたポート番号の使用を強制することができる．

4. 提案方式の実装

本章では，前章で提案したシステムの実装方式について述べる．

4.1 追加メッセージ

まず，RSIP に加えて提案システムで追加したメッセージを運ぶために必要となる，メッセージタイプとパラメータ，および IP オプションについて述べる．

4.1.1 追加するメッセージタイプとパラメータ

提案システムで追加するメッセージタイプを**表 2** に示す．ASSIGN_REQUEST_RSE_IP は，RSIP クライアントが RSIP サーバに提案方式での外部ホストとの通信を要求するためのメッセージタイプであり，もう 1 つの ASSIGN_RESPONSE_RSE_IP は，RSIP サーバが ASSIGN_REQUEST_RSE_IP を許可した際に RSIP クライアントに返す応答である．ASSIGN_REQUEST_RSE_IP は IP_c, P_c, IP_h, P_h, SPI_h, r_c, r_h などをパラメータに持つ．要求に問題がない場合，RSIP サーバは ASSIGN_RESPONSE_RSE_IP を返す．ASSIGN_RESPONSE_RSE_IP は IP_{so}, P_s, IP_h, P_h, SPI_h, r_c, r_h などをパラメータに持つ．これを受信した RSIP クライアントはパラメータに含まれる IP_{so} を送信元アドレス，P_s を送信元ポート番号として使用する．

前述の ASSIGN_REQUEST_RSE_IP や ASSIGN_RESPONSE_RSE_IP で使用されるパラメータに r_c と r_h があるが，これは従来の RSIP プロトコルには定義されていない．そのため，これらを示すパラメータを追加する必要がある．追加されたパラメータのフォーマットを**図 6** に示す．先頭の Type フィールドには，このパラメータの ID の 13 を設定する．次の Length フィールドにはパラメータのデータのサイズを設定する．このパラメータは固定長であるため，Length フィールドはつねに 12 となる．そして 2 つの RandomValue フィールドは 6 バイトずつで，r_c, r_h の順で設定する．

表 2 追加するメッセージタイプ
Table 2 Additional message types.

Value	Message
18	ASSIGN_REQUEST_RSE_IP
19	ASSIGN_RESPONSE_RSE_IP

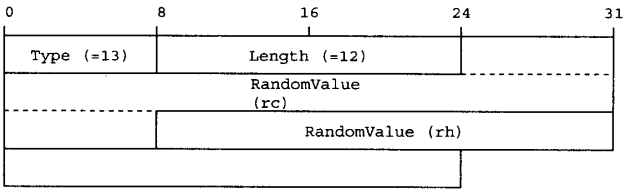


図 6 追加するパラメータ
Fig. 6 Additional parameters.

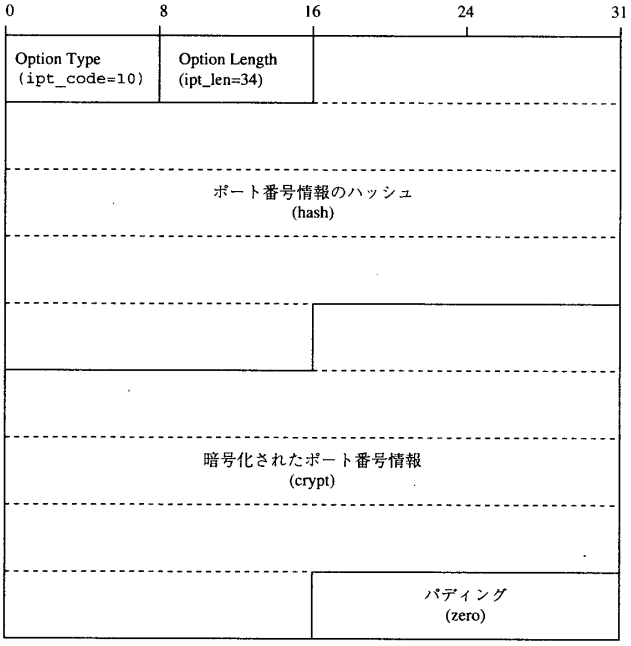


図 7 提案方式によって追加される新たな IP オプション
Fig. 7 Additional IP option for the proposal methods.

4.1.2 追加する IP オプション

RSIP クライアントと外部ホストの間でやりとりされるデータグラムには，ポート番号情報を暗号化した値とハッシュ値が含まれる．これらの値を**図 7** に示す IP オプションに格納して送信する．ここで，オプションタイプは 10 を指定する．また，このオプションはパディングを除いて 34 バイトであるため，オプション長には 34 を指定する．最後に，ポート番号情報のハッシュ値 16 バイトと暗号化した値 16 バイトを格納する．

ハッシュ値を求めるためには，**図 8** のフォーマットにポート番号と乱数をまとめ，MD5 を使ってハッシュ値を求める．ただし，RSIP クライアントから外部ホストに送る場合は，rand_from は r_c，rand_to は r_h，port_from は P_s，port_to は P_h となり，外部ホスト

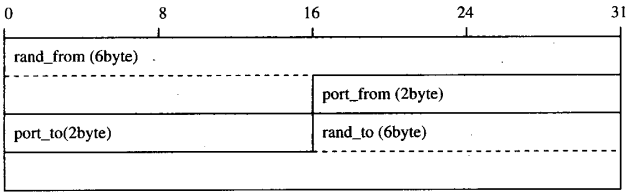


図 8 ハッシュ値を求めるためのフォーマット
Fig. 8 Format to calculate hashed value.

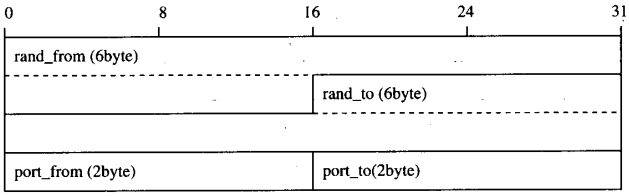


図 9 暗号化された値を求めるためのフォーマット
Fig. 9 Format to calculate encryption value.

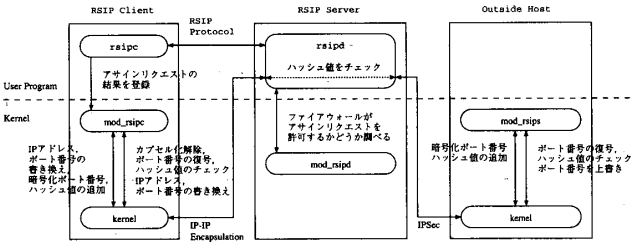


図 10 提案方式の実装
Fig. 10 Implementation of the proposal methods.

から RSIP クライアントに送る場合は，rand_from は r_h，rand_to は r_c，port_from は P_h，port_to は P_s となる．

暗号化した値を求めるためには，**図 9** のフォーマットにポート番号と乱数をまとめ，DES を使って暗号化する．ただし，rand_from, rand_to, port_from, port_to の値は，ハッシュ値を求める際の値と同じである．

4.2 FreeBSD への実装

本節では，前節で述べたメッセージタイプ，パラメータ，IP オプションを含めた提案システムを FreeBSD 上に実装する．ここで，RSIP クライアントは FreeBSD 4.6.2-RELEASE-p2 を，RSIP サーバは FreeBSD 4.6-RELEASE-p2 を，外部ホストは FreeBSD 4.7-RELEASE-p1 を用いた．実装システムの構成を**図 10** に示す．

4.2.1 RSIP クライアント

図 10 に示したように，RSIP クライアントはユーザランドのプログラム rsipc と，カーネルモジュール mod_rsipc から構成される．

rsipc は，RSIP プロトコルによるアサインリクエストを RSIP サーバに送付し，その結果をシステムコールを用いて mod_rsipc に登録する．この情報を基に，mod_rsipc では RSIP クライアントで入出力されるすべてのデータグラムを検査する．その後，提案方式に従って IP アドレスとポート番号の書換えや，カプセル化，ハッシュ値や暗号化されたポート番号の追加などを行う．

以下に，これらのモジュールを用いた動作の詳細について述べる．

4.2.1.1 アサインリクエストの処理

まず，rsipc によるアサインリクエストの処理手順を以下に示す．

(1) rsipc が起動されると，設定ファイルに記載された RSIP サーバに対して REGISTER_REQUEST を送信する．これが RSIP サーバに受理された場合は，REGISTER_RESPONSE が返り，次へ進む．そうでない場合は ERROR_RESPONSE が返され，rsipc は終了する．

(2) 次に，ASSIGN_REQUEST_RSE_IP を RSIP サーバに送信する．これが RSIP サーバに受理された場合は ASSIGN_RESPONSE_RSE_IP が返り，次へ進む．そうでない場合は ERROR_RESPONSE が返され，rsipc は終了する．

(3) システムコールを用いて，前項で得られた応答内容を mod_rsipc に登録する．

4.2.1.2 外部ホストへデータグラムを送信する際の処理

アサインリクエストが受理された後，RSIP クライアントが外部ホストへデータグラムを送る場合の処理手順を以下に示す．

(1) まず，TCP のデータグラムがカーネル内の関数 ip_output に渡されると，このデータグラムに従来の RSIP，または提案方式による変更を加える必要があるかどうかを調べる．

(2) 変更を加える必要がない場合は，そのまま送信する．変更を加える必要がある場合は，送信元 IP アドレスを IP_{so} に，送信元ポート番号を P_s に書き換える．また，書換え前の送信元ポート番号を記録しておく．これらの処理を行ったあと，TCP チェックサムの再計算を行う．

(3) 次に，データグラムに提案方式による拡張を加える必要があるかどうかを調べ，必要がある場合は，4.1.2 項で述べた IP オプションを追加する．

(4) 最後に IP チェックサムを再計算し，データグラムを送信する．

4.2.1.3 データグラムを受信する際の処理

最後に，RSIPクライアントが外部ホストからのデータグラムを受信する際の処理手順について説明する．

(1) 外部ホストから送られてきたデータグラムから，カプセル化で追加されたIPヘッダを除去する．
(2) ip_input関数の最初で，このデータグラムが従来のRSIPまたは提案方式による変更を加える必要があるかどうかを検査する．必要がない場合は，RSIPまたは提案方式のデータグラムではないため，そのまま従来のTCP/IPの処理を行う．
(3) 従来のRSIPによる変更を加える必要がある場合は，(5)へ進む．
(4) 提案方式による変更を加える必要がある場合は，データグラムのIPオプションを検査する．IPオプションを持っていない場合，または，事前にrsipcによって設定されたアサインレスポンスから算出できる値が含まれていない場合は，このデータグラムを廃棄する．
(5) 次に，ESPの復号に必要なSAを検索する．ここで，データグラムの宛先IPアドレスはRSIPクライアントのものではない．そこで，この検索は変更後のIPアドレスにより行う必要がある．
(6) 手に入れたSAでESPを復号する．成功した場合は，データグラムのIPアドレスとポート番号を変更する．このとき，TCPチェックサムの再計算が必要になる．ただし，IPチェックサムはすでにチェックが終了しているため，ここではその再計算は行わない．
(7) 以上の処理を終えた後，データグラムをtcp_inputなどの関数に渡す．

4.2.2 RSIPサーバ

図10に示したように，RSIPサーバはユーザランドのプログラムrsipdと，カーネルモジュールmod_rsipdから構成される．

rsipdは主に3つのスレッドから構成され，各スレッドにより，RSIPプロトコルによるアサインリクエストの受付と，RSIPクライアントと外部ホストの間でやりとりされるデータグラムのハッシュ値の検査が行われる．mod_rsipdは，受け取ったアサインリクエストを許可するかどうかを判断するために用いられる．

4.2.2.1 アサインリクエストの処理

rsipdはrsipcからRSIPプロトコルによるアサインリクエストを受け取ると，これを許可するかどうかをmod_rsipdに問い合わせる．これを行うのが，rsipdのRSIP_Serverスレッドである．

さて，RSIPクライアントがアサインリクエスト送る場合，RSIPサーバに一番最初に送るのはREGISTER_REQUESTメッセージである．REGISTER_REQUESTメッセージを受け取ったRSIP_Serverスレッドは，メッセージの送信者をRSIPクライアントとして登録し，REGISTER_RESPONSEをRSIPクライアントに返す．

RSIPクライアントは，REGISTER_RESPONSEを受け取るとASSIGN_REQUEST_RSE_IPメッセージをサーバに送る．RSIP_Serverスレッドはこれを受け取ると，要求内容のデータグラムがファイアウォールのルールセットによって破棄されないかどうかをmod_rsipdに実装したシステムコールを用いて調べる．その結果，問題があった場合はRSIPクライアントにERROR_RESPONSEを返す．問題がなければ，RSIPクライアントからの要求内容とそこから求められるハッシュ値をアサインリストに追加し，RSIPクライアントにASSIGN_RESPONSE_RSE_IPを返す．

4.2.2.2 外部ホストへデータグラムを送信する際の処理

RSIPクライアントから外部ホストへのデータグラムの処理は，rsipdのRSIP_Gateway_Inスレッドが行う．この処理について説明する．

(1) RSIP_Gateway_Inスレッドは，データグラムを受け取ると，その送信元IPアドレスを検査する．そして，登録されたRSIPクライアント以外からのデータグラムであればこれを廃棄する．
(2) 登録されたRSIPクライアントからのデータグラムであった場合は，カプセル化で追加されたIPヘッダを取り除く．ただし，これにより得られたデータグラムが従来のRSIPによるアサインであった場合は，そのままデータグラムを外部ホストに送信し，(1)へ戻る．
(3) 次に，データグラムのIPオプションを検査する．IPオプションを持っていない場合，または，IPオプションに正しいハッシュ値が含まれていない場合は，データグラムを廃棄する．
(4) 最後に，データグラムを外部ホストに送信し，(1)へ戻る．

以上の処理により，RSIPサーバを超えることができたデータグラムは，正しいハッシュ値を持っていることが保証される．

4.2.2.3 RSIPクライアントへデータグラムを送信する際の処理

外部ホストからRSIPクライアントへのデータグラムの処理は，rsipdのRSIP_Gateway_Outスレッド

が行う．以下では，この処理について説明する．

(1) RSIP_Gateway_Out スレッドはデータグラムを受信すると，そのデータグラムに該当するアサインリストがあるかどうかを検査する．該当するアサインリストが見つからなければ，データグラムを廃棄する．

(2) 次に，データグラムが提案方式によるアサインであるかどうかを調べる．提案方式によるアサインでなければ，見つかったアサインリストに登録されている RSIP クライアントにデータグラムをトンネリングする．

(3) 提案方式によるアサインであった場合，提案方式による IP オプションを調べる．IP オプションが含まれていない場合，または，ここに含まれているハッシュ値が正しくない場合は，データグラムを廃棄する．

(4) 最後に，見つかったアサインリストに登録されている RSIP クライアントにデータグラムをトンネリングする．

以上の処理により，RSIP サーバによるデータグラムのフィルタリングが行われる．

4.2.3 外部ホスト

図 10 で示したように，外部ホストはカーネルモジュール mod_rsips から構成される．この mod_rsips は，受信したデータグラムの拡張に含まれるポート番号の復号と，ハッシュ値のチェックを行い，最後にポート番号の上書きを行う．また，データグラムの送信時には，暗号化されたポート番号，ハッシュ値をデータグラムに追加する．

なお，RSIP クライアントから受け取った IP オプションに含まれる情報（IP アドレス，ポート番号，送信時に使用するハッシュ値と暗号化されたポート番号，受信時に使用されるハッシュ値とポート番号，SPI など）を保持するため，外部ホストにはキーリストと呼ぶ記憶域を準備する．

4.2.3.1 データグラム受信時の処理

まず，外部ホストがデータグラムを受信する際の処理について説明する．

(1) 受信したデータグラムが ip_input に渡されると，これに提案方式による IP オプションが含まれるかどうかを調べる．持っていなければ，通常のデータグラムの処理を行う．

(2) 提案方式による IP オプションを持っていれば，次にキーリストに該当する送信元 IP アドレスを持つものがあるかどうかを調べる．該当するものがなければ，IP オプションからポート番号と乱数を復号し，ハッシュ値を生成する．ここで生成したハッシュ値と受け取ったハッシュ値が一致しなかった場合はデータグラムを廃棄する．ハッシュ値が一致した場合は，IP アドレス，ポート番号，ハッシュ値，暗号化した値，乱数などをまとめ，キーリストに追加する．

(3) キーリストに該当する送信元 IP アドレスを持つものがあった場合，キーリストに登録されたハッシュ値と受信したハッシュ値を比較する．これらが一致しなかった場合は，データグラムを廃棄する．

(4) 最後に，ESP を復号し，IP アドレス，ポート番号を上書きする．ただし，ポート番号を上書きしたあと，チェックサムの再計算を行う必要がある．以上の処理を行ったあと，tcp_input などの上位層の処理に移行する．

以上のように，外部ホストは受信した IP オプションからキーリストを作成し，そのキーリストを基にチェックを行うことで 2 回目以降の処理を高速化している．

4.2.3.2 データグラム送信時の処理

外部ホストがデータグラムを送信する際の処理について説明する．

(1) まず，ip_output に送信するデータグラムが渡される．このデータグラムは提案方式を用いて送信する必要があるかどうかを，キーリストに宛先 IP アドレス，ポート番号が一致するものがあるかどうかを調べることで判定する．提案方式を用いる必要がなければ，そのまま暗号化して送信する．

(2) 提案方式を用いる必要があれば，データグラムを暗号化した後に，提案方式による IP オプションを追加し，データグラムを送信する．

4.3 実装システムの検証

最後に，本実装システムの動作を妥当性を調べるため，実装システムの機能の検証と性能の検証を行う．

4.3.1 機能の検証

実装システムの機能の検証を行うために，提案方式によるアサインリクエストが RSIP サーバに受理された後，RSIP クライアントから以下の不正なデータグラムを外部ホストに向けて送信した．

(1) 提案方式による IP オプションを付けないデータグラム

(2) 不正なハッシュ値を IP オプションに含むデータグラム

(3) 不正なポート番号を暗号化した値を IP オプショ

ンに含むデータグラム

(4) ESPデータ内のポート番号を不正なポート番号に書き換えたデータグラム

まず，RSIPサーバで以上の条件を満たすデータグラムを受け取った場合の動作について検証する．実験の結果，条件(1)と条件(2)のデータグラムを廃棄し，それ以外の条件によるデータグラムは通過させていることが分かった．RSIPサーバでは，ハッシュ値のチェックのみを行うため，この動作は正常である．

次に，外部ホストで受け取ったデータグラムの処理について検証する．ただし，条件(1)と条件(2)のデータグラムはRSIPサーバによってフィルタされるため，条件(3)と条件(4)について検証を行う．その結果，条件(3)のデータグラムを受信した場合はデータグラムを廃棄し，条件(4)のデータグラムを受信した場合は，ポート番号を上書きすることを確認した．これにより，外部ホストの挙動も正しいことが示された．

4.3.2 性能の検証

提案方式による実装は，従来のRSIPにおいて暗号化されたデータグラムを送受信するRSIPSEC[9]に，ポート番号情報の暗号化やハッシュなどの処理を加えたものである．そこで，提案方式と元のRSIPSECについてその性能を比較する．性能の評価のため，外部ホストに構築したwebサーバからRSIPクライアントが100 MBのファイルをダウンロードした．なお，各ホストは100Base-TXで直接接続されており，RSIPクライアントと外部ホスト間のラウンドトリップタイムは0.235 msである．

実験の結果，平均スループットはRSIPSECで5.28 MB/sなのに対し，提案方式では5.10 MB/sとなり，平均スループットの低下はほとんど見られなかった．しかし，ラウンドトリップタイムはRSIPSECでは0.237 msなのに対し，提案方式では0.673 msと約3倍となった．これは，提案方式によりメモリ上での値の読み取り，比較，書きこみなどが増加したことが原因と思われる．ただし，このラウンドトリップタイムの増加はデータグラムが1ホップ程度の経路を通過する時間に相当し，実用では大きな障害とはならない．

5. まとめ

本論文では，RSIPサーバにおけるポート番号によるパケットフィルタリングを提案し，本システムを実装することでその評価を行った．その結果，平均スループットの低下がほとんど見られることなく，登録されたポート番号で通信を行うことを強制できることが示された．

本論文で提案した方式は，RSIPクライアントが外部ホストにアクセスする場合に限定されている．たとえば，RSIPクライアントが外部ホストにアクセスする場合は，アサインリクエストがRSIPクライアントから送られるため，外部ホストから送られてきたデータグラムの配送先をRSIPサーバが知ることができ，通信が成立する．しかし，外部ホストがRSIPクライアントにアクセスする場合は，RSIPサーバは受け取ったデータグラムをローカルネットワーク上のどのRSIPクライアントに送信したらよいのかを知らないため，データグラムをRSIPクライアントに送ることができない．しかし，ポート番号を使用したアドレス変換を行う状況では，本論文で対象としたネットワーク構成のように，内部ネットワークでプライベートアドレスを用いるのが典型である．このような設計の目的の1つとして，外部ネットワークから内部ネットワークへの直接的なアクセスを妨げることがあげられ，このような状況では，本論文による制約は問題にならない．

今後の課題としては，ラウンドトリップタイムの改善があげられる．本方式では，受信側が最後にポート番号を上書きしている．TCPやUDPでは，ヘッダのデータが変更されたときに，そのチェックサムを再計算する必要があるため，本実装ではデータグラムを受信したときはつねにこの再計算を行っている．これはかなり重い処理であるため，システムの大きな負担になっている．そこで，復号したTCPヘッダやUDPヘッダに含まれるポート番号が得られたポート番号と異なる場合は，データグラムを破棄するようにプロトコルを変更することで，チェックサムの再計算をなくし，ラウンドトリップタイムを改善できると思われる．

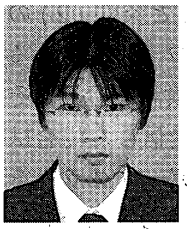
また，今回の実装ではポート番号情報などのハッシュ値を求めるためにMD5を，暗号化するためにDESを固定して使ったが，それ以外のハッシュアルゴリズムや暗号化アルゴリズムを使用できるように，柔軟性を持たせる設計にすることも検討している．しかし，4.1.2項で述べたように，ハッシュ値とポート番号情報などを暗号化した値はIPオプションに格納しているため，IPヘッダの制約からオプションの総バイト数を40バイトまでに抑える必要がある．たとえば，DESは与えられた平文を8バイト単位で暗号化する．このため，本方式で使用する乱数のサイズを1バイト大きくすると，IPオプションのサイズが現在の34バイトから42バイトに増加することになり，制限の40バイトを超えてしまう．この問題を解決するため，IP

データグラムのデータ領域にESPヘッダなどと同様にヘッダを置き，そこに必要なデータを書き込むという手法を考えている．なお，このIPオプションに関する制約はIPv4の場合にのみ存在し，IPv6の場合はこの限りではない．

参考文献

1) 馬場達也：マスタリングIPsec，オライリー・ジャパン (2001).
2) Kent, S. and Atkinson, R.: IP Encapsulating Security Payload (ESP), rfc2406.txt, IETF (1998).
3) Kent, S. and Atkinson, R.: IP Authentication Header, rfc2402.txt, IETF (1998).
4) Srisuresh, P. and Egevang, K.B.: Traditional IP Network Address Translator (Traditional NAT), rfc3022.txt, IETF (2001).
5) Aboba, B.: IPsec-NAT Compatibility Requirements, draft-aboba-nat-ipsec-04.txt, IETF (2001).
6) Borella, M., Lo, J., Grabelsky, D. and Montenegro, G.E.: Realm Specific IP: Framework, rfc3102.txt, IETF (2001).
7) Zaccone, C., T'Joens, Y. and Sales, B.: Address reuse in the Internet, adjourning or suspending the adoption of IP next generation?, *Proc. IEEE International Conference on Networks* (*ICON 2000*), pp.462–468 (2000).
8) Luo, J.-N. and Shieh, S.-P.: The multi-layer RSIP framework, *Proc. Ninth IEEE International Conference on Networks*, pp.166–171 (2001).
9) Montenegro, G.E. and Borella, M.: RSIP Support for End-to-end IPsec, rfc3104.txt, IETF (2001).
10) de Launois, C., Bonnet, A. and Lobelle, M.: Connection of extruded subnets: A solution based on RSIP, *Communications Magazine*, Vol.40, No.9, pp.116–121, IEEE (2002).
11) 石川大法，木村成伴，海老原義彦：RSIPサーバにおけるポート番号によるパケットフィルタリングの提案，コンピュータセキュリティシンポジウム2001論文集，pp.197–202，情報処理学会 (2001).
12) Perkins, C.: IP Encapsulation within IP, rfc2003.txt, IETF (1996).
13) Hanks, S., Li, T., Farinacci, D. and Traina, P.: Generic Routing Encapsulation (GRE), rfc1701.txt, IETF (1994).
14) Pall, G.S., Palter, B., Rubens, A., Townsley, M., Valencia, A.J. and Zorn, G.: Layer Two Tunneling Protocol "L2TP", rfc2661.txt, IETF (1999).
15) Borella, M., Grabelsky, D., Lo, J. and Taniguchi, K.: Realm Specific IP: Protocol Specification, rfc3103.txt, IETF (2001).

(平成15年 4 月 4 日受付)
(平成15年12月 2 日採録)

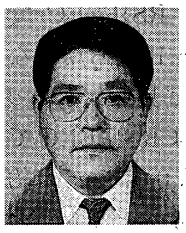


石川 大法

昭和52年生．平成13年立命館大学理工学部情報学科卒業．平成15年筑波大学大学院理工学研究科理工学専攻修了．同年三菱スペースソフトウェア（株）入社．現在に至る．ネットワークセキュリティの研究等に従事．

木村 成伴（正会員）

昭和42年生．平成7年東北大学大学院情報科学研究科情報基礎科学専攻博士課程後期3年の課程修了．同年筑波大学電子・情報工学系講師．平成12年同助教授．現在に至る．博士（情報科学）．プロセス代数，ネットワークプロトコル，通信システムの効率評価等に関する研究に従事．電子情報通信学会，ソフトウェア科学会，IEEE，ACM各会員．



海老原義彦（正会員）

昭和22年生．昭和50年東北大学大学院博士課程電子及通信工学専攻単位取得退学．同年同大学助手．昭和50年筑波大学電子・情報工学系助手，昭和51年同講師，昭和60年同助教授，平成5年同教授．平成10年から11年まで同大学学術情報処理センター長，平成12年から14年まで同大学電子・情報工学系長．現在に至る．工学博士．コンピュータネットワークアーキテクチャ，デジタル通信システムの性能評価，および知的通信システムの研究等に従事．電子情報通信学会会員．