社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＤＳＭ－33 (16)
2004／5／14

IPsecがストリーム伝送に及ぼす影響

浅野　歩　　岸田　崇志　　河野　英太郎　　前田　香織

広島市立大学大学院情報科学研究科　〒731-3194 広島市安佐南区大塚東3-4-1

E-mail:　{asano,takashi}@v6.ipc.hiroshima-cu.ac.jp,　{kouno,kaori}@ipc.hiroshima-cu.ac.jp

あらまし　ストリーム伝送においても，通信の安全性の向上への要求が高まりつつある．本研究では，暗号化や認証の一手法としてIPsecを用いる場合のストリーム伝送へ及ぼす影響について調査した．その結果，自動鍵交換によってパケットロスが発生することを明らかにし，IPsecのストリーム伝送への適用可能性について考察した．

キーワード　IPsec，ストリーム伝送，鍵交換，セキュリティ

Effects of IPsec on Stream Transmission

Ayumu ASANO　Takashi KISHIDA　Eitaro KOHNO　Kaori MAEDA

Graduate School of Information Science, Hiroshima City University

3-4-1 Ozuka-Higashi, Asa-Minami-ku, Hiroshima, 731-3194, Japan

E-mail:　{asano,takashi}@v6.ipc.hiroshima-cu.ac.jp,　{kouno,kaori}@ipc.hiroshima-cu.ac.jp

Abstract　There is much demand of secure transmission of streams. We investigate the effect of IPsec which is one of the techniques of encryption or authentication on stream transmission. Through some experiments, we show that packet losses occur by key exchange, and consider the possibility to use IPsec on stream transmissions.

Keyword　IPsec, stream transmission, IKE, security

1. はじめに

ネットワークの高速化に伴い高品質なストリーム伝送を行なう機会が増えてきており，テレビ会議，遠隔講義なども一般的になってきている．例えば，DVTS[1]や mpeg2ts[2]などの高品質な動画像伝送システムが遠隔講義や遠隔会議で使われている．しかし，多くの場合は通信内容に暗号化などの対策がされておらず，第3者による通信の盗聴などを防ぐことができない．盗聴を防ぐ手法の一つに暗号化があり，その手法の1つに IP 層でパケットの暗号化や認証を行なうIPsec[3]がある．アプリケーション層やトランスポート層で暗号化をする場合は，個々のプログラムで暗号化の機能を実装しなければならないが，IPsecを用いることで上位層のアプリケーションに変更を加えることなく，暗号化や認証の機能を追加することができる．そして，ストリームデータの暗号化の手段としても有効であると考えられる．しかし，IPsecをストリーム伝送に適用した場合のストリーム再生に及ぼす影響は十分に検討されていない．

本研究では，IPsecがストリーム伝送に及ぼす影響について調べるため，いくつかの IPsec が使用できる環境を構築し，ストリーム伝送の品質に大きな影響を与える帯域，遅延，パケットロスなどについて調査した．そして IPsec をストリーム伝送に適用させた場合の問題点と対策方法を明らかにし，IPsecのストリーム伝送への適用可能性について考察した．

本稿では，2章で IPsecのストリーム伝送への適用について述べ，次に3章で IPsecのストリーム伝送への影響に関する各種測定とその結果について考察する．また，4章で今後の課題について述べる．

2. IPsecのストリーム伝送への適用

IPsec は IP 層で暗号化，復号化をするため，既存のアプリケーションに暗号化機能を追加することなく利用できるという利点がある．例えば，DVTS や mpeg2ts などの動画像転送システムは暗号化機能を持たないが，

IPsec を用いることでプログラムを拡張することなく暗号化された通信が可能である．また，トンネルモードを用いることで ViewStation[4] など暗号化機能を持たないテレビ会議システムでも通信の暗号化可能となる．

IPsec で使われる暗号化アルゴリズムは DES, 3DES, AES などがよく使われる．これらのアルゴリズムは時間をかけたり，セキュリティホールが見つかることで解読される可能性があり，盗聴された場合は絶対に安全であるとはいえない．しかし，一般に IPsec で使われているアルゴリズムはかなりの強度があり，個人のテレビ電話，学校間の遠隔交流，講義の配信などにおいて，IPsec をストリーム伝送の暗号化に用いることにより，通信の安全性がかなり向上する．そして，今後はストリーム伝送に IPsec を用いる場面も増えると想定している．

ただし，IPsec は IP パケットに対して暗号化や認証をするため処理負荷が増加する．そこで，IPsec のストリーム伝送への影響について調査しておくことは重要と考えている．

3. IPsec のストリーム伝送への影響

3.1. 測定項目と実験環境

数種類の IPsec の実装を用意して最大転送速度，遅延，パケットロスを計測した．IPsec の実装は，ソフトウェアの実装として FreeBSD 4.8 上に KAME IPsec，Red Hat Linux 9 上に FreeS/WAN を拡張した USAGI IPsec，ハードウェアの実装として暗号処理プロセッサを搭載した Allied Telesis 社 AR450S の 3 種類を調べた．IPsec の動作モードは USAGI IPsec がトランスポートモード，AR450S がトンネルモード，KAME IPsec は両方について調べた．これらを比較することで，個々の実装の評価とともに，ソフトウェアでの実装の違い，ソフトウェアとハードウェアの実装の違い，IPsec 動作モードの違いについて大まかな傾向を知ることができる．測定に用いた機器は図 1 のように接続した．FreeBSD と Linux はパーソナルコンピュータに構築した．これらはスイッチやクロスケーブルで直接つなぐことで，できるだけパケットロスや遅延が少ない状態にした．どちらの IPsec 動作モードも，IPsec の処理をする PC は同じ仕様になるようにしてある．トランスポートモードのホスト 1,2 と，トンネルモードの GW1,2 は同じマシンであり，IPsec の処理をする．各種測定は両モードともホスト 1,2 上で行なう．GW1,2 は IPsec ゲートウェイに相当し，AR450S を用いた測定時には図中の GW1,2 として AR450S を接続した．実験に用いた PC の仕様を表 1 に示す．

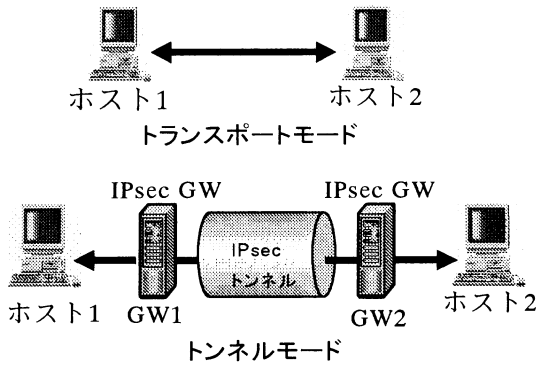

図 1．実験時の PC の構成

表 1．実験に用いたマシンの仕様

動作モード	ホスト名	CPU	Memory
トランスポート	ホスト 1,2	Pentium4 2.6GHz	512Mbyte
トンネル	ホスト 1,2	PentiumIII 1GHz	512Mbyte
	GW1,2	Pentium4 2.6GHz	512Mbyte

3.2. 最大転送速度

現在ある IPsec の実装でどれくらいの帯域のストリーム伝送が可能であるかを調べるため，Iperf[5] を用いて最大転送速度を求めた．Iperf は IPv4 と IPv6 の両方に対応しており，帯域，パケット長，測定時間を指定して擬似的にストリーム伝送の状態を作りだすことができる．パケット長はフラグメントが発生しない適当な大きさとして，トランスポートモードは 1300byte，トンネルモードは 1100byte に設定した．IPsec の設定は暗号化に 3DES，認証に MD5 を用いた場合を計測する．Iperf を用いて 1Mbps 単位で帯域を変化させ，各帯域で 30 秒～数分連続してパケットを送り，パケットロスが発生し始めるまでの帯域を調べる．結果を表 2 に示す．

表 2．最大転送速度

	トランスポート		トンネル	
実装	FreeBSD	Linux	FreeBSD	AR450S
通常時[Mbps]	93	93	92	94
IPsec 時[Mbps]	54	46	57	47※

表 2 より，IPsec を用いることで，最大転送帯域は大幅に低下していることがわかる．これ以上の帯域で

※ 実測値であり，カタログ値等詳細は http://www.allied-telesis.co.jp/products/list/router/ar450s/catalog.html を参照

ストリームを送るとパケットの暗号化，認証の処理が間に合わないため，処理能力の限界を超えてパケットロスが発生する．DV であれば 1 ストリーム，MPEG2 であれば数ストリームが転送できる限界である．トンネルモードを使って双方向でストリーム伝送を行う場合には，IPsec ゲートウェイに大きな負荷がかかるため，表 2 で示した IPsec 時の最大転送速度よりも下がると予想される．例えば，DVTS を双方向で用いる場合は約 60Mbps の帯域が必要であり，今回調べた IPsec の実装の暗号化処理能力では難しいと考えられる．解決方法として，DVTS の送信，受信するマシンそれぞれにトンネルモードをする機器を接続する，もしくトランスポートモードを用いる．しかし，前者の場合は IPsec の処理をする機器が余分に必要になり，後者の場合はストリームの再生に加えて IPsec の処理が余分に必要になることを考慮する必要がある．

3.3. 遅延

ストリーム伝送時の遅延時間の測定では，ストリームの帯域や IPsec のアルゴリズムを変えた場合の遅延時間の変化を求める．遅延は ping を用いて RTT 値を測定する．ping は 1 秒間隔で実行し，60 回測定した平均を求める．また，ICMP エコーリクエストのペイロード長を調整することで，IPsec の処理に渡すデータ長を 100byte の倍数になるようにし，フラグメントしないサイズまで大きくしていく．Iperf による負荷トラフィックは 10Mbps と 30Mbps，パケット長はトランスポートモードが 1300byte，トンネルモードが 1100byte である．IPsec の設定は，暗号化は DES と 3DES の 2 種類を測定する．また，認証などの設定は 2 つの動作モードでできるだけ同じになるようにした．設定した動作モード，セキュリティプロトコル，認証アルゴリズムは表 3 の通りである．

表 3．設定項目

動作モード	セキュリティプロトコル	認証
トランスポート	ESP と AH	MD5
トンネル	ESP	

ストリームの帯域と暗号化のアルゴリズムを変えた場合の一例として，図 2 に KAME FreeBSD でトランスポートモードを用いた場合の結果を示す．

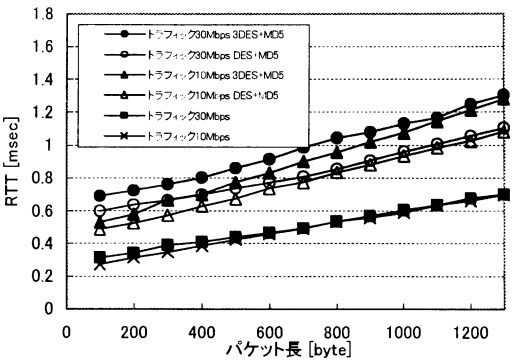

図 2．KAME トランスポート時の RTT

図 2 から IPsec によって RTT が通常時より大きくなっていることが分かる．3DES と DES の暗号化アルゴリズムの違いで RTT に大きな差はないが，ストリームの帯域によって RTT が大きくなっていることが分かる．このような傾向は他の実装でも同じであった．3 つの実装の比較として，暗号化に 3DES，認証に MD5 を用いた場合の RTT の測定結果を図 3 に示す．

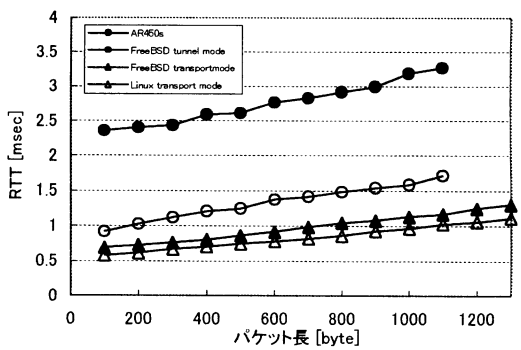

図 3．各実装の RTT の比較

AR450S の RTT が一番大きい結果となったが，どの実装でもベストエフォートのインターネット利用時の転送遅延やストリームデータの処理遅延と比較すると十分に小さいと考えられる．しかし，広域 Ethernet などでイントラネットを構成している場合など遅延が数 msec のネットワークも出てきており，そのような環境では IPsec による遅延の増加は無視できない．

3.4. 鍵交換による影響

3.4.1. 鍵交換の影響によるパケットロス

鍵交換がない状態で IPsec をした場合は，表 2 の IPsec 時の帯域以下であればパケットロスは発生しない．しかし，ストリーム伝送中に鍵交換が発生した場

合には鍵更新の処理負荷や鍵の切り替わりにより，ストリーム伝送に影響を及ぼす可能性がある．そこで，ストリーム伝送中に適当な間隔で鍵交換し，その間のパケットロスを測定する．ここではKAME IPsecをトランスポートモードで動作させた場合を示す．接続は図1のトランスポートモードと同じようにした．パケットロスの測定にはIperfを用い，帯域は30Mbps，パケット長は1300byteに設定した．そして，1秒間隔でパケットロス率を出力させながら，1200秒間測定する．IPsecの設定は，暗号化に3DES，認証にMD5を用い，鍵の有効期限を60秒とした．また，鍵交換はPFS(Perfect Forward Secrecy)を有効にすることでDH(Diffie-Hellman)鍵交換によって鍵を生成する．鍵の強度に関わる素数の大きさは1024bitを用いるDHグループ2に設定した．

鍵の有効期間が近づいて鍵更新をする場合，どちらかが先に鍵更新をするためのメッセージを相手に送る．先に鍵交換を要求する方をinitiator，受け答える方をresponderという．KAME IPsecの鍵交換デーモンracoonは設定によって自分から鍵交換を要求しないようにすることができる．ホスト2にこの設定をすることで，ホスト1が常にinitiatorとなって鍵の更新を開始するようにする．この状態で片方ずつからストリームを流すことで，initiatorかresponderかの違いを調べることができる．

IPsecで用いられる鍵には有効期限があるが，完全に有効期限が切れる前に新しい鍵を生成する仕組みになっている．新しい鍵が生成されると，古い鍵の有効期間が経過するまで2種類の鍵を持っている状態がある．この状態ではどちらかの鍵を選択することになるが，USAGI IPsecとAR450Sは新しい方の鍵を使うようになっている．KAME IPsecは設定によって新しい鍵を使うか古い鍵を有効期限まで使い続けるかを選択できる．そこで，鍵の選択時に新しい鍵を使用する設定と古い鍵を使用する設定の両方について測定する．図4〜7は，ストリーム伝送時のパケットロスの状況を示したものである．図中のinitiator→responderとはinitiator側からresponder側方向へストリームを送っていることを意味する．

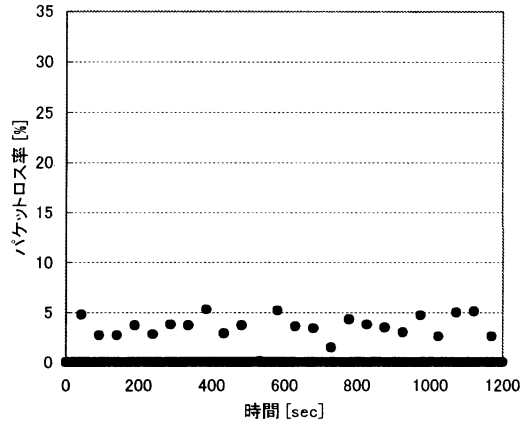

図4．新しい鍵を使用，initiator→responder

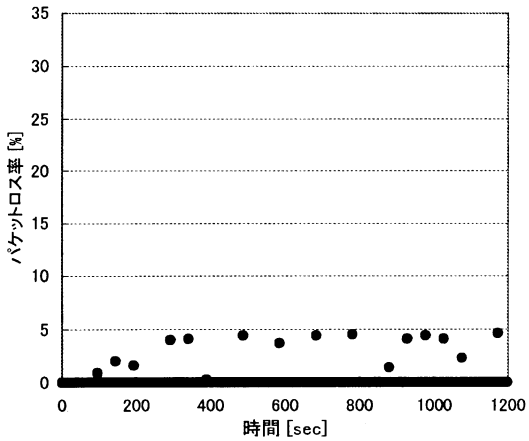

図5．新しい鍵を使用，responder→initiator

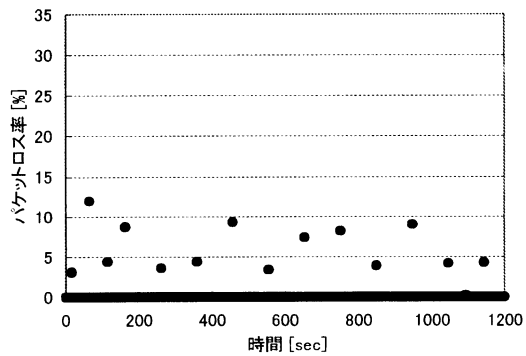

図6．古い鍵を使用，initiator→responder

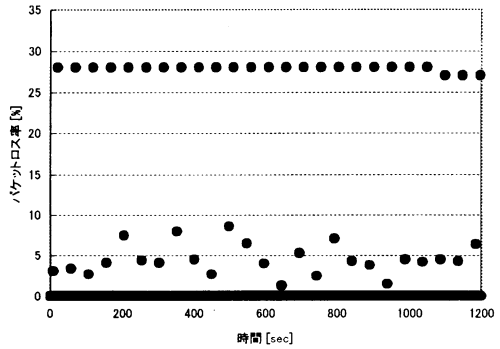

図 7. 古い鍵を使用，responder→initiator

図 4,5 は新しい鍵を使う設定である．initiator，responder に関係なく，ほぼ定期的に数%のパケットロスが発生している．この原因は鍵交換であると推測できる．図 6,7 は古い鍵を使い続ける設定であるが，initiator→responder 方向は図 4,5 の新しい鍵を使う設定の場合と同じようなパケットロスの傾向が見られるが，responder→initiator 方向では 30%弱のパケットロスが定期的に発生している．数%の割合で発生しているパケットロスは鍵交換時に起こっているが，30%弱のパケットロスは古い鍵の有効期限が経過して新しい鍵を使い始めるタイミングが取れていないことが原因である．これらの詳細については 3.5.2 と 2.5.3 で考察する．

USAGI IPsec と AR450S についても同様の測定を行った．これらは initiator と responder を固定化する設定ができないため，initiator と responder 間のストリーム伝送の方向とパケットロスの関係は特定できないが，鍵交換時に図 4,5 と同様のパケットロスが発生している．二者を比較すると，AR450S は他の実装に比べるとパケットロスが少なかった．

3.4.2. 鍵交換時のパケットロス

DH を用いた鍵生成は多くのリソースを消費する．そこで，PFS を無効にすることで DH を用いずに鍵生成をするように設定して同様に測定したところ，パケットロスは発生しなかった．鍵交換をしているときのパケットの振る舞いをより詳しく調べるため，10msec 間隔で UDP パケットを送信するプログラムを作成し，送信側で Ethereal を用いて鍵交換処理前後のパケットの時間とシーケンスナンバーを測定してグラフ化した．パケットは initiator 側から responder 側の方向へ送信した．

図 8 は PFS を無効にした場合，図 9 は PFS を有効にして鍵の生成に 768bit の素数を用いる DH グループを 1 にした場合，図 10 は PFS を有効にして DH グループを 2 にした場合である．DH グループ 1 と 2 では 2 の方が鍵の生成に使う素数の数が大きいため安全に鍵が生成されるが，より多くのリソースを消費する．また，古い鍵を使い続ける設定にすることで，新しい鍵ができても鍵の切り替えによるパケットロスが発生しないようにし，問題が鍵交換処理によるものか，鍵の切り替えによるものかを切り分けた．

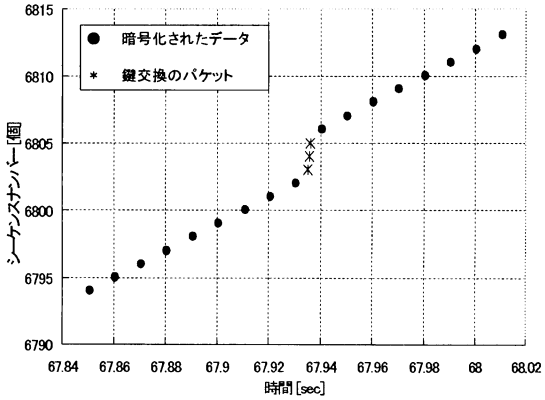

図 8. PFS 無効

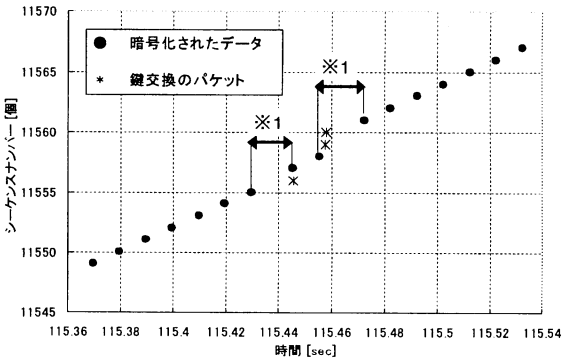

図 9. PFS 有効　DH グループ 1

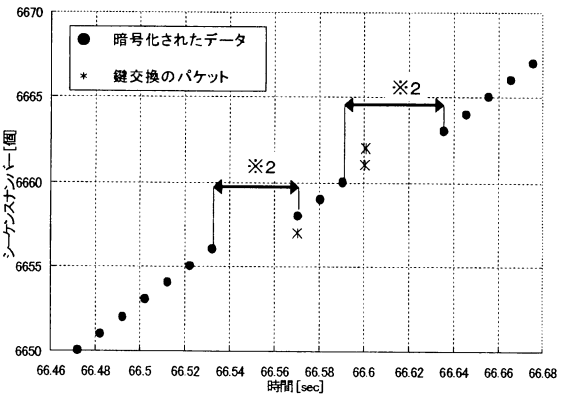

図 10. PFS 有効　DH グループ 2

鍵交換処理は3つのパケットが交換される．図8はPFSを無効にした場合で，短時間で鍵交換が完了しており，パケットの送信間隔に影響は見られない．図9,10はPFSを有効にした場合で，※1,2のように鍵交換処理前後でパケットが送信できていない時間がある．また，図10の※2の方がパケットを送信できていない時間が広いことから，DHによる鍵生成処理負荷によってリソースを消費し，その間はパケットが送信できない状態になっていると考えられる．今回作成したUDPパケットを送信するプログラムは1秒間に100回，数十byteのパケットしか送信しないため，通常のストリーム伝送に比べると負荷はかなり小さい．そのためパケットロスは発生していなかったが，より大きな帯域を伝送した場合にパケットロスが発生する可能性も十分に考えられる．ストリーム伝送時には鍵交換をしない，もしくは鍵交換をする場合でもPFSを無効にすることでパケットロスの発生を抑えることができるが，PFSを有効にした場合に比べて安全性は低下する．

3.4.3. 鍵の有効期限のズレによるパケットロス

図7のように，新しい鍵ができても古い鍵を使い続ける設定にしてresponder→initiator方向にパケットを送った場合，鍵交換処理以外の部分で大きなパケットロスが発生していた．原因を調べるため，両ホスト間の鍵の有効期限の関係について調べた．両ホストで所持している鍵の関係を図11に示す．横線は鍵の有効期限を表す．例えばt1～t6がresponder側の古い鍵の有効期限である．斜線はパケットの送信を意味し，太い斜線は古い鍵，細い斜線は新しい鍵を用いていることを表す．initiator側がt5で古い鍵が削除されているのに対して，responder側はt6に古い鍵が削除されている．

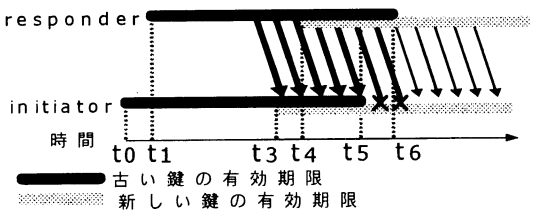

図11．鍵の同期不良の問題

t5以前はresponder側は古い鍵を用いてデータを暗号化しており，initiator側もそれを復号化する鍵を保持しているため問題ない．t5を経過するとinitiator側は古い鍵の有効期限が切れるが，responder側は古い鍵の有効期限がt6なので，古い鍵のままパケットを送信する．しかし，initiator側はそのパケットに対する鍵を所持しておらず，復号化することができない．t5～t6間のパケット伝送時にパケットロスが発生する．t6以降はresponder側は新しい鍵を使い始めるため，正しく復号化できる．鍵の有効期限のズレは，鍵が生成される時間がinitiator側とresponder側とで違うのが原因だと考えられる．また，常に新しい鍵を用いる設定にしておけばこの問題は発生しない．

3.4.4. 鍵更新が頻繁に発生する場合の考察

3.5.1で示したように，DH鍵交換を用いた鍵生成は現在の実装ではパケットロスが発生する．そのため，鍵交換が頻繁に発生するとストリームの質が劣化することになる．鍵交換が頻繁に発生する可能性として，例えばメンバーの参加や離脱のたびに鍵更新が要求されるようなグループ間通信のストリームデータの暗号化がある．IPsecを用いてグループ間通信をする場合の鍵の配信，更新の手法としてGDOI(The Group Domain of Interpretation)[6]が提案されている．DH鍵交換は2者間だけの共通鍵しか生成できないため，GDOIではDH鍵交換を用いずに他の鍵でグループ共通の鍵を暗号化することで，鍵の配信や更新を実現している．そのため，DH鍵交換を用いた場合のようにパケットロスは発生しないと考えられるが，詳しいことは調査中である．

4. まとめ

本稿では，IPsecがストリーム伝送に及ぼす影響について各種測定を行い，IPsecの適用可能性について調査した．特に，鍵交換によるパケットロスについてはその原因について詳しく考察した．今後はパケットロスの少ない鍵交換や，グループ鍵配信について検討する．

謝辞

本研究の一部は広島市立大学特定研究費(平成15年度3207)と通信・放送機構平成15年度受諾研究「工業高校における IPv6 を用いたロボット遠隔操作環境実現とロボット遠隔操作実験および総合的情報家電模型操作環境実現と遠隔操作実現に関する研究開発」の支援を受けて実施されている．ここに記して感謝の意を示す．

文　献

[1] 杉浦一徳，小川晃通，中村修，村井純，"民生用DVを用いたインターネットビデオ会議システム"，情報処理学会学会誌，Vol.40，No.7，1999.

[2] 近堂徹，西村浩二，相原玲二，前田香織，大塚玉記，"高品質動画像伝送におけるFECの性能評価"，情報処理学会論文誌，Vol.45，No.1，pp.84-92，2004.

[3] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", RFC2401，1998.

[4] Polycom, http://www.polycom.com.

[5] Iperf, http://dast.nlanr.net/Projects/Iperf.

[6] B. Weis, T. Hardjono, H. Harney, "The Group Domain of Interpretation", RFC3547, 2003.