iSCSI ストレージアクセスにおける安全な通信を行うシステムソフトウェアの検討

神坂 紀久子 † 山口 実靖 ‡ 小口 正人 †

低価なコストでストレージ統合を可能にする IP ネットワークストレージの実現技術として，iSCSI プロトコルが注目を集めている．iSCSI では TCP/IP ネットワークを介して遠隔ストレージにアクセスする場合があるため，セキュリティが重要な課題となる．iSCSI においては強力な暗号化と認証機能を提供する IPsec を用いることが可能であるが，暗号化処理による性能低下と CPU に対する負荷が問題となっている．
本稿では，iSCSI ネットワークを利用した安全なストレージアクセスを行うため，暗号化，復号化処理を IPsec 層より上位層で行うことによって，性能を向上する手法を提案する．まず，iSCSI を用いない環境において，単純なソケット通信でシーケンシャルリードアクセスの実験を行い，IPsec を用いた通信より，提案手法の方が大幅に性能が向上することがわかった．次に，iSCSI ネットワークで，提案手法を模擬したシステムを用いて，IPsec との性能比較実験を行い，さらに，TCP パケット転送の振る舞いの可視化を行った．

A Proposal of System Software in Secure Storage Access using iSCSI

Kikuko Kamisaka† Saneyasu Yamaguchi‡ Masato Oguchi†

iSCSI protocol, used in building IP-based storage network, is becoming more important because it realizes consolidation of storage at low cost. Since one of a key issues in iSCSI is a security measure to access remote storage via IP networks, it can employ IPsec, which offers strong encryption. However, IPsec encryption increases processing load on the server's CPU and degrades its performance.
In this paper, for realizing secure storage access efficiently on iSCSI networks, we propose the idea of encryption scheme in the higher-level layer instead of IPsec encyrption scheme. We measure the performance on simple socket communication and on iSCSI communication using the proposed model, and compare our proposed scheme with IPsec. Consequently, our proposed method of encryption in the higher-level layer outperforms that of encryption using IPsec. Moreover, The behavior of TCP layer is analyzed by visualizing TCP packet transfer.

1. はじめに

データ量と管理コストの急増にともない，ストレージ群とサーバ群を高速なネットワークで接続し，ストレージ統合と集中管理が可能な SAN(Storage Area Network) に関心が高まっている．現在 SAN の構築技術として，FC(Fibre Channel) とよばれる専用ネットワークが主流を占めているが，Gigabit Ethernet などの広帯域な LAN 環境の普及により，膨大なデータを IP ネットワークを通して遠隔のストレージに格納することが可能になった．この IP-SAN の技術により，管理が容易になり，導入や運用コストを抑えることができる．

主要な IP-SAN のプロトコルとして，現在で

† お茶の水女子大学大学院 人間文化研究科
Graduate School of Humanities and Sciences,
Ochanomizu University
‡ 東京大学生産技術研究所
Institute of Industrial Science,
The University of Tokyo

は 2002 年 2 月に IETF により正式承認された iSCSI(Internet SCSI) プロトコルが期待されている [1][2]．しかし，iSCSI はまだ新しい規格であり，通信の性能に関して課題を残している．iSCSI では，TCP/IP ネットワークを介して通信を行うが，TCP は高速な通信インフラを想定して設計されたプロトコルではないため，広帯域のネットワークでストレージアクセスを行う際には性能が問題となる場合がある．また専用回線を用いる場合と比べ，オープンであるインターネットを介するため，セキュリティが重要な課題となる．iSCSI では IP ネットワークにおいて信頼性があり強力なセキュリティ技術である IPsec を利用することができるが，IPsec 層における暗号化や復号化処理により，サーバの CPU に負荷を与え，処理の性能を低下させる．これは，暗号化処理の計算量が多いことだけでなく，IP 層と同位の IPsec 層において，小さなサイズに分割されたデータブロックごとに暗号化処理を行うことが影響していると考えられる．

そこで本稿では，iSCSI ネットワーク環境において，安全な通信を行うため，IPsec 層で行われていた暗号化を，IPsec 層より上位層で行うことにより，効率的に暗号化と復号化を行う手法を提案する．

まず，基礎実験として，iSCSI を利用しない通常の IP ネットワーク環境において，単純なソケット通信を行うことにより，提案手法と IPsec を利用した場合の性能を比較する．その結果，提案手法により，スループットが約 30%上昇することを確認した．さらに，提案手法の模擬システムを試作し，iSCSI を使用したストレージアクセスの，性能評価実験を行い，スループットが 17%ほど向上した．

また，各実験の CPU 負荷の点においても，ユーザモードで処理される提案手法の方がカーネルモードで処理される IPsec を利用する方式より，負荷が低くなることがわかった．

さらに，iSCSI ストレージアクセスにおける，TCP 層での実際の振る舞いを確認するため，TCP パケット転送の様子を可視化した．

2. 研究背景

2.1 iSCSI

現在では，ストレージにアクセスする方法として，従来利用されていたサーバの I/O バスにストレージを直結する DAS(Direct Attached Storage) から，SAN に移行しはじめている．SAN は，サーバ群とストレージデバイス群を接続する高速な専用のネットワークであり，ストレージの統合と集中管理により，容易で効率的なストレージ管理が実現され，また遠隔ストレージによる非常災害対策も可能である．

SAN の構築手法として，現在では，FC とよばれるネットワーク技術を使用した FC-SAN が広く普及している．しかし，FC は，接続距離に限界があり，スイッチやホストバスアダプタなどのハードウェアが高価で，FC を管理する技術者が少ないなどの問題があった．最近では，Gigabit/10Gigabit Ethernet の出現により，ネットワークが広帯域化したことから，FC ではなく，IP ネットワークを介した SAN 接続を可能にする IP-SAN が提案されている．IP-SAN では，FC-SAN と比べ，汎用性のある TCP/IP を利用していることから，接続距離の限界がなく，既存の IP ネットワークとのシームレスな統合ができ，安価なコストで導入や運用管理を実現する．

IP-SAN で用いられる主要なプロトコルとして，2003 年に IETF に承認されたブロック単位でストレージアクセスを行う iSCSI が注目を浴びている．iSCSI では，SCSI コマンドを TCP/IP パケットにカプセル化することで，サーバ (イニシエータ) とストレージシステム (ターゲット) を IP ネットワーク経由で接続する．iSCSI では，標準の SCSI コマンドを利用できるため，ユーザはサーバに直接接続されたローカルディスクと同じように，遠隔の SCSI デバイスを操作することが可能になる．

しかし，TCP は高速な通信インフラを想定して提案されたプロトコルではないため，性能面の問題が予想される．さらに，iSCSI でストレージアクセスを行う際には，サーバの CPU に大きな負荷をかけるという問題が指摘されている．

2.2 iSCSI における IPsec の適用と問題点

IP ネットワーク経由でストレージにアクセスする iSCSI では，セキュリティ対策が重要な課題となる．その対策として，iSCSI ではインターネットで広く使われ信頼性のあるセキュリティ技術である IPsec がサポートされている．IPsec は，IP パケットに対してネットワーク層で強固な暗号化と認証機能を提供する．

IPsec では，暗号方式として安全性の高い 3DES(Triple Data Encryption Standard) が一般に広く利用されている．しかし，3DES の暗号化処理は計算量が非常に多いため，性能低下につながり，高い CPU 負荷の原因となる．

iSCSI ネットワーク環境における，IPsec を利用

したシーケンシャルリードアクセスの性能評価実験から，暗号化を行わない通常のアクセスにおいては，ブロックサイズの増加にともない，スループットが上昇するのに対し，IPsec を用いた場合は，一貫して大幅な性能低下と著しく高い CPU の負荷が確認されている [3]．性能劣化の主要な要因と考えられる TCP 層に着目して，時間軸上に TCP パケットの可視化を行った解析の結果 [4] によると，IPsec による暗号化を行わないアクセスでは，SCSI コマンドによって要求されたブロックサイズ分の多数の TCP パケットが，ほぼ同時刻にターゲットからイニシエータに転送されていた．しかし，IPsec を用いることによって，各 TCP パケットの送信間隔が暗号化処理により拡大され，これが全体の性能劣化に影響していることがわかった [5]．

2.3 関連研究

iSCSI の関連研究としては，Wee Tech Ng らによる iSCSI の性能に関する研究 [6] や，P. Sarkar らによる iSCSI のソフトウェア実装とハードウェア実装の比較に関する研究がある [7][8]．また，P. Radkov らは，iSCSI と NFS の性能比較を研究している [9]．セキュリティを考慮した研究では，S-Y. Tang らによって iSCSI における IPsec と SSL の性能が比較されている [10]．しかし，現在までセキュリティ技術を利用した iSCSI の性能向上に関する研究はまだ十分になされていない．

3. 上位層における暗号化方式の提案

セキュリティとパフォーマンスの間には基本的にトレードオフの関係があるため，安全にデータを転送するためには，双方に適切な対策をとることが重要である．iSCSI ネットワーク環境において IPsec による安全なストレージアクセスを行う際，暗号化や復号化処理により，性能がある程度低下することは当然であるが，効率的に暗号化を行うことにより，通常の iSCSI 通信と遜色のない性能が得られる可能性がある．

しかし，iSCSI 通信で一般に使用されている IPsec を用いる場合には，IPsec は TCP などの上位層における処理を知ることができないため，効率的な暗号化処理を行うことは困難である．大規模なデータをシーケンシャルにアクセスする場合には，IPsec は上位層である TCP 層などから，データブロックを受け取った時点で，順次暗号化処理を行い，そのまま下位層に渡す．たとえば，ストレージからサーバにデータが転送され，TCP 層でその ACK を待っている間は，IPsec 層では何も処理が行われない．また，下位層である IPsec 層に渡されるまで上位層で分割された小さなパケットごとに暗号化処理を行うため，効率的な処理をしているとはいえない．

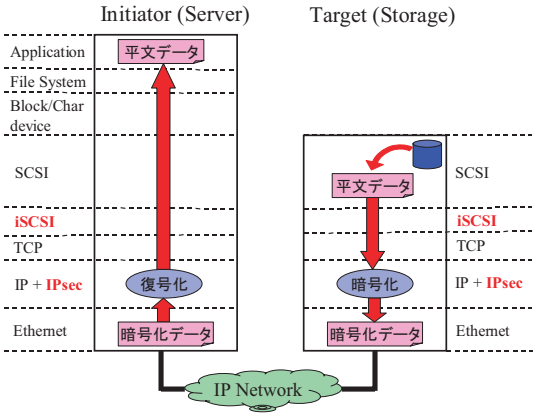

図 1: IPsec を用いて暗号化するストレージアクセス方式

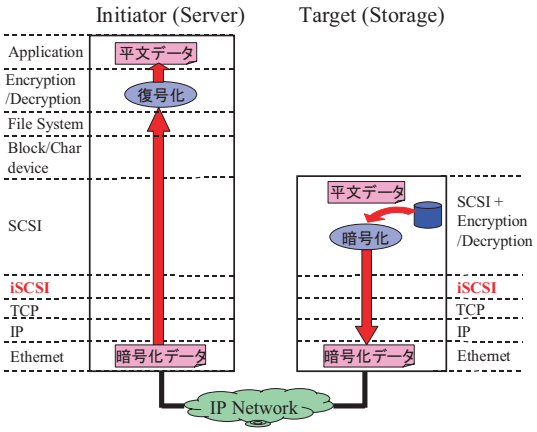

図 2: 上位層において暗号化するストレージアクセス方式 (提案手法)

そこで本稿では，安全な iSCSI ストレージアクセスにおける性能を向上するため，従来の IPsec 層で暗号化を行う代わりに，より上位層で暗号化，復号化処理を行う手法を提案する．

iSCSI ネットワークにおいて，IPsec で暗号化を行うシーケンシャルリードアクセス方式と，IPsec を使わずに上位層で暗号化を行う場合の実現方式をそれぞれ図 1，2 に示す．IPsec を利用した方式 (図 1) では，SCSI 層でターゲットのディスクに格納されているデータが読み出され，iSCSI 層，TCP 層を通り，データが小さいブロックに分割される．分割されたデータはその後 IPsec 層で暗号化処理が行

われる．復号化も同様に IPsec 層で行われた後，イニシエータのアプリケーションに渡される．

一方，提案手法を用いた場合 (図 2) は，ターゲットのディスクに格納されたデータは，上位層である SCSI 層と同位の暗号化/復号化層で暗号化される．まとまったブロックを上位層で暗号化するため，IPsec でより小さなパケットごとに暗号化を行うよりも効率的であると考えられる．提案手法において，イニシエータから読み込み要求を受け取った時点で随時暗号化を行う機能を実装した場合，TCP 層で ACK を待っている間にも次のパケットの暗号化処理を行うことが可能である．暗号化処理時間を通信処理の待ち時間に隠蔽することによって，効率的な暗号化を行うことができ，性能を向上させる手法として有効である．

4. 提案方式の性能評価実験

本稿では，提案手法を実装した際の性能を予測するために，IPsec を用いる方式と上位層で暗号化する方式の性能の比較実験を行い，提案手法を実装した際の評価を行った．まず，基礎実験として，iSCSI を利用しない単純なソケット通信を用いたシーケンシャルリードアクセスの性能を測定した．次に，iSCSI ネットワークにおいてリモートストレージアクセスを行った場合も同様に性能を評価した．

4.1 実験環境

実験に用いたシステム環境を表 1，2 に示す．各実験において，サーバ (イニシエータ) とストレージ (ターゲット) を Gigabit Ethernet スイッチで接続する．どちらの手法においても，暗号化アルゴリズムとして 3DES を用いている．提案手法の上位層における暗号化には，インターネットの標準であるさまざまな暗号化アルゴリズムを実装している OpenSSL の crypto ライブラリ [11] を用いた．iSCSI の実装には，ニューハンプシャー大学 InterOperability Lab[12][13] が提供している実装を用い，IPsec の実装には，Linux において広く利用されている FreeS/WAN[14] を用いている．

また UNH-iSCSI の Linux 実装の問題により，通常より大幅に小さいサイズ (0.5KB 程度) の微小 iSCSI PDU(Protocol Data Unit) が発行されてしまうことによる著しい性能低下が確認されている [5]．この微小パケットにより，TCP の Nagle アルゴリズムが起動され，遅延 ACK との相互影響が著しい性能劣化の一因であることがわかっている．その性能劣化を軽減するため，本実験では Nagle アルゴリズムを停止する Linux の TCP_NO_DELAY オプションを用いた．

表 1: 実験環境：使用実装

OS	initiator : Linux 2.4.18-3 target : Linux 2.4.18-3
CPU	Intel Xeon 2.4GHz
Main Memory	512MB DDR SDRAM
HDD	36GB SCSI HD
NIC	Intel PRO/1000XT Server Adapter on PCI-X (64bit, 100MHz)

表 2: 実験環境：使用実装

iSCSI	UNH-iSCSI Initiator and Target for Linux ver. 1. 5. 3
IPsec	FreeS/WAN ver. 2.01

4.2 単純なソケット通信によるシーケンシャルリードアクセスの実験

まず，上位層で暗号化する手法の性能への影響を評価するため，基礎実験として，iSCSI を用いない環境における単純なソケット通信によるシーケンシャルリードアクセスの性能を測定した．

IPsec を使用する従来の暗号化方式と，提案手法の暗号化方式のどちらにおいても，ターゲット側に相当するホストのディスクからデータが読み出され，IP ネットワークを経由して転送され，イニシエータ側に相当するホストがこれを受信する．その際，IPsec を用いる方式 (図 3) では，上位層であるアプリケーションはデータの送受信だけを行い，下位層として IPsec を起動して，IPsec 層において暗号化，復号化を行う．上位層において暗号化する方式 (図 4) では，ターゲット側では，上位層であるアプリケーションがディスクからデータを読み出した後に crypto ライブラリから暗号化処理を呼び出すことによって暗号化を行い，イニシエータ側も，同様に上位層 (アプリケーション) でデータを受け取ってから復号化を行う．

4.3 iSCSI ネットワークにおけるリモートストレージアクセスによる実験

次に，iSCSI ネットワークでイニシエータからターゲットの RAW デバイスに対してシーケンシャルリードアクセスを行い，性能を測定した．本実験では，上位層で暗号化，復号化を行うことによって，通信の待ち時間の間に次のパケットを暗号化するなどの

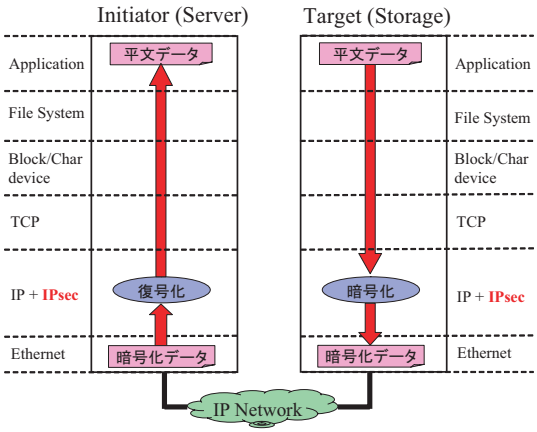

図 3: IPsec を利用した単純なソケット通信を用いたシーケンシャルリードアクセス

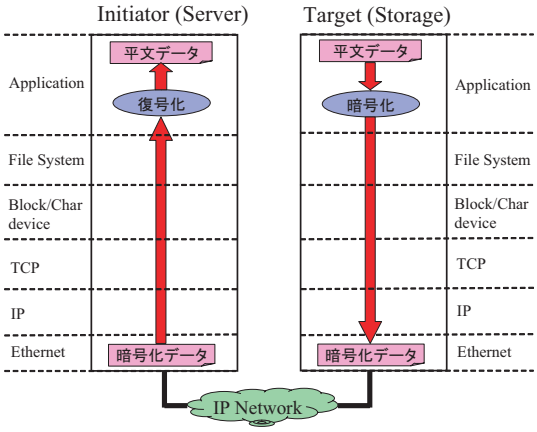

図 4: 提案手法による単純なソケット通信を用いたシーケンシャルリードアクセス

機能を実装した際の有効性を評価するため，提案手法を一部実装した試作システムを用いている．

IPsec を用いた場合は，IPsec を起動して，通常の iSCSI を用いたストレージアクセスを行う (図 1)．ターゲットのディスクに格納されているデータが読み出されると，下位層にある IPsec 層で暗号化されてイニシエータに送信される．イニシエータでは同様に IPsec 層で復号化を行い，アプリケーションに渡される．一方，提案手法を模擬したシステムの場合 (図 5) は，crypto ライブラリを使用して 3DES の暗号化を行ったデータをあらかじめターゲットのディスクに格納しておく．シーケンシャルリードアクセスを行う際に，ターゲットのディスクから事前に暗号化されたデータを読み出し，イニシエータに転送する．その後，イニシエータの上位層 (アプリケーション) で crypto ライブラリを使用して復号化

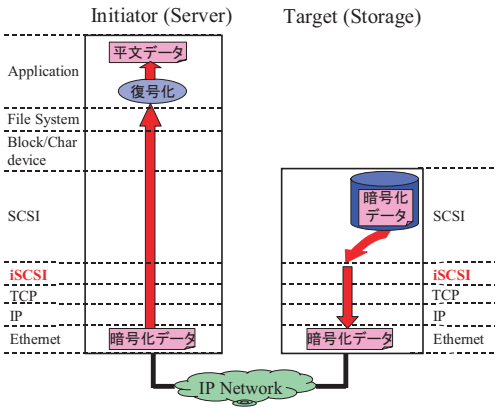

図 5: 提案手法を模擬した iSCSI アクセス実験

を行う．

本実験では，提案方式においてシーケンシャルリードアクセスの際に暗号化処理を行ってはいないため，IPsec を用いた場合との公平な比較ではない．しかし，提案手法の実装が完成し，暗号化処理時間を通信処理の待ち時間に隠蔽することが可能になった場合における理想的なケースをモデル化したものと考えることができる．

5. 実験結果の考察と解析

5.1 スループットの比較

単純なソケット通信の実験 (4.2 節) と iSCSI によるストレージアクセスの実験 (4.3 節) において，ブロックサイズを変化させてスループットを測定した結果を図 6，8 に示す．

iSCSI を用いない場合においては，上位層で暗号化する方式は，IPsec を用いた方式より性能が高く，全体のブロックサイズによる平均では約 12.3MB/sec であり，約 30%ほど向上することがわかった．これは，上位層でブロックをまとめて暗号化した方が IPsec でパケットごとに暗号化するより効率が良かったためと考えられる．

一方，iSCSI を用いたストレージアクセスを行った実験では，平均で約 10.4MB/sec に達し，IPsec による暗号化より，17%ほどの向上がみられた．図 8 は，先述のように理想的なケースをモデル化した場合との比較であるが，提案方式は IPsec を用いた方式に比べ，大幅な性能の向上がみられた．

また実際には，IPsec と OpenSSL の暗号化コードは全く同じでは無いが，3DES の暗号化は負荷が大きく，暗号化の計算が実行の大部分を占めるため，この点に関してはほぼ公平な比較であると考えら

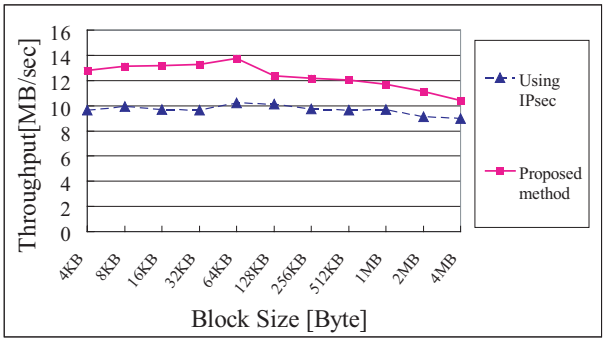

図 6: 単純なソケット通信によるシーケンシャルリードアクセスのスループット測定結果

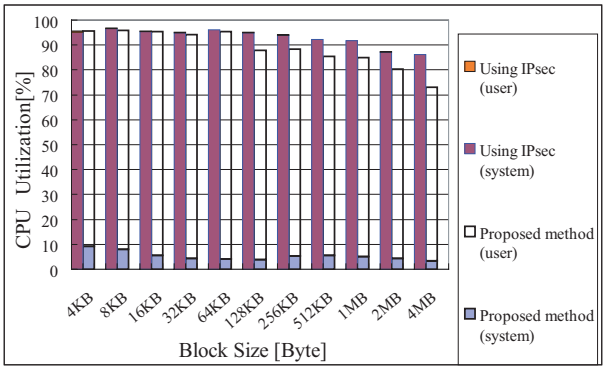

図 7: 単純なソケット通信によるシーケンシャルアクセスの CPU 使用率測定結果

れる

5.2 CPU 使用率の比較

2 つの実験 (4.2, 4.3 節) において，システムとユーザの内訳で全体の CPU 使用率をイニシエータ側で測定した結果を図 7，図 9 に示す．CPU 使用率の測定には，Linux の “iostat” コマンドを用いて 1 秒ごとに測定している．

iSCSI を用いない場合の実験では，比較的大きなブロックサイズでデータを転送すると，上位層で暗号化する提案方式の方が全体の CPU 負荷が小さくなることがわかった．測定したブロックサイズにおける平均では，全体の CPU 使用率は IPsec を利用した場合と比べて，約 3%低くなっており，ブロックサイズが 4MB の場合においては，CPU の負荷が 15%減少した．

iSCSI 通信において提案手法の模擬システムを利用した実験では，IPsec を利用する場合と比較して，

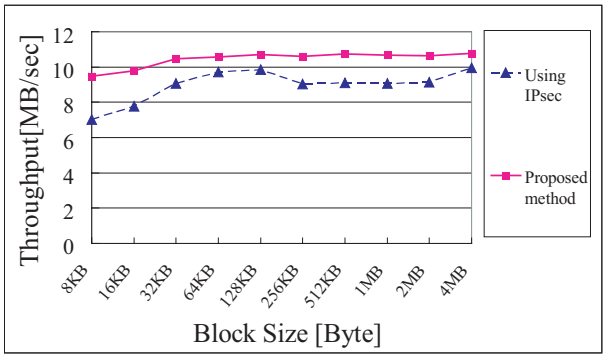

図 8: iSCSI を用いた試作システムによるシーケンシャルリードアクセスのスループット測定結果

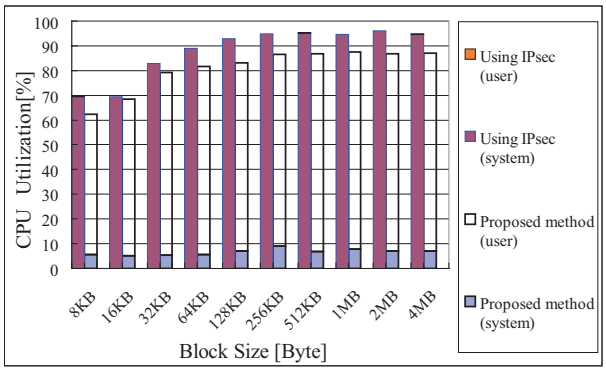

図 9: iSCSI を用いた試作システムによるシーケンシャルリードアクセスの CPU 使用率測定結果

平均で全体の CPU 使用率が 8%減少した．

また，2 つの実験において，IPsec を利用した通信では，“user” の CPU 使用率がほとんど 0%に近く，“system” の CPU 使用率が大部分を占めている．それに対して，提案手法による通信では，“system” の CPU 使用率は，10%にも満たず，CPU 使用率の大部分が “user” に占められている．

IPsec の暗号化はカーネルモードにおける処理であり，一方，提案手法である上位層の暗号化処理はユーザモードにおける処理である．一般に，カーネルモードの実行の方がプロセスの優先度が高いため処理が速いが，それにもかかわらずこの場合はユーザモードにおける実行の方が速く，CPU の負荷も軽減されている．また，優先度の高いカーネルモードで CPU 負荷の高い処理を大量に行うことは，ユーザモードで動作する他のサービスの著しい性能低下や停止につながるため好ましくなく，これをユーザモードの方へ移動させる上位層における暗号化方式

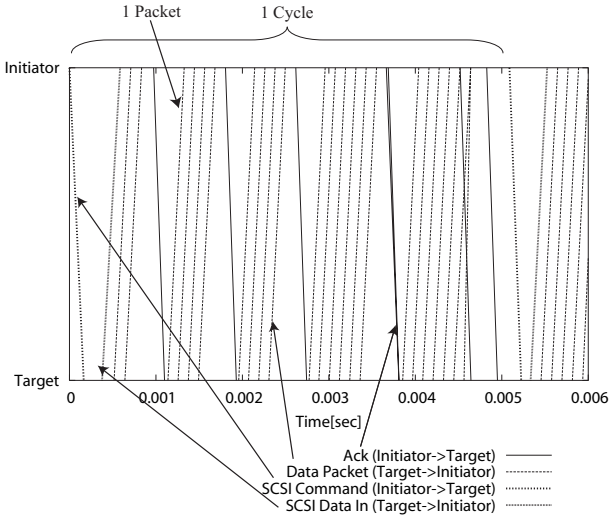

図 10: IPsec を用いた方式の TCP パケット転送

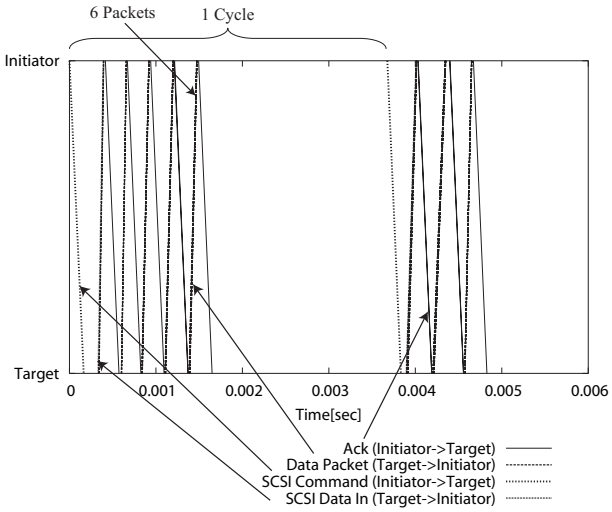

図 11: 提案方式による TCP パケット転送

は望ましい．付録は，通常の単純な演算処理を実行したカーネルモードとユーザモードの処理速度の比較した結果である．

現状では，上位層における暗号化方式において，通信処理の ACK 待ちの間に次のブロックを暗号化するような処理は実現していないが，これを実現して暗号化処理を通信時間に隠蔽するなどを行えば，図 8 で示された理想的なケースに近い性能の向上が期待できる．

5.3 TCP パケット転送の可視化

提案方式と IPsec による暗号化方式の振る舞いを確認するため，TCP 層における TCP パケット転送の様子を時間軸上に可視化した．ブロックサイズ 32KB において 0 秒から 0.006 秒までイニシエータとターゲット間に転送された TCP パケットの様子を図 10，11 に示す．

提案方式 (図 11) は IPsec を用いる方式 (図 10) に比べ，ブロックサイズ分のデータをイニシエータに転送しおわる 1Cycle がより短くなっている．また，IPsec を用いる方式においては，ターゲットは小さなサイズのデータを 1 パケットずつ暗号化/復号化処理を行いながら送受信しているため，各パケットの送信間隔が拡大している．提案方式においては，ブロックサイズ分のデータ 6 パケットをまとめてイニシエータに送信しているため，送信間隔は拡大しておらず，一方，1Cycle が終わった後に，まとめて復号化処理を行っていることがわかる．

6. まとめ

本稿では，iSCSI ネットワークで安全なストレージアクセスを実現するために，IPsec による暗号方式の代わりに，IPsec 層より上位層で暗号化する方式を提案した．基礎実験として，iSCSI を用いない環境において，単純なソケット通信によるシーケンシャルリードアクセスの比較実験を行った．また，提案手法の一部を実装した模擬システムを利用して，iSCSI を使ったシーケンシャルリードアクセスの実験を行った．どちらの実験においても，提案手法によって IPsec を利用した暗号化手法より，スループットと CPU 使用率の点において，性能が向上することがわかった．さらに，iSCSI を用いた場合の実験において，TCP パケット転送を可視化し，両方式の振る舞いの違いを確認した．

今後の課題として，提案手法によるシステムを完全に実装し，その性能を評価する．

謝辞

本研究は一部，文部科学省科学研究費特定領域研究課題番号 13224014 によるものである．

参考文献

[1] IETF IP Storage (ips) Charter, http://www.ietf.org/html.charters/ips-charter.html

[2] Storage Networking Industry Association, http://www.snia.org/

[3] 神坂 紀久子，山口 実靖，小口 正人: “IPsec を利用した iSCSI ネットワークにおけるシーケンシャルアクセスの考察”, 信学会全国大会，B-16-10, p.619, 2004 年 3 月.

[4] 山口 実靖, 小口 正人, 喜連川 優: 「iSCSI 解析システムの構築と高遅延環境におけるシーケンシャルアクセスの性能向上に関する考察」, 通信学会論文誌, Vol. J87-D-I, No. 2, pp. 216-231, 2004 年 2 月

[5] 神坂 紀久子, 山口 実靖, 小口 正人: “IPsec を利用した iSCSI ストレージアクセス時の TCP パケット転送の解析”, 情報処理学会研究報告, 2004-HPC-97, HOKKE2004, pp. 145-150. 2004 年 3 月.

[6] W. T. Ng, B. Hillyer, E. Shriver, E. Gabber, and B. Ozden, “Obtaining High Performance for Storage Outsourcing,” *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, pp. 145-158, Jan. 2002.

[7] P. Sarkar and K. Voruganti, “IP Storage: The Challenge Ahead,” *Proc. Tenth NASA Goddard Conference on Mass Storage Systems and Technologies*, pp. 35-42, Apr. 2002.

[8] P. Sarkar, S. Uttamchandani, and K. Voruganti, “Storage over IP: When Does Hardware Support help?,” *Proc. FAST 2003, USENIX Conference on File and Storage Technologies*, pp. 231-244, Jan. 2002.

[9] P. Radkov, L. Yin, P. Goyal, P. Sarkar and P. Shenoy, “Performance Comparison of NFS and iSCSI for IP-Networked Storage,” *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, pp. 101-114, Mar. 2004.

[10] S-Y. Tang, Y-P Lu and D. H. C. Du, “Performance Study of Software-Based iSCSI Security,” *Proc. First International IEEE Security in Storage Workshop*, pp. 70-79, Dec. 2002.

[11] OpenSSL Project, http://www.openssl.org/

[12] InterOperability Lab in the University of New Hampshire, http://www.iol.uhn.edu/consortiums/iscsi/

[13] iSCSI Draft, http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt

[14] FreeS/WAN Project, http://www.freeswan.org/

付録

カーネルモードとユーザモードにおける処理速度の比較

多くの OS ではカーネルモードにおける処理の方がユーザモードにおけるよりも処理よりも優先度が高く, 同一処理を行った場合カーネルモードにおいて実行した方が処理時間が短くなる. 本章で Linux 2.4 における両モードでの処理時間の比較例を示す. (1) 再帰処理によりフィボナッチ数列を求める処理, (2)2 整数の剰余演算を多数回繰り返す処理, の 2 種類の処理を両モードで行いその処理時間を計測し図 12, 13 の結果を得た. 図 12 の横軸は同数列の項数であり, 縦軸はその計算にかかった時間を表す. 同処理では関数呼び出しを多数回繰り返す. 同図よりカーネルモードにおける処理の方が処理時間が短い事が確認され, その比は 1.2 倍程度であった. 図 13 の横軸は整数の剰余演算の反復回数であり単位は百万回である. 同様に同図より, カーネルモードで処理を行った方が短い時間で処理を終えられることが確認でき, 同例においてはその比は 1.77 倍程度であることが確認できる.

また, 剰余演算を 15 億回の例において, カーネルモード, ユーザモードの処理時間はそれぞれ 35.08 秒, 61.95 秒であったが, これらを同時に処理させた場合カーネルモードにおける処理時間が 35.08 秒であり, ユーザモードが 132.26 秒であった. これによりカーネルモードにおける処理が優先度が高くユーザモードの処理に CPU が割り当てられていないことが確認できる.

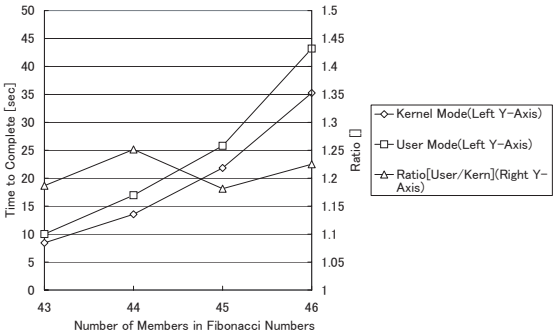

図 12: カーネルモードとユーザモードの比較 (A) : フィボナッチ数列

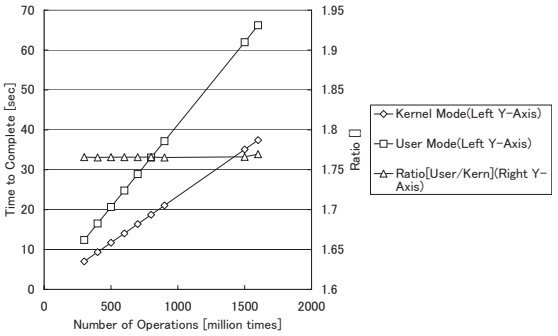

図 13: カーネルモードとユーザモードの比較 (B) : 2 整数の剰余演算