社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

iSCSI ストレージアクセス時の 3DES アルゴリズムを用いた上位層における暗号化適用方式の実装および IPsec との性能比較

神坂紀久子† 　山口　実靖†† 　小口　正人†

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2–1–1
†† 東京生産技術研究所　〒 153–8505 東京都目黒区駒場 4–6–1
E-mail: †kikuko@ogl.is.ocha.ac.jp, ††sane@tkl.iis.u-tokyo.ac.jp, †††oguchi@computer.org

あらまし　ネットワークストレージ技術である IP-SAN の代表的なプロトコルである iSCSI は，TCP/IP と Ethernet を使用することにより，ストレージの導入および管理コストを軽減することが可能である．iSCSI では，安全な通信を行うため IPsec を使用することができる．しかし IPsec は 3DES アルゴリズムの計算量が多いため，ストレージアクセスの通信性能を大幅に低下させる．また IPsec は下位の IP 層で処理するため，効率的な暗号化を行うことが難しい．そこで本稿では，iSCSI 層より上位層で暗号化処理を行う方式によって性能を向上させるシステムを実装した．また低遅延・高遅延環境において，上位層で暗号処理の最適化を行った場合を想定し，複数プロセス起動によるシステムの性能を評価した．さらに TCP パケット転送の様子を解析することにより，IPsec 使用時との比較について論じた．その結果，暗号処理の最適化を行う提案手法は高遅延環境において有効であることが示された．

キーワード　iSCSI，IPsec，ネットワークストレージ，暗号化，3DES

Implementation of Encryption Application System in the Upper Layer using 3DES algorithm and Performance Comparison with IPsec in iSCSI Storage Access

Kikuko KAMISAKA†, Saneyasu YAMAGUCHI††, and Masato OGUCHI†

† Ochanomizu University　Otsuka 2–1–1,Bunkyo-Ku, Tokyo 112–8610 Japan
†† Institute of Industrial Science, The University of Tokyo　Komaba 4–6–1, Meguro-ku, Tokyo, 153–8505 Japan
E-mail: †kikuko@ogl.is.ocha.ac.jp, ††sane@tkl.iis.u-tokyo.ac.jp, †††oguchi@computer.org

Abstract　iSCSI, common protocol in network storage IP-SAN, allows us to reduce management costs and costs of introducing storages by using TCP/IP protocol and Ethenet. To access remote storage securely, IPsec which encrypts transferred data can be employed in iSCSI. However, since 3DES algorithm needs a lot of amount of calculation, the performance of storage access degrade remarkably. In addition, it is difficult to execute encryption processing efficiently, because IPsec layer is located in a lower-level. In this paper, we implemented the system of executing encryption processing in the upper layer instead of IPsec to improve the performance. Furthermore, we simulated the idea of optimization of encryption in the upper layer and measured its performance by running multiple processes in a low-latency environment and a high-latency environment. As a result of analyzing the performance by visualizing TCP packets transfer, our proposed scheme of encryption processing optimization is effective in the high-latency environment.

Key words　iSCSI, IPsec, Network Storage, Encryption, 3DES

1. はじめに

インターネットなどの通信性能が急激に向上し，組織や企業などで保存，管理されるデータ量が年々指数関数的に増大している．これらの要因により，ストレージ群とサーバ群を高速なネットワークで接続する SAN(Storage Area Network) が多

く導入されるようになってきた．従来のストレージ形態であるサーバにストレージを直接接続する DAS(Direct Attached Storage) と比べ，SAN はストレージを統合することにより，バックアップ，キャパシティプランニングなどに伴う管理負荷を大幅に削減できる．

現在では SAN を構築するために，高速な専用回線を用いる Fibre Channel(FC) 技術が多く使用されている．しかし，FC 用のスイッチやインタフェースが高価であること，FC 管理技術者が少ないことなど SAN を新たに構築するには障害があった．そのため，FC よりも安価に導入や管理が可能である IP-SAN が提案された．IP-SAN は，FC に代わり TCP/IP プロトコルと Ethernet で構築されるため，導入コストや管理コストを抑えることができる．また既存のネットワークとのシームレスな統合ができ，FC よりネットワークを柔軟に設計できる．

IP-SAN で使用される技術で代表的なものに，2003 年 2 月に IETF により正式承認された iSCSI(Internet SCSI) プロトコルがある [1]．iSCSI では，サーバ (Initiator) とストレージ (Target) 間を Ethernet で接続し，TCP/IP パケットの中に SCSI コマンドをカプセル化することによってストレージアクセスを行うデータ通信プロトコルである．これにより，iSCSI はアクセスインタフェースを変更することなく，ローカルストレージと同様に，遠隔ストレージへのアクセスを可能にする．

TCP/IP ネットワークを介する iSCSI プロトコルを使用する場合，安全に通信を行うためにはセキュリティ対策を考慮する必要がある．そこで iSCSI では IP パケットに対し強固な暗号化と認証機能を提供する IPsec を利用することが可能となっており，暗号化・復号化に共通鍵暗号化アルゴリズムである 3DES(Triple Data Encryption Standard) を使用する場合が一般的である．しかし大量のデータがバースト的に発生する特徴をもつ SAN では，3DES の暗号化・復号化処理によって通信性能が大幅に低下する．また IPsec では下位層である IP 層で暗号化処理を行っており，上位層から渡されるデータセグメントに対して暗号化処理を繰り返すという逐次的な処理を行うため，TCP 層などの処理に柔軟に対応できず，効率的な暗号化を行っているとは言いがたい．

そこで本稿では，安全に iSCSI ストレージアクセスを行うために，iSCSI 層より上位層で暗号化を行うシステムを実装した．これにより，TCP 層などの処理内容を把握した上で，性能を向上するための様々な手法を上位層で適用することができる．また本稿では，低遅延・高遅延環境において，実装したシステムを用い，通信を行っている間に次のデータを暗号化する暗号処理の先処理による最適化を模擬し，複数プロセス起動による iSCSI シーケンシャルリードアクセスの性能評価実験を行った．さらに，TCP パケット転送の様子を解析し，実装したシステムと IPsec を使用した場合の iSCSI シーケンシャルリードアクセスの性能について論じた．

本稿の構成は以下の通りである．まず，2 章において上位層における暗号化適用方式とその実装について説明し，3 章で実装したシステムの複数プロセス起動による性能評価実験とその結果を示す．また 4 章で TCP パケット転送の解析による性能評価実験の考察について述べ，最後に 5 章でまとめる

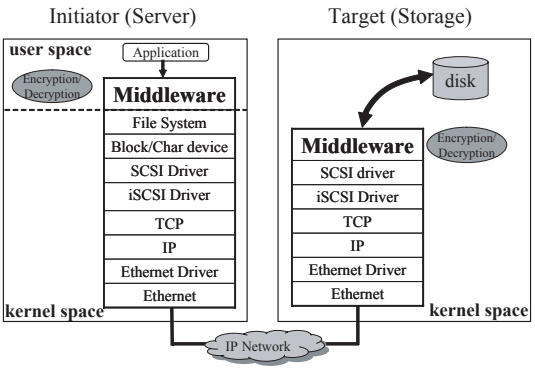

図 1　提案手法による iSCSI ストレージアクセスモデル

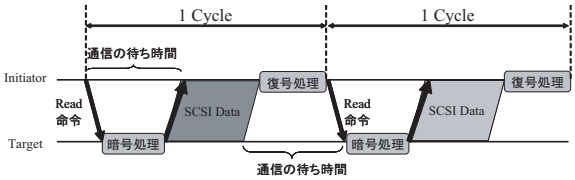

図 2　暗号化/復号化を使用した iSCSI シーケンシャルリードアクセス

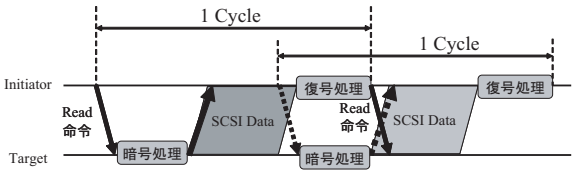

図 3　最適化した場合の iSCSI シーケンシャルリードアクセス

2.　上位層における暗号化適用方式の実装

iSCSI プロトコルを用いて安全にストレージアクセスを行う際の課題として，IPsec を用いた場合のストレージアクセス性能の低下が挙げられる．これは IPsec で一般的に使用される 3DES の暗号化処理が大きな障壁となっている．

IPsec は IP パケットを暗号化するため，上位のソフトウェアを変更する必要がなく透過的に暗号化を行うことができるが，一方で下位層に位置しているため，上位層から渡されたデータを逐次的に処理するのみであり，効率的に暗号化処理ができない．また性能を向上するための機能を容易に追加することができない．しかし iSCSI より上位層で暗号化を適用することにより，アプリケーションや SCSI 層，TCP 層などにおける処理に柔軟に対応することができ，上位層のソフトウェアで様々な性能向上手法を適用できる．本稿では上位層で暗号化処理を適用する方式に基づくミドルウェアを実装した．実装したミドルウェアによる iSCSI ストレージアクセスモデルを図 1 に示す．本システムでは，Initiator 側においてはアプリケーションより下位にあるミドルウェアとして暗号化・復号化機能を実装し，Target 側においては暗号化・復号化を行うカーネルモジュールを実装してミドルウェアを構築している．暗号化・復号化機能をミドルウェアとして独立させることにより，簡単に性能を向上させる手法を適用し，機能を追加，改良することができる．

また本稿のシステムでは，IPsec においてデフォルトで使用されている 3DES 暗号化アルゴリズムと同じ実装コードを使用している．

実装したミドルウェアを用いたシステムでは，我々が提案してきた暗号処理最適化による性能向上手法を容易に適用することが可能である [2]．図 2 より，iSCSI プロトコルを用いて暗号化・復号化を行うシーケンシャルリードアクセスでは，まず iSCSI Read コマンドが Initiator から Target に発行され，その応答として Target のディスクからデータが読み出された後，ミドルウェアで順次暗号化が行われる．その後暗号化されたデータが下位層に渡され，Initiator に送信される．送信されたデータは Initiator の上位にあるミドルウェアで復号化され，確認応答 (Ack) を返す．しかし Initiator で復号化を行っている間，Target で暗号化を行っている間などに通信の待ち時間が発生する．

図 3 に示すように，暗号処理最適化による性能向上手法では，この通信の待ち時間の間に次のデータの暗号化・復号化処理を行うことによって，iSCSI の Cycle をオーバーラップさせ，CPU 処理の空き時間を有効に使用し，性能を向上させる．

このように提案手法においては，上位層で暗号化/復号化するミドルウェアによって暗号化の先処理による最適化を行うなど，上位層で様々な工夫を行うことができ，性能向上に貢献することが可能となる．

3. 高遅延環境を用いた上位層における暗号化適用方式の性能評価実験

暗号処理最適化手法を評価するため，構築したシステムを用いて，Target の raw デバイスに対する iSCSI シーケンシャルリードアクセス時の性能を評価し，IPsec を用いた場合の性能と比較した．本実験では，アプリケーションにおいて複数プロセスを起動することにより，CPU 処理の空き時間の間に連続的にデータの暗号化を行って，暗号化処理最適化手法を模擬した性能を評価している．また，IP-SAN が非常災害対策などのために比較的遠距離で使用されることを想定し，高遅延環境において提案システムの性能を評価した．

3.1 実験環境

実験に用いたシステム環境を表 1，2，3 に示す．低遅延環境における実験では，Initiator と Target を Gigabit Ethernet スイッチで 1 対 1 接続した単純な構成になっている．高遅延環境における実験では，図 4 に示すように，Initiator と Target の間に人工的な遅延装置として FreeBSD Dummynet を設置した．片道遅延時間は 0ms，1ms，2ms，4ms，8ms と設定し測定した．

Initiator と Target の OS には Linux を用いた．また iSCSI の実装には，ニューハンプシャー大学 InterOperability Lab [3] が提供しているオープンソースの実装 (UNH-iSCSI) を用い，IPsec の実装には，Linux において広く利用されているオープンソースの FreeS/WAN [4] を用いた．IPsec の設定に，ホスト間の通信を暗号化するトランスポートモードおよび ESP プロトコルを使用している．セキュリティの観点から IPsec との比較について考慮すると，IPsec はゲートウェイ間通信を暗号化するトンネルモードを用いた場合には TCP 層のデータと TCP ヘッダを暗号化する．本稿の IPsec との比較による性能評価実験において，提案手法では TCP ヘッダは暗号化されないが，IPsec のトランスポートモードと比較を行っているため，どちらの方式も IP ヘッダは暗号化されない．よって，セキュリティの面においてはどちらの方式も同等のレベルであると考えられる．

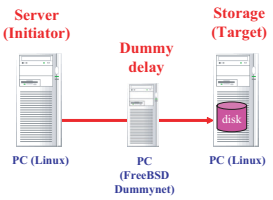

図 4　高遅延環境における実験環境

表 1　性能評価実験環境 1：使用計算機

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　性能評価実験環境 2：使用計算機

Dummynet OS	Free BSD 4.9 - RELEASE
CPU	Intel Xeon 2.4GHz
Main Memory	512MB DDR SDRAM
NIC	Intel PRO/1000MT Server Adapter on PCI-X (64bit, 100MHz)

表 3　性能評価実験環境：使用実装

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

3.2 スループット測定結果と考察

低遅延環境 (片道遅延時間 0ms) において，本稿のシステムを用いて複数プロセスを起動し測定した際のスループットと IPsec を使用した際のスループットを図 5 に示す．また，高遅延環境 (片道遅延時間 8ms) におけるスループットを図 6 に示す．表 4 は，IPsec を 1 とした場合の各プロセスにおけるスループット向上比率を全ブロックサイズで平均したものである．

同表より，片道遅延時間 0ms である低遅延環境において 1 プロセスを起動した場合，提案システムは IPsec よりスループットが低く，0.75 倍にとどまっている．しかし，2，3，4 プロセスを起動した場合には IPsec よりも 1.3 倍から 1.5 倍程度性能が向上することがわかった．高遅延環境においては，ブロックサイズを増加させるとスループットも向上するという結果が得

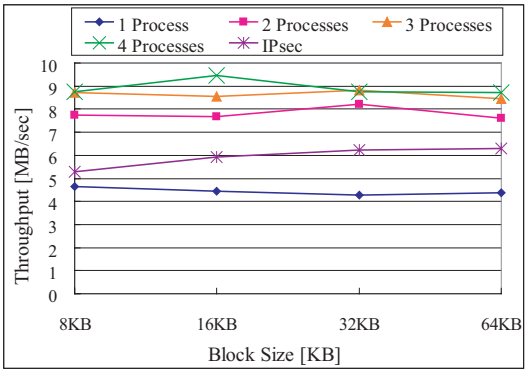

図 5　片道遅延時間 0ms におけるスループット

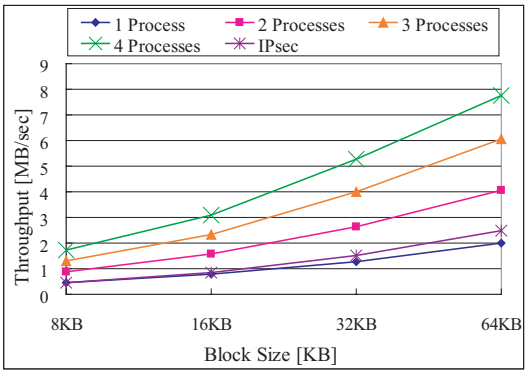

図 6　片道遅延時間 8ms におけるスループット

表 4　IPsec に対するスループットの向上比率 (片道遅延時間 0ms〜8ms)

	IPsec	1pro	2pro	3pro	4pro
0ms	1.000	0.752	1.323	1.465	1.512
1ms	1.000	0.719	1.379	1.825	2.082
2ms	1.000	0.772	1.510	2.093	2.522
4ms	1.000	0.804	1.607	2.382	3.013
8ms	1.000	0.894	1.803	2.692	3.537

られた．これは 1 つのデータセグメントを送信するのにかかる時間が長くなるためであると考えられる．また片道遅延時間を増加させると，提案システムの性能向上比率も増加し，最終的に片道遅延時間が 8ms の場合には，2 プロセスで 1.8 倍，3 プロセスで 2.6 倍，4 プロセスで 3.5 倍の性能向上を確認できた．

低遅延環境においては，通信時間はデータセグメントの暗号化・復号化時間と比較して相対的に短い．そのため，提案手法における暗号化の先処理による最適化はプロセス数を増加させてもそれほどの効果は見られない．しかし，高遅延環境においては，通信時間はデータセグメントの暗号化・復号化時間と比較して相対的に長くなる．そのため，通信の待ち時間，つまり CPU 処理の空き時間が長くなることにより，1 つの暗号化サイクルが終了しないうちに，連続的に次のデータセグメントを暗号化する暗号処理最適化手法の効果が高くなるため，性能向上比率が大きくなったと考えられる．

3.3　CPU 使用率測定結果と考察

図 7，8 は，低遅延環境 (片道遅延時間 0ms) と高遅延環境 (片道遅延時間 8ms) において，本稿のシステムを用いて複数プロセスを起動し測定した際の CPU 使用率と IPsec を使用した際の CPU 使用率である．本実験では Linux の iostat コマンドを用い，Target 側で CPU 使用率を測定した．表 5 は測定した全ブロックサイズの CPU 使用率の平均である．

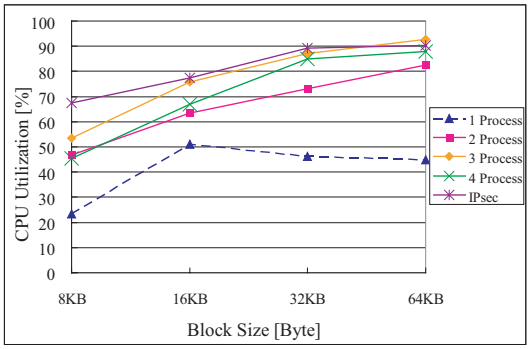

図 7　片道遅延時間 0ms における CPU 使用率

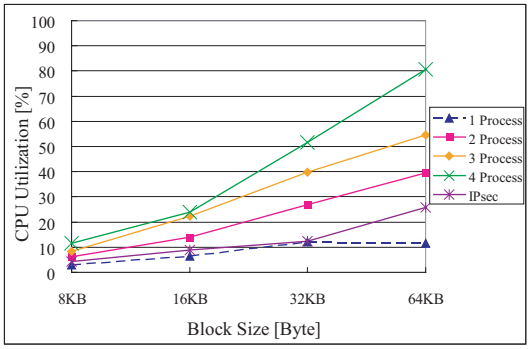

図 8　片道遅延時間 8ms における CPU 使用率

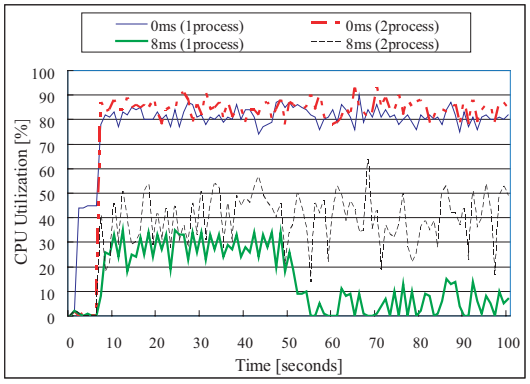

図 9　提案システムの CPU 使用率の時間遷移図

表 5　各遅延時間における CPU 使用率の平均 (%)

	IPsec	1pro	2pro	3pro	4pro
0ms	81.148	41.408	66.428	77.306	71.375
1ms	43.777	29.066	54.386	73.014	71.792
2ms	33.593	22.429	43.092	59.738	67.348
4ms	18.768	14.305	30.803	42.707	56.577
8ms	12.899	8.351	21.642	31.228	41.915

低遅延環境において，IPsec を用いた際の CPU 使用率は提案システムの複数プロセスを起動した場合よりも高い値となった．表 5 より，提案システムの場合が最大で 77%であるのに対し，IPsec を使用した場合は 81%に達している．高遅延環境に

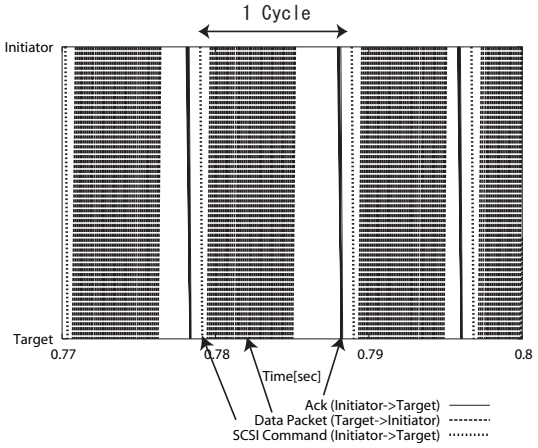

図 10　片道遅延時間 0ms における IPsec を使用した場合の TCP パケット可視化図

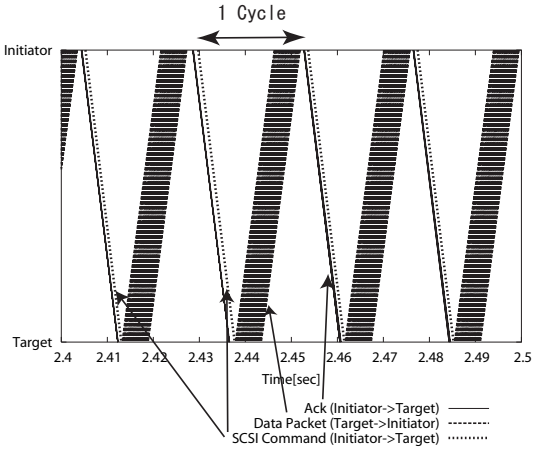

図 12　片道遅延 8ms における IPsec を使用した場合の TCP パケット可視化図

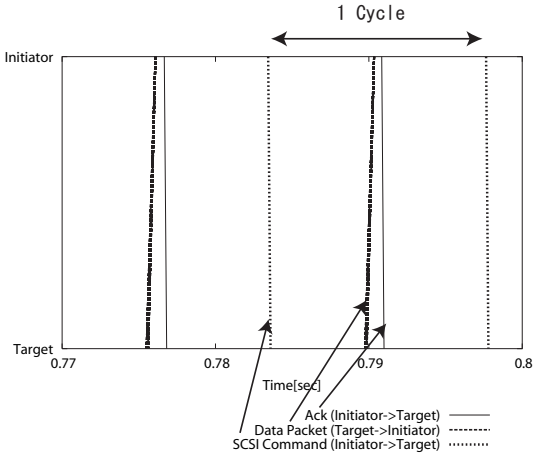

図 11　片道遅延時間 0ms における提案手法で 1 プロセスを起動させた場合の TCP パケット可視化図

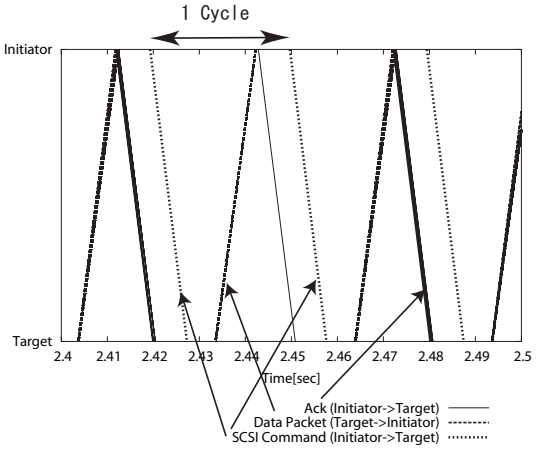

図 13　片道遅延 8ms における提案手法で 1 プロセスを起動させた場合の TCP パケット可視化図

おいては，IPsec を使用した際の CPU 使用率は提案システムの 2 プロセスを起動した際の CPU 使用率より低い値を示した．また図 8 より，ブロックサイズが増加するとともに CPU 使用率も増加し，ブロックサイズ 64KB の場合には 4 プロセスにおいて CPU 使用率が約 80%に達していることがわかる．一方，片道遅延時間を増加させると提案システムの CPU 使用率は減少する．

図 9 はブロックサイズ 64KB の場合の提案システムで 1 プロセス，2 プロセスを起動した場合の CPU 使用率の時間遷移である．同図は片道遅延時間 0ms と 8ms で Target 側で測定した．これより，低遅延環境の場合は全体的に CPU 使用率が高く，高遅延環境の場合には全体的に CPU 使用率が高いことがわかる．また 1 プロセスよりも 2 プロセスの方が CPU 使用率が高く，CPU の空き時間を効率的に活用して暗号化を行っているといえる．

低遅延環境では提案システムの CPU 使用率は IPsec よりも低くなるが，全体的に高い CPU 使用率となった．一方高遅延環境においては，CPU 使用率は IPsec よりも高くなるが CPU においてはまだ余裕がある．これは通信時間が長いために CPU 処理の空き時間が長くなり，その間に次のデータの暗号化を進める余裕が大きくなるためと考えられる．

4.　TCP パケット転送の可視化による解析

本節では提案手法と IPsec の暗号化の振る舞いを詳細に確認するため，ブロックサイズ 64KB の場合の TCP パケットを tcpdump ツールによってキャプチャし，Initiator と Target 間で転送された TCP パケットを時間軸上に可視化した [5] [6].

図 10 は低遅延環境において，IPsec を用いてシーケンシャルリードアクセスを行った際の TCP パケット転送の様子である．IPsec を使用した場合は下位層である IP 層に渡されるまでに小さなセグメントに分割されてから暗号化される．そのため，一つのパケットに対する暗号化時間が短くなっている．図 11 は低遅延環境において，提案システムを用いて 1 プロセスを起動した際の TCP パケット転送の様子である．1 プロセスを起動し提案手法である上位層で暗号化を行った場合は，上位層で大きなブロックをまとめて暗号化・復号化しているため，TCP 層では一つのブロックに対する暗号化・復号化時間が長くなっている．

次に，片道遅延時間 8ms の高遅延環境の場合も同様に TCP パケット転送の様子を可視化した．図 12 は IPsec を用いた場

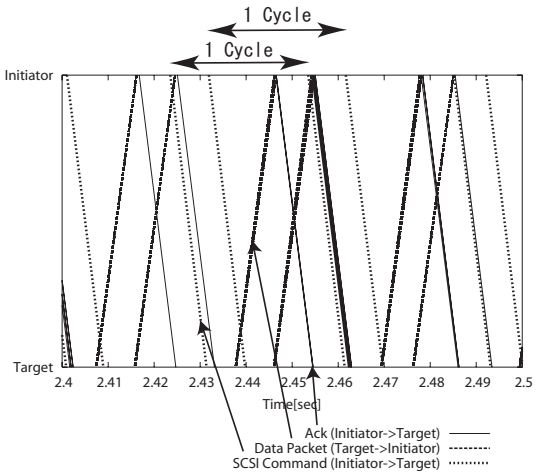

図 14　片道遅延 8ms における提案手法で 2 プロセスを起動させた場合の TCP パケット可視化図

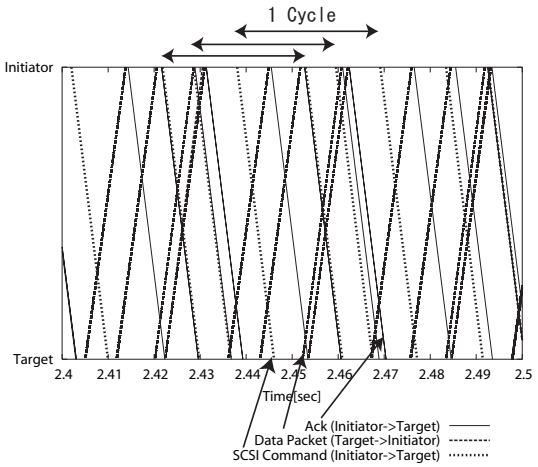

図 15　片道遅延 8ms における提案手法で 3 プロセスを起動させた場合の TCP パケット可視化図

合であり，図 13，14，15 は提案システムを用いて 1 プロセス，2 プロセス，3 プロセスを起動した場合である．これらの図より，1 プロセス起動よりも 2 プロセス起動の方が同じ時間内に SCSI Command を発行する回数が多くなっており，CPU 処理の空き時間を利用して暗号化・復号化処理を効率的に行っているといえる．さらに 3 プロセス起動の場合には，2 プロセスよりも SCSI Command を多く発行しており，IPsec を使用した場合と比較してデータセグメント暗号化の並列性が高くなり，提案システムが有効であることを示している．

また低遅延環境の場合と比較すると，どちらの図も iSCSI の 1Cycle が大幅に長くなっている．高遅延環境では通信時間が暗号化・復号化時間よりも相対的に長い．そのため，CPU の空き時間が多くなり，暗号化の先処理による最適化手法が低遅延環境より有効であるといえる．

5. 関連研究

iSCSI の関連研究としては，まず文献 [7] においては，Sarkar らによる iSCSI のソフトウェア実装と TOE(TCP Offload Engine) や HBA(Host Bus Adapter) を用いた iSCSI ハードウェア実装の比較に関する研究が行われており，ハードウェア実装は，CPU の負荷を軽減させることはできるが，総合的にはソフトウェア実装の方が性能が高くなることが実証されている．文献 [8] において，Aiken らは iSCSI ソフトウェア実装における遅延を考慮した性能の比較と詳細な分析を行い，ネットワーク遅延が増加するにつれ，iSCSI プロトコルの通信性能が急激に低下するという結果を得た．現在まで iSCSI の性能に関する研究は多く発表されているが，暗号化などのセキュリティ実現方式が考慮された研究は十分になされていない．

6. まとめと今後の課題

本稿では，iSCSI ストレージアクセスにおける安全性と性能を考慮するために，3DES アルゴリズムを用いて IPsec の代わりに上位層で暗号化を行うシステムを実装した．また低遅延・高遅延環境において，暗号化処理の最適化による性能向上手法を模擬した複数プロセス起動による iSCSI シーケンシャルリードアクセス性能を評価し，IPsec を用いた場合の性能と比較した．さらに，TCP パケット転送の様子を時間軸上に可視化することにより，提案手法と IPsec を用いた場合の性能を解析的に比較し，考察を述べた．その結果，上位層で暗号化の先処理による最適化を行う手法は高遅延環境において有効であることがわかった．

今後の課題としては，Initiator と Target における暗号処理最適化手法をミドルウェアに組み込み，実装を完成させる．

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．

文　　献

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

[2] 神坂紀久子, 山口実靖, 小口正人: iSCSI ストレージアクセスにおける暗号化処理の最適化を考慮したシステムの提案と性能評価, 先進的計算基盤システムシンポジウム (SACSIS 2005), pp. 435–442 (2005).

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

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

[5] 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, 電子情報通信学会和文論文誌 データ工学特集号, Vol. J87-D-I, No. 2, pp. 216–231 (2004).

[6] 神坂紀久子, 山口実靖, 小口正人: 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, 情報処理学会研究報告, 2004-OS-97, SWoPP2004, pp. 97–104 (2004).

[7] Sarkar, P., Uttamchandani, S. and Voruganti, K.: Storage over IP: When Does Hardware Support help?, *Proc. FAST 2003, USENIX Conference on File and Storage Technologies*, pp. 231–244 (2002).

[8] Aiken, S., Grunwald, D., Pleszkun, A. R. and Willeke, J.: A Perfomance Analysis of the iSCSI Protocol, *Proc. 20th IEEE Symposium on Mass Storage Systems and Technologies (MSS '03)*, pp. 123–135 (2003).