# 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

神坂 紀久子[†] 山口 実靖[††] 小口 正人[†]

ネットワークストレージ技術であるIP-SANは，IPネットワーク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．そのIP-SANのプロトコルとして，iSCSIが注目を集めている．iSCSIストレージアクセスにおいて安全に通信を行うことは重要であるが，セキュリティ対策についてはまだ完全に確立されていない．転送データの暗号化にはIPsecを使用することも可能であるが，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はできない．そこで本稿では，最適化された暗号化処理を目的として，IP層より上位層で暗号化処理を行い，また暗号化の最適化によって性能を向上する手法を提案する．さらに，上位層で暗号化処理を行うシステムを実装し，複数プロセス起動による最適化処理を想定したシステムの性能評価実験を行った．

# Proposal and Performance Evaluation of iSCSI Storage Access Optimized for Encryption Processing

Kikuko Kamisaka,[†] Saneyasu Yamaguchi[††] and 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 the key issues in iSCSI is a security measure to access remote storage via IP networks, which is not necessarily unestablished. IPsec can be employed in iSCSI. However, it is difficult to performe encryption processing efficiently, because IPsec layer is located in a lower-level.

In this paper, to realize secure storage access optimized for encryption processing on iSCSI networks, we propose the idea of encryption scheme in the upper layer and optimization of encryption instead of IPsec encyrption scheme. We implemented the system of our proposed method and measured its performance by running a number of processes.

## 1. はじめに

ブロードバンド技術の普及によって，マルチメディアコンテンツなど大容量データが通信，管理されることが多くなってきた．企業や組織が取り扱うデータ容量も年々指数関数的に増大し，管理コストの急増が課題となっている．これらの要因により，ストレージ技術は，従来使用されてきたサーバにストレージを直接接続するストレージ体系であるDAS(Direct Attached Storage)から，ネットワークストレージに移行しつつある．現在では，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，ストレージ統合によって遠隔ストレージにある大容量データを容易に管理可能なSAN(Storage Area Network)が多く使用されている．

† お茶の水女子大学
Ochanomizu University
†† 東京大学生産技術研究所
Institute of Industrial Science, The University of Tokyo

現在使用されているSANとしては，ファイバチャネル技術を使用するFC-SANが主流を占めている．しかし，FC-SANはファイバチャネルスイッチやホストバスアダプタなどが高価であることや，接続距離に制限があるなどの問題が存在するため，FC-SANを導入するには障害があった．

それらの問題点を解消するためにIPネットワーク技術を使用するIP-SANが提案され，徐々に普及しはじめている．IP-SANでは，スイッチなどのハードウェアコストや管理コストが安価であるだけでなく，既存のネットワークとのシームレスな統合が可能であること，接続距離に限界がないことなどの利点がある．また近年のEthernetの劇的な進歩から，性能に関する将来性も期待できる．

そのようなIP-SANで使用される技術には，FCIP，iFCP，iSCSIなどのプロトコルが存在するが，中でも，2002年2月にIETFにより正式承認されたiSCSI(Internet SCSI)プロトコルが最も期待されて

いる[1]．

iSCSI[2] は，サーバ（Initiator）とストレージ（Target）間で用いられ，TCP/IP ネットワークと Ethernet を使用して IP-SAN を構築するデータ通信プロトコルである．iSCSI は，標準的に使用されている SCSI コマンドを TCP/IP パケットの中にカプセル化することによって，専用回線を用いることなく，IP ネットワーク上で遠隔ストレージへのアクセスを可能にする．また iSCSI は，SCSI over iSCSI over TCP/IP over Ethernet という複雑な階層構造をしている．

iSCSI を使用して遠隔ストレージにアクセスする場合，オープンな TCP/IP ネットワークを介するため，セキュリティ対策について十分考慮する必要がある．そこで iSCSI では IP ネットワークで信頼性のある IPsec とよばれる技術が利用可能になっている．

しかし，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だけであり，効率的な暗号化を行っているとは言いがたい．セキュリティとパフォーマンスはトレードオフの関係にあるため，暗号化などの安全な通信を行いながら可能な限り効率的に処理を行う必要がある．

そこで本稿では，iSCSI ストレージアクセスを行う際に，上位層で暗号化処理を実行し，かつ暗号化を最適化する手法を提案して，これに基づくシステムの実装を行った．提案手法を実現した場合，IPsec を使用するケースの逐次的処理とは異なり，通信を行っている間に，次のデータを暗号化するなどの先処理機能の実装や，複数のデータを平行して暗号化するなど，上位層でさまざまな工夫をすることができ，柔軟な処理が可能になる．

提案手法の有効性を評価するために，本稿ではまず，上位層で暗号化し，最適化を行った際の理想的なケースをモデル化した予備評価実験を行った結果，IPsec を使用する場合よりスループット，CPU 使用率の点で性能が向上することがわかった．次に，暗号化の最適化を行う模擬システムを実装し，性能を評価した．上位層でデータセグメントの暗号化の先処理を行った場合の効果を評価するため，複数プロセスを使用した iSCSI ストレージアクセスを行った．その結果，最適化を行う提案手法が有効であることがわかった．

本論文の構成は以下の通りである．まず，2 節において iSCSI について簡潔に説明し，3 節において本稿において提案する iSCSI システムについて説明する．そして，4 節において提案システムの予備評価実験，5 節で iSCSI Target 側の実装方法について詳細に説明し，6 節で複数プロセスを使用した場合の性能評価実験，7 節でまとめと今後の課題とする．

## 2. 研究背景

### 2.1 関連研究

iSCSI の関連研究としては，Wee Tech Ng らによる独自の SCSI over IP 実装の性能に関する研究[3] がある．また，P. Sarkar らによる iSCSI のソフトウェア実装と TOE や HBA による iSCSI ハードウェア実装の比較に関する研究などがあり，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，総合的な性能で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[4][5]．本研究では CPU 負荷軽減するだけでなく，スループットなどの総合的な評価を対象としているため，iSCSI のソフトウェア実装を用いている．また S. 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を比較している[6]．P. Radkov らは，iSCSI と NFS の性能比較を研究している[7]．K. Z. Meth らは，iSCSI プロトコルデザインについて詳細に研究している[8]．

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[9]．現在まで iSCSI の性能に関する研究は多くされているが，性能と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．

### 2.2 iSCSI ストレージアクセスにおける IPsec 適用の問題点

TCP/IP 技術に基づく iSCSI を使用してストレージアクセスを行う際には，IP ネットワークで広く使用され，IP パケットに対して強固な暗号化と認証機能を提供する IPsec を標準で用いることとされている．IPsec では，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，3DES は処理時間が長く，大量のデータを送受信するストレージアクセスで使用する際には，通信性能が大幅に低下する．

iSCSI シーケンシャルリードアクセスにおける IPsec を使用した場合と使用しない場合の性能を比較したところ，スループットが大幅に落ち，CPU 使用率も飽和状態になることが確認されている[10]．実験結果を解析したところ，3DES の暗号化処理によって，リードアクセス性能が著しく落ちることは当然であるが，IPsec は IP 層で暗号化を行っているため，暗号化処理が効率的に行われていないことがわかった[11][12]．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することができ

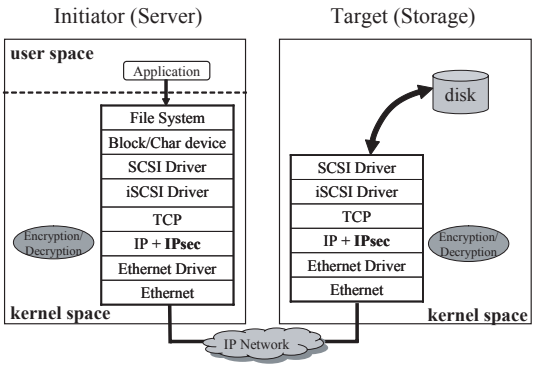


**図 1** IPsec を適用した iSCSI ストレージアクセスモデル

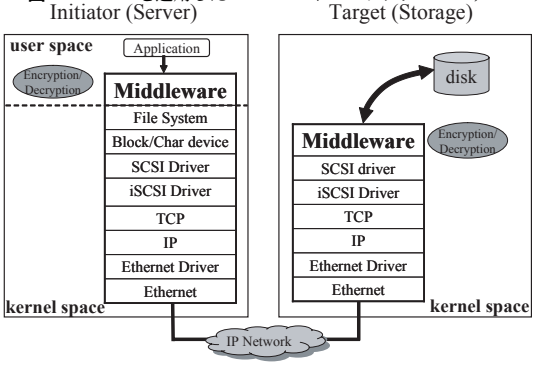


**図 2** 提案手法による iSCSI ストレージアクセスモデル

る．しかし，一方で IPsec は下位層に位置しているため，大規模なデータをシーケンシャルにアクセスする場合であっても，上位層から渡された小さなデータセグメントに対して暗号化処理や IPsec ヘッダの付加処理を逐次的にしか行うことができない．つまり，アプリケーション層，SCSI 層，TCP 層などの上位層の処理内容を把握した上で，最適化を行うなどの高機能な処理が実現不可能である．

Target のディスクから読まれたデータは SCSI 層に渡され，SCSI 層から TCP 層にデータが渡ったとき MSS（Maximum Segment Size）ごとにフラグメント化される．IPsec は，その細分化されたデータセグメントに対し，逐次的に暗号化処理やヘッダ処理を行う．よって IPsec を使用すると，3DES の計算量が多いだけでなく，暗号化処理やヘッダ処理などが細分化されたデータセグメントごとに行われるため，そのオーバーヘッドによって性能が低下する．

## 3. 提 案 手 法

そこで本稿では，IPsec より上位層で暗号化，復号化処理を行うためのミドルウェアを用いるシステムを提案する．IPsec を使用した場合の iSCSI Initiator と Target の階層プロトコルモデルを図 1 に，IPsec を使

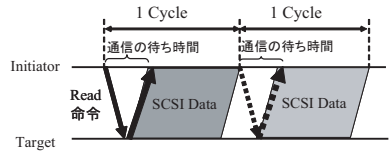


**図 3** 通常の iSCSI シーケンシャルリードアクセス

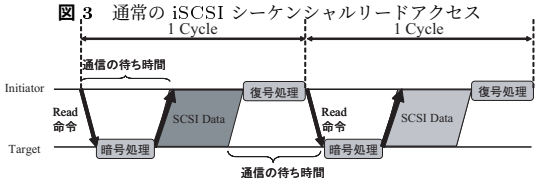


**図 4** 暗号化/復号化を使用した iSCSI シーケンシャルリードアクセス

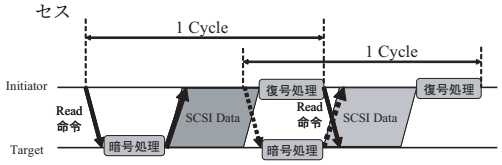


**図 5** 最適化した場合の iSCSI シーケンシャルリードアクセス

用せずに，上位層で暗号化処理を行った提案手法の場合を図 2 に示す．

iSCSI シーケンシャルリードアクセスを行う際，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て，下位層に渡され，その後 Initiator に送信される．最終的に Initiator のアプリケーションに全てのデータが渡されないと，次の Read コマンドは発行されない（図 3）．ここでは，Initiator から SCSI Read コマンドが発行され，Target のディスクからデータが読み出されて，そのデータが Initiator に送信されるまでを 1Cycle としている．

IPsec を使用した場合には（図 4），まず Target のディスクに格納されているデータが読み出される．次にそれらのデータが SCSI Driver，iSCSI Driver，TCP の各層を通って小さなサイズに分割され，IP+IPsec 層に渡される．そして IP+IPsec 層でそれらの小さなデータセグメントに対し，順次暗号化処理を行い，Initiator に転送する．Initiator 側では，下位層である IP 層において復号化を行い，上位層に平文データを渡す．このように IPsec を使用した場合には，暗号化は小さなデータセグメントごとに下位層で逐次処理するだけであるため，暗号化したデータを Initiator に送信している間に，次のデータの暗号化をするなどの柔軟な対応は困難である．

しかし，上位層で暗号化/復号化処理をするミドルウェアを用いたシステムでは，まとまったサイズのデータを暗号化することができ，かつ上位層で柔軟に処理を行うことができる．それによって，通信の待ち時間

に別のデータの暗号化/復号化処理を行い，iSCSI の Cycle をオーバーラップさせるという暗号化の先処理などの最適化を実現することが可能になる（図 5），

以上より，提案手法によって上位層で暗号化/復号化を行い，暗号化の先処理による最適化を行うと，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，上位層でさまざまな工夫をすることが可能になるため，アクセスの性能向上につながると考えられる．

## 4. 提案手法の簡易実装による予備評価実験

iSCSI シーケンシャルリードアクセスにおいて，上位層で暗号化処理を行う提案システムの予備評価実験を行った[13)14)]．iSCSI シーケンシャルリードアクセスでは，まず，Initiator が SCSI Read コマンドを発行し，これを受けた Target がディスクに格納されている平文データの暗号化を行い，その暗号データを IP ネットワーク経由で Initiator に送信する．そして，これを受信した Initiator 側で復号化を行う．

まず基礎実験として，iSCSI を使用しない通常の 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，上位層で行う性能に関する有効性を評価した．提案手法の上位層における暗号化には，OpenSSL の crypto ライブラリを用いており，アプリケーションから crypto ライブラリの暗号化・復号化ルーチンを呼び出して実験を行った．

次に上位層で暗号化を行うシステムを実装し，暗号化の先処理による最適化をモデル化した場合の iSCSI 通信の予備評価実験を行った．この実験では Initiator から Target 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っている．Initiator においては，基礎実験の場合と同様に OpenSSL の crypto ライブラリを使用して復号化処理を行うが，Target においては，データを先読みして暗号化した際の理想的なケースをモデル化しているため，シーケンシャルリードアクセスを行う際に暗号化処理を行っていない．つまり，暗号化され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き，シーケンシャルリードアクセスを行う際にはそのデータを Target 側のディスクから読み出している．

各実験の性能は，IPsec を使用した場合のスループットおよび CPU 使用率との比較を行った．

### 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，本稿では暗号化・復号化処理の性能をに焦点を当てて評価するため，Initiator と Target を Gigabit Ethernet スイッチで接続した単純な構成になっている．

表 1 実験環境：使用実装

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

表 2 実験環境：使用実装

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

iSCSI の実装には，ニューハンプシャー大学 InterOperability Lab[15)2)] 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[16)] を用いた．IPsec の設定では，ホスト間の通信を暗号化するトランスポートモード，ESP プロトコルを使用している．どちらの手法においても，暗号化アルゴリズムとして 3DES を用いており，OpenSSL における 3DES 暗号化の実装コードは IPsec の 3DES とほぼ同じものである．

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[12)]．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．

### 4.2 予備評価実験結果と考察

基礎実験である単純なソケット通信による実験結果を図 6，7 に示す．CPU 使用率は，kernel 空間と user 空間における処理の内訳で合計の CPU 使用率を Initiator 側で測定した結果である．提案手法である上位層における暗号化の際には，IPsec を使用した場合に比べ，平均してスループットが 30%向上し，CPU 負荷も約 3%軽減されるという結果が得られた．基礎実験では iSCSI を使用していないため，図 3 における SCSI Read コマンド発行による通信の待ち時間が発生しない．従ってこれは，待ち時間の間に暗号化処理を行う最適化をモデル化したケースを iSCSI を使用せずに評価した実験に相当し，本稿における提案を実装

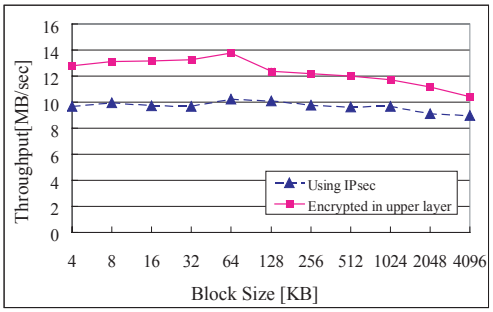


**図 6** 単純なソケット通信によるシーケンシャルリードアクセスのスループット

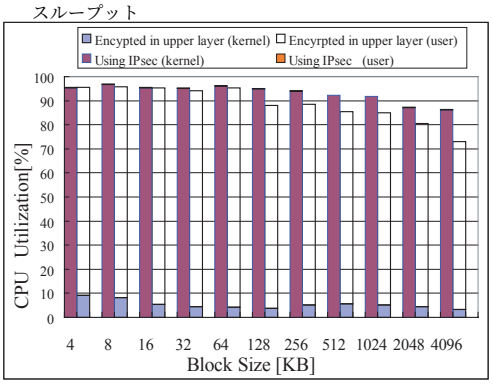


**図 7** 単純なソケット通信によるシーケンシャルリードアクセスの CPU 使用率（Initiator）

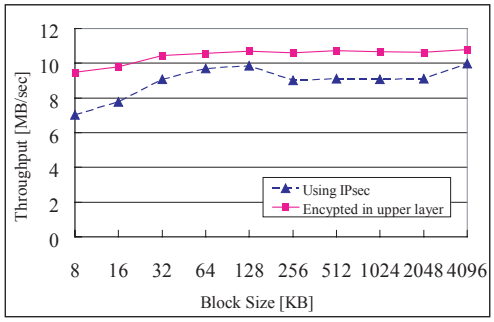


**図 8** iSCSI を用いたシーケンシャルリードアクセスのスループット

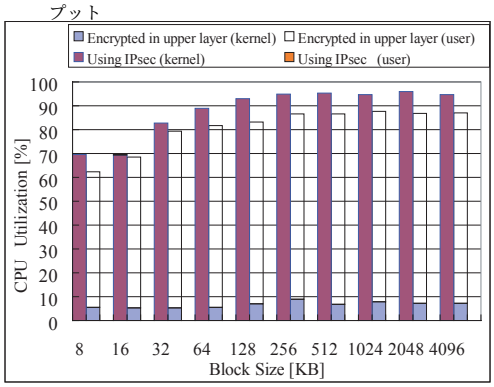


**図 9** iSCSI を用いたシーケンシャルリードアクセスの CPU 使用率（Initiator）

した場合には，この程度の性能向上が期待できると考えられる．

次に，iSCSI ネットワークを利用した簡易実装における各方式の予備評価実験結果を図 8，9 に示す．本実験は Target 側における暗号化処理を省略し，暗号化処理の最適化を行った場合，すなわち，Target のディスクにあるデータの暗号化先処理をした際の理想的なケースをモデル化した場合における比較である．その結果，提案方式は IPsec を用いた方式に比べ，約 17%のスループット向上がみられた．また，合計の CPU 使用率は，上位層で暗号化する手法が平均で 8%減少した．基礎実験より性能向上が劣っている要因は，iSCSI を介したストレージアクセスを行っているためであり，iSCSI は階層構造が複雑であるため，基礎実験による単純なソケット通信よりも性能が劣化すると考えられる．これらの結果より，上位層で暗号化の先処理による最適化をする提案手法が性能に関して有効であるといえる．

セキュリティの観点から考察すると，IPsec はホスト間通信を暗号化するトランスポートモードにおいて，TCP 層におけるデータと TCP ヘッダを暗号化する．本節の IPsec との比較による予備評価実験において，提案手法では TCP ヘッダは暗号化されないが，トランスポートモードで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．

## 5. 提案手法の iSCSI ターゲット側における実装

予備評価実験においては，提案システムの Target 側における暗号化処理を行っていない．本節では，Target 側における実装について説明する．

従来の Target 側の実装を図 10 に，提案手法の実装を図 11 に示す．提案手法では暗号化・復号化処理を Initiator のユーザ空間におけるミドルウェアとして実装し，Target 側においては，カーネル空間のミドルウェアとして，ともに上位層で実装している．

提案手法の Target 側では，3DES アルゴリズムを使用して，暗号化/復号化処理プログラムを実装し，それをカーネルモジュールとして構築してカーネル空間における処理を行っている．ただし，本稿の実験で実装している 3DES は，IPsec の 3DES とアルゴリズムは同じであるが，実装コードは独自のものを使用しているため異なる処理となっている．

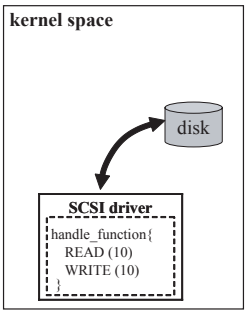


**図 10** 従来の Target 実装

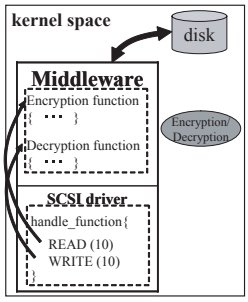


**図 11** 提案手法の Target 実装

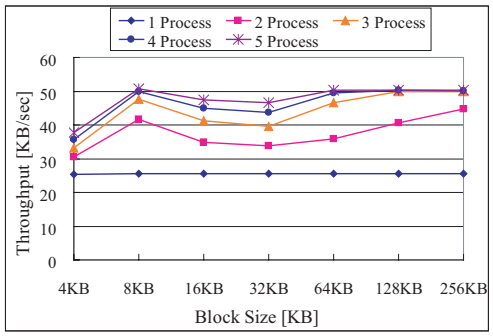


**図 12** 複数プロセスによる iSCSI シーケンシャルリード時の提案システムのスループット

**表 3** 1 プロセスに対するスループットの向上比率

| プロセス数 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 比率 | 1.000 | 1.465 | 1.722 | 1.812 | 1.864 |

ミドルウェアをカーネルモジュールとして提供することにより，カーネルを再コンパイルすることなく，簡単に暗号化，復号化の機能を利用することができ，また，SCSI デバイスドライバからの独立性を高めることによって，提案手法をミドルウェアとして提供することが可能になる．さらに，SCSI 層よりも上位層で処理を行うことによって，下位層である SCSI 層，TCP/IP 層などの処理内容を把握することができる，これにより，上位層でさまざまな工夫をすることができ，柔軟な処理を行うことが可能になった．

具体的には，従来の手法（図 10）では，Target 側のディスクに格納されているデータを読み出し，SCSI デバイスドライバで処理を行っているだけであるが，本稿の実装（図 11）では，ミドルウェアとして自作カーネルモジュールを起動し，SCSI デバイスドライバから暗号化，復号化ルーチンを呼び出すことにより，上位層で処理を行うことを可能にしている．

## 6. 複数プロセスを使用した提案システムの性能評価実験

5 節で説明した Target 側の自作カーネルモジュールを使用し，Target の raw デバイスに対する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．実験環境は表 1 と同じである．本実験では，上位層で一つの暗号化処理サイクルが終了しないうちに，次のデータセグメントの暗号化を行うという暗号化の先処理による最適化の効果を確認するため，複数プロセスを起動して，iSCSI ストレージアクセスを行っている．

本実験システムにおいて，Target 側ではディスクに存在するデータを先読みして暗号化処理する機能を実装していないため，Initiator 側のアプリケーションとして最大で 5 プロセスを起動させ，iSCSI Read コマンドを複数発行している．つまり，Target 側のディスクの異なるアドレスにあるデータを順次読み出し，本稿で実装したカーネルモジュールを用いて各データを連続的に暗号化して転送し，Initiator 側のミドルウェアで復号化を行うことによって，iSCSI ストレージアクセスを最適化する提案手法を模擬している．

### 6.1 実験結果と考察

複数プロセスを起動した場合の iSCSI シーケンシャルリードアクセス時のスループットを図 12 に示す．1 プロセスを起動した場合，ブロックサイズの違いによるスループットの変化はほとんど見られなかった．表 3 は，ブロックサイズ 4KB から 256KB までのスループット値を平均した場合における 1 プロセスに対する複数プロセスのスループット向上比率である．これより，2 プロセスを起動すると約 1.5 倍，3 プロセスでは約 1.7 倍にスループットが増加することがわかる．4 プロセス以上になると約 1.8 倍程度となり，プロセス数を増加させてもそれ以上スループットは増加しない．これは CPU 使用率の限界に達しているためであると考えられる，本実験においては，自作カーネルモジュールの暗号化処理時間が通信処理にかかる時間と比較して相対的に長いため，プロセス数の増加による性能向上はあまり大きくはなかった．しかし，高遅延なネットワーク環境において，あるいはより高速な CPU を用いた場合には，提案手法に基づく暗号化の最適化を行うことによって，iSCSI シーケンシャルリードアクセスの性能がより向上すると考えられる．

1 プロセスを起動した場合の iSCSI シーケンシャルリードアクセス時の CPU 使用率を図 13 に示す．同図は，Initiator 側において，Linux の iostat コマンドを用い，“kernel” と “user” の内訳で合計の使用率を 1 秒ごとに測定したものである．図 13 より，Initiator 側では，ユーザ空間で実装を行っているため，user の CPU 使用率が大部分を占めていることがわかる．一方

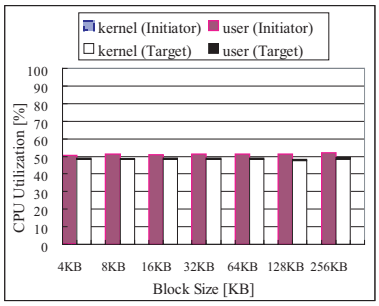


**図 13** 1 プロセスにおける kernel と user の CPU 使用率

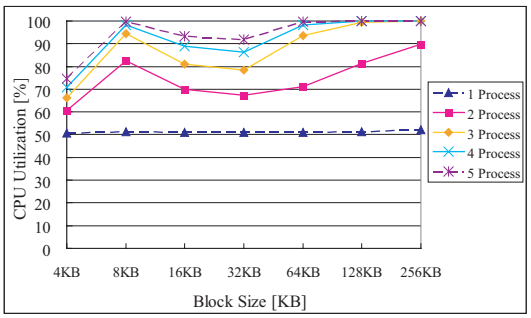


**図 14** 複数プロセスにおける CPU 使用率（Initiator）

**表 4** 複数プロセスにおける CPU 使用率の平均 (Initiator)

| プロセス数 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 平均（%） | 51.241 | 74.623 | 87.597 | 91.760 | 94.166 |

Target 側では，カーネルモジュールとしてカーネル空間で提案手法を実装しているため，kernel の CPU 使用率が大部分を占めた．また，1 プロセスを起動した場合においては，ブロックサイズの違いによる CPU 使用率の変化はほとんどなく，ブロックサイズに依存しないことがわかった．

図 14 は，Initiator 側で複数プロセスを起動して，同様にシーケンシャルリードアクセスを行った場合の CPU 使用率である．測定は，Initiator 側と Target 側の両方で測定を行ったが，両者にはほとんど差がなかった．測定結果より，プロセス数を増加させると CPU 使用率が高くなることがわかった．各プロセス数において，ブロックサイズ 4KB から 256KB までの CPU 使用率の平均を表 4 に示す．1 プロセスのみを起動した場合は，約 51%であり，2 プロセスの場合は約 75%に増加することがわかった．また 4 プロセス以上を同時に起動した場合は，約 90%以上の CPU 使用率になっており，CPU 性能の限界に達している．

### 6.2 スループットのモデル化

提案手法の性能評価を行う際の指標として，シーケンシャルリードアクセスにおけるスループットのモデル化を行った．

Initiator から SCSI Read コマンドを Target 側に転送し，暗号化されたデータが復号化されるまでを 1Cycle とする．図 4 より，1Cycle に要する時間（1CYCLE_TIME），データ転送時間（TRANSFER），RTT（Round Trip Time），暗号化時間（ENC），復号化時間（DEC）の関係は以下の通りである．

$$1CYCLE\_TIME = RTT + TRANSFER + ENC + DEC \quad (1)$$

データ転送時間は，転送データサイズ（DATA），下位層のスループット（SOCKET）を用いて以下のように表される．

$$TRANSFER = \frac{DATA}{SOCKET} \quad (2)$$

ここで，下位層のスループットとは，iSCSI 層以下のスループットであり，iSCSI を用いない単純なソケット通信における値である．ブロックサイズ 256KB の場合を例にとると，実験により下位層のスループットを測定したところ，約 58.106MB/sec であるため，データ転送時間は，4.302ms となる．

また，暗号化時間（ENC）は，転送データサイズ（DATA），暗号化速度（ENC_TH）から，

$$ENC = \frac{DATA}{ENC\_TH} \quad (3)$$

である．暗号化と復号化の 3DES アルゴリズムはほぼ同じであるため，復号化時間（DEC）も同様に表され，ここでは暗号化時間とほぼ同じ値となると予測される．以上より，1 プロセスによりシーケンシャルリードアクセスを行った場合に予測されるスループットは，ブロックサイズ（BLOCK）を用いて次の式でモデル化することが可能である．

$$THROUGHPUT = \frac{BLOCK}{RTT + \frac{BLOCK}{SOCKET} + \frac{BLOCK}{ENC\_TH} + \frac{BLOCK}{DEC\_TH}} \quad (4)$$

同様に測定により，RTT=0.392ms，復号化速度=50.461KB/sec であったため，(4) 式によって，予測されるスループットを計算すると，ブロックサイズ 256KB の場合のスループットは 25.219KB/sec となる．よって，図 12 の実測値 25.560KB/sec と比較し，このモデル化がほぼ正しいことが実証された．

一方，通信の待ち時間の間に別のデータの暗号化処理をする最適化を行った後のスループットモデルは，図 5 より，暗号化処理に復号化処理が隠蔽されるため，

$$THROUGHPUT = \frac{BLOCK}{RTT + \frac{BLOCK}{SOCKET} + \frac{BLOCK}{ENC\_TH}} \quad (5)$$

とモデル化することができる．そこで，(5) 式におい

て，ブロックサイズ 256KB の場合について計算すると，50.414KB/sec となった．図 12 のブロックサイズ 256KB，2 プロセスの場合の実測値 44.641KB と比較して，この値は最適化処理後のモデルにかなり近い性能が出ているといえる．よって本実験により，暗号化処理をオーバーラップさせ，最適化処理を行った場合，本稿の提案手法が有効であることがわかった．また十分な性能の CPU を導入すれば，CPU に制限されず，プロセス数をさらに増大させた場合にスループットの向上が期待できる．さらに，本実験では，RTT とデータ転送時間は，暗号化処理時間と比較して相対的に 0 に近い値であるが，高遅延環境においては，通信の待ち時間が大きくなり，暗号化の最適化を効果的に行うことができるため，本稿の提案手法による性能向上の割合が大きくなると考えられる．

## 7. 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，IPsec の代わりに上位層で暗号化し，暗号化処理の最適化を行う手法を提案した．また提案システムの実装を行い，性能評価実験を行った．さらに，スループットのモデル化を行い，複数プロセスを起動させた実験の測定結果から，提案手法が有効であることがわかった．

今後の課題として，構築した提案システムにおいて，高遅延環境におけるストレージアクセスや，シーケンシャルライトアクセスについても考慮し，アプリケーションレベルのベンチマークを使用した総合的な性能評価を行う．

**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) Storage Networking Industry Association, http://www.snia.org/.
2) iSCSI Draft, http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt.
3) Ng, W. T., Hillyer, B., Shriver, E., Gabber, E. and Ozden, B.: Obtaining High Performance for Storage Outsourcing, *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, pp. 145–158 (2002).
4) Sarkar, P. and Voruganti, K.: IP Storage: The Challenge Ahead, *Proc. Tenth NASA Goddard Conference on Mass Storage Systems and Technologies*, pp. 35–42 (2002).
5) 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).
6) 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).
7) Radkov, P., Yin, L., Goyal, P., Sarkar, P. and Shenoy, P.: Performance Comparison of NFS and iSCSI for IP-Networked Storage, *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, pp. 101–114 (2004).
8) Meth, K.Z. and Satran, J.: Design of the iSCSI Protocol, *Proc. 20th IEEE Symposium on Mass Storage Systems and Technologies (MSS '03)*, pp. 116–123 (2003).
9) Tang, S.-Y., Lu, Y.-P. and Du, D. H. C.: Performance Study of Software-Based iSCSI Security, *Proc. First International IEEE Security in Storage Workshop*, pp. 70–79 (2002).
10) 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, 電子情報通信学会全国大会, B-16-10, p. 619 (2004).
11) 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, 電子情報通信学会論文誌, Vol. J87-D-I, No. 2, pp. 216–231 (2004).
12) 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).
13) 神坂紀久子, 山口実靖, 小口正人: IP-SAN を利用したセキュアなストレージアクセスにおける性能向上手法の提案と検討, 第 3 回 情報技術レターズ, Vol. 3, No. LD-003, pp. 59–61 (2004).
14) Kamisaka, K., Yamaguchi, S. and Oguchi, M.: Performance improvement of an iSCSI-based secure storage access, *the 16th IASTED International Conference on Pararellel and Distributed Computing and Systems (PDCS 2004)* (Gonzalez, T.(ed.)), IASTED, pp. 522–527 (2004).
15) InterOperability Lab in the University of New Hampshire, http://www.iol.uhn.edu/consortiums/iscsi/.
16) FreeS/WAN Project, http://www.freeswan.org/.