2003－HPC－94 (5)
2003／6／13

IPv6 によるスケーラビリティに優れたセキュアグリッド環境の構築

史　宏宇*, 武田　伸悟*, 長谷川　一郎†, 伊達　進*, 水野（松本）由子‡, 下條　真司§
{shi,stakeda,ichiro,date,yuko}@ais.cmc.osaka-u.ac.jp,
shimojo@cmc.osaka-u.ac.jp

概要　グリッドはネットワーク上の PC や WS などのプロセッサ資源やデータストレージなどといった多様な要素を動的に統合可能とし、多数のユーザで共有できる一つの統一システムを構築する技術である。現在、高度なコラボレーション形態やコスト削減を実現する新しい機能を備えたグリッドは科学の分野だけではなく、ビジネスへの応用といった観点からも利用されつつある。このような背景から、ユーザのグリッドへのアクセスは、安全で頑強であることが要求される。本研究では，IPsec (IP Security) と IKE (Internet Key Exchange) を基盤技術としてセキュリティソリューションをグリッド環境に応用することにより、安全なグリッド環境の提案と構築を行った。さらに、グローバルコンピューティングへの対応を考慮し、そのグリッド環境を IPv6 ネットワーク上で構築した。本論文では、IPv6 ネットワークにおけるセキュアなグリッド環境の構築技法とそのメカニズムについて論じる。

An IPv6-based Secure Grid Environment with High Scalability

Hongyu Shi*, Shingo Takeda*, Ichiro Hasegawa†, Susumu Date*, Yuko Mizuno-Matsumoto‡, Shinji Shimojo§
{shi,stakeda,ichiro,date,yuko}@ais.cmc.osaka-u.ac.jp,
shimojo@cmc.osaka-u.ac.jp

Abstract The Grid is a technology that can dynamically integrates various elements, such as processor resources including PC and WS and data storage, on a network to build a unified system that can be shared among many users. Now the Grid technology equipped with new functionalities to realize a form of advanced collaboration and to achieve dramatic cost reduction has been used not only for a scientific field but also for business. Under such situation, required is that users can access to the Grid in a safe and robust way. In this research, a secure Grid environment is proposed and constructed through the application of a suite of security solutions to the Grid environment. The solutions have adopted IP security (IPsec) and Internet Key Exchange (IKE) as base technology. Furthermore, in this paper, our secure Grid environment is realized on IPv6 network, so that our Grid environment addresses the shortage of IPv4 addresses. Throughout the paper, we describe the techiniques to build a secure Grid environment on an IPv6 network and its mechanism in detail.

1. 序論

グリッドコンピューティングは、今日の多種多様なネットワークおよびコンピューティング資源を、密接かつシームレスに統合されたコンピューティング環境へと発展させるための技術である。グリッドコンピューティングによって、数多くの分散コンピューティング資源は結合され、本質的に単一で大規模な仮想コンピュータへと進化する。グリッドによって、企業など様々な組織は地球規模に広がるインターネットや組織内イントラネットのデータ、CPU、ストレージ、およびその他あらゆる異種コンピューティング資源を利用することができる。近年グリッド技術が成熟するにつれて、グリッドは製薬、生物情報科学、医学などのような巨大な計算能力を必要とする多くの学問・技術開発分野で利用されているが、高度なコラボレーションやコスト削減を実現する新しい機能を備えたグリッド技術はビジネスの世界にも利益をもたらす。

現在、多くのプロジェクトがグリッドコンピューティングのために必要なツールを提供している。その中で、Globus は代表的なツールであり、グリッド環境構築の事実上の標準になりつつある。Globus を

* 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
† NEC システムテクノロジー
NEC System Technologies, Ltd.
‡ 大阪城南女子短期大学幼児教育科
Advanced Diploma in Child Education, Osaka Jonan Women's Junior College
§ 大阪大学 サイバーメディアセンター
Cybermedia Center, Osaka University

用いることで、インターネット上の複数の計算資源を同時に利用することが可能となる。

しかし、近年のグローバルコンピューティング時代への急速な移行を鑑みれば、現在の 32 ビットで構成されるインターネットアドレス空間ではネットワークアドレス不足という深刻な問題を引き起こす。このような問題に対して、現在インターネットアドレス空間を拡大する手段として NAT (Network Adddress Translation) が主に使用されている。しかし、NAT はその性格上、送受信ホストの通信を仲介する。このことは、送受信ホスト間でやりとりされるデータの機密性の欠落、送受信ホスト間認証などといったセキュリテイ要件でのリスクを高めることを意味する。実際、創薬過程で必要となる化合物情報や個人情報を含む医療データなどの高機密性データの解析へグリッドを応用することへの期待の高まりは、高いデータ機密性をはじめとするセキュリティ対策の重要性についての議論を産み出している。このようなことを考慮すれば、最新インターネットプロトコル IPv6 のグリッドへの導入は必然的であるといえる。

しかし、Globus は開発途上ということもあり現段階ではグリッドコンピューティングに参加するユーザおよびホストの認証に焦点がおかれ、デフォルトセッティングではデータの機密性を十分に確保することはできない。その理由として、データ機密性を保持するためのデータ暗号化などのセキュリティ対策が、本来大量データ処理を実現するグリッドの計算パフォーマンスを低減させることも一要因である。このように、現状でのセキュリティ対策の欠如は Globus を用いた医療データなどの機密性の求められるデータの解析が妨げられる一因となっている。

上述の考察から、グリッド上で高機密データを扱いたいという要求をもつ研究者及び科学者は頑強なセキュリティ対策と高い計算パフォーマンスという相反する要求をもつといえる。本研究では、これら２つの相反するユーザ要求のバランシングを可能とする。セキュリティ機構をグリッドへシームレスに統合することにより、ユーザ利便性の高いセキュアなグリッド環境の実現を目的とするより具体的には、グリッドの標準実装である Globus Toolkit を IPv6 化し、21 世紀のグローバルコンピューティング時代いち早く対応するとともに、Internet Protocol Security (IPsec) の提供する暗号化機能をグリッドに統合することにより、セキュリティとパフォーマンス要求のバランスのとれるセキュアなグリッド環境の構築する。

本論文の構成は以下の通りである。次章で、セキュアなグリッド環境を構築するためのいくつかの基盤技術について述べる。第 3 章で、セキュアなグリッド環境の設計について述べる。第 4 章では、Globus 2 の IPv6 対応についての実装手法について詳細に述べ、第 5 章では、評価について述べ、第 6 章では、本論文をまとめる。

2. 基盤技術

2.1 Globus Toolkit

Grid は 1990 年代の半ば頃に生まれた概念で、高度な科学技術のための分散コンピューティング基盤を意味する [1]。今日までに、Grid 概念を実現とすべく、多様な実装が異なるアプローチから開発されてきた。Legion [2]、Ninf [3] はその一例である。一方、Globus Toolkit はアプリケーションプログラミングインタフェース (API) の提供というアプローチから、Globus プロジェクトによって開発が進められている実装の一つである。Globus Toolkit は 1990 年代の半ばから、グリッド研究の分野において、重要な役割を占めており、グリッドを実装するデファクトスタンダードになりつつある。

Globus Toolkit は、複数の組織を仮想研究室 (Virtual Laboratory) 化する。言い換えれば、物理的な組織構造とは関係なく、自由に論理的な組織構造を動的に構築することが可能となり、同一の目的をもつ組織間でのデータ、プロセッサ資源などの共有利用を実現する。Globus はそのような仮想研究室化を実現するために必要とされるサービスの集合とそれらに対応する API を提供する。グリッド環境上で頑強かつ自由度の高い認証サービスを提供する Grid Security Infrastructure (GSI) [4]、リソースとプロセス管理のための Globus Resource Allocation Manager (GRAM)、リモートファイル操作のための Global Access to Secondary Storage (GASS)、情報管理のための Monitoring and Discovery Service (MDS) などのサービスはその一例である。ユーザはこれらのサービスとそれらに対応する API を用いて、ネットワーク上に分散するプロセッサ資源、データ資源へのアクセスを容易に行うことが可能となる。

2.2 Grid Security Infrastructure (GSI)

Globus によって提供されるサービスの中で、GSI は特にセキュアなグリッド環境を構築する際に重要な役割を占めている。GSI の最も重要な機能はシングルサインオン機能である。これによって、ユーザが一度パスフレーズを入力するだけで、透過的に様々な計算資源とグリッドサービス（ジョブの投入、ファイル転送、情報検索など）を利用できる。この機能は公開鍵基盤（Public Key Infrastructure：PKI) を基盤技術として実現されている。PKI では公開鍵の信頼性は認証局 (Certificate Authority:CA) によって保証されるため、繁雑な鍵交換をユーザや管理者が行う必要がなく、大規模なシステムでは PKI は必要不可欠な技術となっている。GSI は、認証局によって発行される公開鍵の電子証明書を使用してユーザだけでなく、計算資源の相互認証法を提供し、Distinguished Name (DN) でユーザとリソースを識別する [5]。証明書は x.509v3 のフォーマットでエンコードされ、発行者、サブジェクト、有効期限、発行者によって署名された公開鍵などの情報が含まれており、GSI では、これらの利用を用いて、ユーザと計算資源が

CA によって発行された証明書を用いて相互認証を行っている。例えば、遠隔計算資源上でジョブを実行するために、ユーザは遠隔計算資源上で実行されている gatekeeper と相互認証を確立しなければならない。この gatekeeper はジョブリクエストの監視やジョブの管理といったリソース管理を行う。

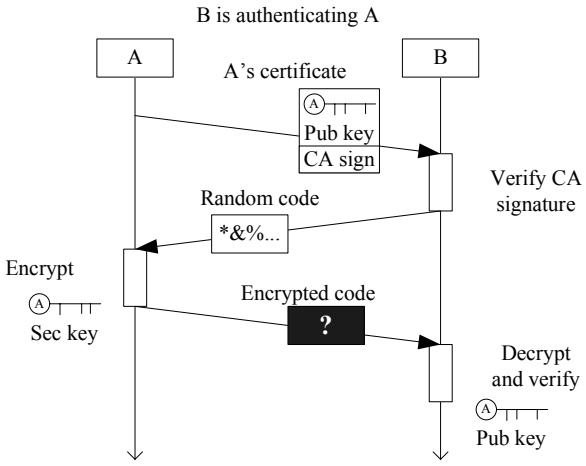

図 1: Authentication in GSI

図 1 は GSI で行われる、2 者間での基本的な相互認証メカニズムを示す。A と B の間で相互認証を行うケースを想定する。まず、A が証明書を B に送る。B が証明書を受け取った後、Globus CA の電子署名をチェックすることにより、送られてきた証明書が有効であるかどうか、改ざんされていないかを確認する。その後、B はランダムなメッセージを生成し、A に送信する。A はこのランダムなメッセージを自分の秘密鍵を使用して暗号化し、B に送り返す。B は A の証明書に含まれる A の公開鍵を使用して、暗号化されたメッセージを復号する。B は復号されたメッセージを元のランダムなメッセージと照合し、もし一致すれば、A の身分を確認できる。A は B の身分を確認するため、同じように逆の手順を取る。ユーザがジョブを投入するため、この例のようにユーザ (プロキシ) と gatekeeper 間で相互認証を行う。

2.3 Internet Protocol Security (IPsec)

IPsec はネットワーク層でデータの機密保護と認証を提供する強力な標準である。これを用いることにより、ユーザは IP パケット・レベルでのデータセキュリティを確立できる。ネットワーク層で提供されるセキュリティ対策は、SSL、SSH、HTTPS などのアプリケーション層で提供されたセキュリティ対策と比較して多くの利点を持つ。SSL や SSH などは強力な認証機能により、セキュアなネットワーク環境を構築できるが、これらは OSI モデルにおけるアプリケーション層で機能する。このことはこれらのプロトコルにより提供されるセキュリティメカニズムを Globus に統合するには、既存のアプリケーションの通信と関連する部分に変更を加える必要があることを意味し、ユーザへの負担も大きい。しかし、一方で、IPsec においては、上位のアプリケーションに関係なく、また透過的にデータセキュリティ機能が提供され、ユーザへの負担を最小とできる。

IPsec では、「認証ヘッダ」 (AH) と「暗号ペイロード」(ESP) の二つのヘッダの使用によりセキュリティ機能を提供する。「認証ヘッダ」は、IP アドレスを含めたパケット全体のハッシュをとることにより、IP パケットにメッセージ認証の機能を提供する。「暗号ペイロード」は、暗号化やトンネリングの機能を提供し、IP パケットに機密性と完全性を提供する。これらの仕組みは独立して動作するので、IP パケットのメッセージ認証の機能を利用したい場合は AH を利用し、データの暗号化やトンネリングの機能を利用したい場合は ESP を利用する。また、両方を組み合わせて利用することも可能である。これらのヘッダが IPv4 ではオプションとして使われているが、IPv6 では必須となっている [6]。

IPsec では、暗号化や認証に使用するアルゴリズムを規定していない。このため、様々な種類のアルゴリズムの中から利用したいものを選択することが可能となる。しかし、相手側がどのアルゴリズムを使用して暗号化したのか、どのアルゴリズムを利用して認証しているのかということが分からなければパケットを受け取ることができない。このような情報を保持するために、IPsec では Security Association (SA) を利用する。SA には、使用する暗号化アルゴリズムの種類、暗号化アルゴリズムのモード、認証アルゴリズムの種類、転送モード、暗号化鍵、認証鍵などの情報が保持される。IPsec では、Security Association Database (SAD) と Security Policy Database (SPD) との二つのデータベースを用いて SA の情報をシェアしている。SAD には Security Parameter Index (SPI) によって管理される様々な SA が格納されている。SPD には送信パケットと受信パケットをどう処理するかに関するセキュリティポリシが格納されている。図 2 では、A-B 間で IPsec 通信の確立までの手順を示したものである。最初、A は IPsec 処理を適用するかどうかを決定するために SPD を検索する。IPsec 処理を適用する場合は、A は B の保持する SA のデータベース SAD の中から利用できるものを選択し、SPI の値をそのパケットの認証ヘッダや暗号ペイロードのフィールドに含めて B に送る。

IPsec の重要な点として、IPSec が設定自由度の高いフレームワークを提供していることがある。すなわち、IPsec では、暗号路を確立するために SPD、SAD、SA などからなるフレームワークのみを規定しているため、ユーザは必要とされるセキュリティレベルに応じた暗号化アルゴリズム、ポート、IPsec による暗号路の確立に使われるプロトコル (ESP、AH) を使用することができる。この設定の柔軟性は、多様なセキュリティ・ポリシが存在するネットワーク上にセキュアなグリッド環境を構築するのに非常に有効である。例えば、われわれは、TCP/UDP ポートごとに異なるセキュリティ・ポリシを適用すること

ができる。第 3 章でこの利点を利用するセキュアなグリッド環境の設計とアーキテクチャを詳述する。

2.4 Internet Key Exchange (IKE)

2.3 節に述べたように、IPsec はネットワーク層でデータの機密を保護するためのソリューションを提供する。このソリューションの利点は、設定における柔軟性とそのソリューションがネットワーク層で機能することから、アプリケーションへの変更を最小限にとどめることができる点にある。このような利点を有する IPSec であるが、グリッド上での運用管理を考慮した場合、IPsec の管理コストの低減が非常に重要な研究課題となる。例えば、n 個のホストがお互いに暗号路を確立しようとすると、$n(n-1)/2$ 個の SA の情報を共有するため、サイト管理者は設定しなければならない。インターネット上に分散する数百、数千、あるいはそれ以上の数の計算機を集約し、計算パフォーマンスを引き出そうとするグリッドにおいては、その管理コストは非常に重要な問題となる。また、IPSec の静的な管理下では、一つの SA に変更が生じた場合に、すべてのサイト管理者がサイト管理下にある計算機すべての変更を最新状態に設定しなければならない。この運用管理作業は、煩雑であるだけでなく設定ミスなどにつながり、結果としてセキュリティリスクを高める結果となる。

Internet Key Exchange (IKE) は、自動的に IPsec の SA をネゴシエートし、手動による事前設定なしに IPsec の安全なコミュニケーションを可能にするプロトコルである。二つのホストが IPsec による暗号路を確立する際に、まず IKE によって、相互認証を行ってから、安全な通信路の保護の下で、IPsec SA の交換を行う。IPsec と IKE の併用により管理コストを大幅に削減できる。IPsec と IKE の併用はセキュアなグリッド環境の構築に優れた機能を提供するが、ここにもまた一つの問題が存在する。今日、通常では、IKE のペア間の相互認証は共有秘密鍵を用いて行われている。IPsec と IKE の併用を行う場合にも、グリッド上の計算資源すべてに対して、管理者が IKE の相互認証のため、予め共有秘密鍵を交換しなければならないという運用管理の問題がある。これはスケーラブルなセキュアなグリッド環境を構築する際に大きな障害になる。

3. Globus 2 への IPv6 の実装

3.1 IPv6 の必要性

序論で述べたように、グリッドにおいては、ホスト間で頻繁に相互通信が行われ、また発生する通信に対してもセキュリティ対策を施す必要がある。パケットレベルで頑強なデータセキュリティを実現するためには、IPv6 アドレスは必要不可欠となる。また、このことは、ユビキタス、Peer-to-Peer、グリッドなどに代表されるグローバルコンピューティングという観点からも説明できる。

現在、Linux、FreeBSD、Windows2000 などのような多種のオペレーションシステムが IPv6 をサポートしている。IPv6 ネットワークへの移行はこれから加速されると考えられる。実際に、多くのプロジェクトにおいて、IPv6 接続テストを実施したり、IPv6 ネットワーク上で実証実験を行っているのが現状である。

さらに、近年グローバルコンピューティングへの期待が著しい。それはユビキタス、P2P、グリッドへの取り組みからも説明できる。グローバルコンピューティングでは、インターネットに接続している様々なリソースを統合し、安全かつ拡張性の高いアクセス手法が要求される。これらの資源の最適化・仮想化によって、様々なビジネス課題を迅速に解決したり、膨大な計算を必要とする研究やデータ分析を行うことが可能となるため、グリッドへ期待するユーザの数も急速に増加している。このような観点から鑑みても、IPv4 のアドレスの不足は深刻な問題となる。

以上のような考察から、本研究では、グリッドのデファクトスタンダード実装である Globus を IPv6 に対応させる。実装プラットホームとしては、比較的に安定し高度なプロトコルスタックを展開する FreeBSD をプラットホーム OS とすることも考えられるが、現状の最新バージョンである Globus は FreeBSD を正式なプラットホーム OS としてサポートしておらず、動作確認がとれていない。そのような考察から、われわれは Globus での正式サポートが得られている Linux をプラットホーム OS として、選定当時最新であった Globus2.2.3 を v6 化することに着手した。

3.2 Globus 2.2.3 への IPv6 の実装

2.1 節で述べたように、Globus はインターネット上でグリッドを構築するために必要となる様々なサービスとその API を提供する。しかし、Globus によって提供されるサービスの中で実装されている通信に関する部分の多くは、globus_io や Nexus のような通信機能を提供するモジュールに集約されている。それゆえ、これらの通信機能を提供するモジュールを拡張することによって、Globus の提供するサービスを IPv6 ネットワーク上で運用することができることが可能となる。以下のリストは Globus2.2.3 において、変更を必要とする 15 個のソースファイルを示す。

- Globus_IO
  - globus_io_attr.c
  - globus_io_securesocket.c
  - globus_io_socket.c
  - globus_io_tcp.c
  - globus_io_udp.c
- GASS
  - globus_gass_cache.c
- Communication
  - nexus/pr_tcp.c
  - nexus/nexus_resource.c
  - nexus/nexus_resource.h
- Resource Management

  - globus_gatekeeper.c
  - globusrun.c
- Information Service
  - globus_openldap-2.0.22/ pkg_data_src.gpt
- Miscellaneous
  - globus_libc.c
  - javasasl.c
  - cyrus-sasl-1.5.27-patch

これらのサービスを IPv6 ネットワーク上でも実行できるように拡張するため、IPv6 に対応したソケットを使う必要がある。構造体定義の変更は "struct sockaddr_storage"に変更したが、複数のアドレスを扱う必要がある場合は "struct addrinfo *" を用いた。また、IPv4 に依存した部分を取り除くため、getaddrinfo(3) などの IPv6 に対応した関数を使用した。getaddrinfo(3) に AI_PASSIVE フラグをつけて使用すれば、待ち受ける必要があるアドレスすべてを得られ、複数のソケットで待ち受けるように拡張することができる。以下の例に示すコードは、globus_gatekeeper.c の中で使われ、コマンドライン引数によって、適切なプロトコルを選択する部分である。

```
static char * listener_family = "IPv4-6"
......

"[-listener_pf] [IPv4/IPv6/IPv4-6(default)]",
......

if(strcmp(listenerpf, "IPv4") == 0)
     hints.ai_family = PF_INET;
else if(strcmp(listenerpf, "IPv6") == 0)
     hints.ai_family = PF_INET6;
else hints.ai_family = PF_UNSPEC;

hints.ai_flags = AI_PASSIVE;
hints.ai_socktype = SOCK_STREAM;
getaddrinfo(NULL, strport, &hints, &listen_addrs);
....
```

gatekeeper では、AF_INET ワイルドカードアドレス と AF_INET6 ワイルドカードアドレスのどちらか、あるいは、両方を待ち受ける。ディフォルトの場合は両方を待ち受ける。このように、IPv4/IPv6 デュアルスタックノード上でも動作するように拡張を行った。

4. IPsec 機能を GSI への統合の設計

2.3 節で述べたように、IPsec を IPv6 での必須機能として規定している。本章では、IPsec と GSI の連携統合技法について述べる。まず、IPsec を GSI に統合することにより、解決された二つの問題点について述べる。次の節では、セキュアなグリッド環境のアーキテクチャとメカニズムについて、詳述する。

4.1 設計の方針

IPsec の機能を GSI への統合において、以下の二つの点を考慮にいれるべきである。

1. Globus に変更を加えない。
   2.1 節で述べたように、Globus はすでに、グリッドを実現するデファクトスタンダードになっている。そのため、数多くのグリッドプロジェクトでは、Ninf-G のような独自のアプリケーションシステムやグリッドミドルウェアを構築する部品として、Globus を採用している。したがって、IPsec 機能を GSI に統合するため、Globus によって、提供される API の引数の数のような仕様を変更されるのが望ましくない。実際に IPsec 運用により、Globus への変更を最小限に抑えることができる。
2. 計算パフォーマンスとデータ機密性の間にトレードオフ要求の均衡をとるための選択性を提供。
   グリッドにおいては、データ機密性に対する要求が様々に異なるデータが扱われることが想定される。また、グリッド上で扱われるデータのサイズも数キロバイトから数ペタバイトまで広範である。この状況は、データ量とデータの機密性といった基準に従って、機密性の保護強度を切り替えできる機能を要求する。例えば、医師が患者から測定したデータを分析したい時、測定データと患者の個人情報からなるデータファイルに含まれる患者の名前がセキュアな通信で送られて欲しいが、測定データ自身に関しては、パフォーマンスの低下を防ぐために、IPsec で保護される通信を行う必要がないというような状況が想定される。そこで、本研究では、ユーザが計算パフォーマンスと機密性を考慮し、必要とされるデータ機密性に応じて暗号化強度を変更できる仕組みを提案する

4.2 アーキテクチャ

GSI と IKE の認証に gatekeeper の証明書を使用することがわれわれ利用したセキュアなグリッドメカニズムの鍵となる。2.4 節で述べたように IKE の使用により、IPsec SA の共有にかかるコストを削減できるが、ホスト間の IKE における相互認証において鍵管理の問題が存在する。われわれの提案したセキュアなグリッド環境ではこの問題を解決するため、ホスト間の GSI における相互認証と同様に、gatekeeper の証明書を使用することにより、ホスト間の IKE における相互認証を行う。GSI では gatekeeper が自分の証明書を持って、ユーザによって投入されるジョブを実行するコンピューター上で実行されるため、IKE における相互認証に gatekeeper 証明書の使用はシームレスに GSI のセキュリティモデルに統合される。

図 2 は相互認証の観点から、このセキュアなグリッド環境の概観を示す。このセキュアグリッドのメカニズムは以下のように動作する。まず、ホスト間では、Globus CA によって、gatekeeper に発行される証明書を使用し、IKE における認証を行う。その後、IKE では SPD と SAD の情報をベースに、IPsec ホスト間の SA をネゴシエートする。次に、IKE の通信によって、IPsec ホスト間の相互認証を行い、さ

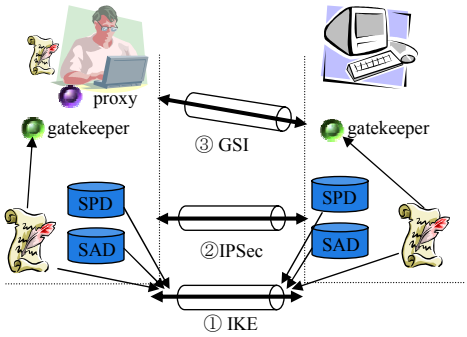

図 2: Secure grid environment

らに IPsec による暗号路を確立する。この暗号路は Globus システムが起動される時に、自動的に確立される。最後に、リモート側でジョブを実行するために、ユーザの代わりに認証を行うプロキシとリモートコンピュータ上の gatekeeper 間で GSI における相互認証を行う。このプロキシはユーザの秘密鍵によって証明され、ユーザの振る舞いをする。最も重要なのは、GSI のアーキテクチャを変更することなくデータ暗号化機能を使用することにある。これは送受信トラフィックを保護するために既存のアプリケーションを変更する必要がないことを意味する。

4.3 ユーザビュー

4.1 節で述べたように、既存のグリッド・アーキテクチャに変更することなく、セキュアなグリッド環境を実現できる。IPsec と IKE の機能を容易に、シームレスに Globus に統合することができる。われわれの提案したセキュアなグリッド環境における GSI と IPsec と IKE の認証メカニズムの統合は一貫性があり、強力である。さらに、このセキュアなグリッド環境では計算パフォーマンスと機密性の間の要求に基づいて、暗号化アルゴリズムの強度を選択することができる。例えば、ホスト A からホスト B へ高機密性データを送信することを仮定し、ホスト A では 20000 番から 25000 番までの送信側ポートに IPsec の ESP モードによりデータが暗号化される設定であったとする。また、一方、ホスト B で 24000 番から 25000 番までの受信側ポートで受信したパケットは反対にデータが復号化される設定であるとする。この場合、ユーザは globus_io_tcp_connect()、globus_io_tcp_listener() といった API に通信で利用するポートを渡すことにより、容易にデータ機密性を確保できる。したがって、ユーザは Globus の提供する API を用いて暗号路を確立するために、IPsec で使用される TCP 或いは UDP のポート番号を知る必要がある。よって、既存の Globus の提供した API を用いたアプリケーションでは、IPsec によって保護されるポートの番号だけを置き換えればよい。

5. まとめと今後の課題

本論文では、IPsec と IKE を用いて IPv6 上のセキュアなグリッド環境について論じた。Globus Toolkit を IPv6 に対応させることによって、巨大なアドレス空間を利用できるグリッド環境の構築を可能にした。また、この環境では、GSI、IPsec、IKE という三つのセキュリティ技術をシームレスに統合することによって、一貫性のある、ロバストなセキュリティを提供できた。さらに、この環境によって、ユーザに計算パフォーマンスとセキュリティ強度のバランスを取ることができる。また、既存の Globus アーキテクチャに変更を加えない。これにより、ユーザは IPSec で利用するポート番号を globus_io_tcp_connect() といった Globus API に与えることによって、計算パフォーマンスとデータ機密性のバランシングを行いつつアプリケーションを構築できる。

今後の課題として、実際広域環境上に本論文で述べたセキャアグリッド環境を適用し、暗号化による通信のパフォーマンスの測定などの評価をアプリケーション構築の観点から行いたいと考えている。また、実装した IPv6 に対応した Gloubs 2 を FreeBSD にも対応させる予定である。

謝辞

本研究に協力していただいた NEC の方々に感謝の意を表す。本研究は科学研究費補助金特定領域研究 (C)「Grid 技術を適応した新しい研究手法とデータ管理技術の研究」 (13224059) の助成を受けて行われた．また，本研究は文部科学省科学技術振興費主要 5 分野の研究開発委託事業の IT プログラム「スーパーコンピュータネットワークの構築」の一環として実施された研究成果の一部である．

参考文献

[1] I. Foster, C. Kesselman,(eds.)., ”The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann, (1999).

[2] Anand Natrajan, Marty A. Humphrey, Andrew S. Grimshaw, ”Capacity and Capability Computing Using Legion”, Lecture Notes in Computer Science, Vol. 2073, pp. 273-280, (2001).

[3] H. Nakada, M. Sato, and S. Sekiguchi, ”Design and implementations of Ninf: towards a global computing infrastructure”, Future Generation Computing Systems, Metacomputing Issue, Vol. 15, Issues 5-6, pp. 649-658, (1999).

[4] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, ”A security architecture for computational Grids”, Proc. 5th ACM Conference on Computer and Communications Security, pp. 83-92, (1988).

[5] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, V. Welch, ”A National-Scale authentication infrastructure”, IEEE computer, Vol. 33, No. 12, pp. 60-66, (2000).

[6] Shelia Frankel, ”Demystifying the IPSec”, Artech House, (2001).

.