グリッドでKISS!
～安全・安心なグリッドを目指して～

峯尾真一

Ⅰ．はじめに

グリッドはネットワーク上のIT資源を仮想化し，誰にでも簡単にアクセスできるようにします。例えばスーパーコンピュータ（スパコン）の利用方法もKISS（Keep It Simple and Sweet.）にできるということです。

もちろん必ずしも簡単なことが良いことではありません。マーフィーの法則的に言えば「サルでも間違いなく使用できるシステムを考案すると，サルだけが使用したがる。」ということになるからです。

しかし，自動車のオートマチックトランスミッションの場合はどうでしょうか。車好きの人の中には，オートマを軽蔑する人もいますが，簡単にすることでその恩恵が社会に大きく広がり，開発競争や大量生産により，よい車が安く手に入ることになれば，みんなが幸せになれるということになります。

グリッドの開発者は，スパコンを含むIT資源を車のオートマなみに簡単に利用できるようにしたいと考えています。

ただし簡単に使えることと矛盾する問題があります。それはセキュリティの確保です。そこで，ここではグリッドがどのようにセキュリティを確保しているかについて，NAREGIでの実装を例として解説したいと思います。

Ⅱ．グリッドの進化

グリッド開発の歴史の中で大きなターニング･ポイントとなったのが，グリッドサービスを“ステートフル”なWebサービスとして定義したOGSA（Open Grid Services Architecture）の出現でした。[1] これにより，グリッドは“Webサービス標準仕様”という計算機世界の共通言語を手に入れたことになりますが，セキュリティの面でも大きな影響を受けています。

WebサービスではSOAPプロトコル上に各種の仕様が定義されています。グリッドのセキュリティに関係が深い部分を図示すれば“図1 Webサービスのセキュリティ関連標準”の通りです。

Webサービスに関連する標準化は，W3CやOASISで進められています。基本的にはSOAPを含むWebに関する基礎技術はW3Cで，WS-*のような，その上位のアプリケーション寄りの標準化はOASISで扱うことになっていますが，WS-PolicyはW3Cで扱われているというように少し複雑です。Webサービスの標準化はまだ途上であり，今後その進捗を注意して見ていく

必要があります。

以下にそれぞれを簡単に説明します。

WS-Security

メッセージの暗号化や署名の実施を行います。

WS-SecureConversation

相互認証，鍵共有，メッセージ認証や管理を行います。

WS-Trust

異なるドメインにて信頼関係の確立を行います。

WS-Policy

ネットワークの到達点（エンドポイント）のセキュリティ要件であり，認証データに対してポリシーを定義します。

WS-Federation

複数ドメイン間での認証情報のやりとりを行います。WS-Security，WS-Policy，WS-Trust，WS-Secure Conversation をベースに実現します。

WS-Authorization

アクセス制御の枠組みです。認証データとポリシーを元に実行権限を決定します。

WS-Privacy

Web サービスでのプライバシー保護を行います。

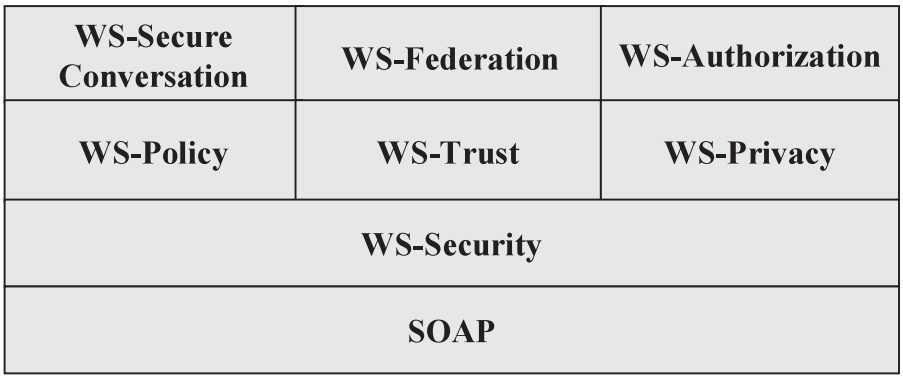

図 1　Web サービスのセキュリティ関連標準

Ⅲ．セキュリティの実現方法

グリッドのセキュリティを考える前に一般的なセキュリティの要件を整理し，それに対しグリッドがどのような対策を取っているかを説明します。

1．一般的なセキュリティ要件

（1）　何はともあれすべてを識別すること：**Identification**（識別）

最も基本的な要件として，管理したいすべての対象を識別することが必要です。すなわち現実世界の実体（例えば利用者やサーバ）に ID という識別子をマッピングすることです。

(2) 次に安全な通信路の確保

一般的に安全な通信の 3 条件と言われるものは次の通りです。

・**Authentication**（認証）

通信相手が本人であることが保証されること。

・**Confidentiality**（秘守性）

他人に盗聴されないこと。

・**Integrity**（完全性）

通信内容が途中で改ざんされないこと。

(3) サービスとしての要件

グリッドを“サービス”と考えると (1) と (2) だけでは不足し次の要件が必要となります。

・**Authorization**（認可）

限定した人にサービスを提供できること。

・**Non-repudiation & Auditing**（事後否認防止 & 監査）

やり取りの証拠が記録できること。

(4) 安全と言える根拠を示すこと

異なる組織間の資源がシームレスに連携可能となることがグリッドの特徴です。そのためには組織相互に安全と言える根拠を示した上で信頼関係を結ぶ必要があります。

2．グリッドにおける対策

上記の要件に対するグリッドでの対策は次の通りです。GSI（Grid Security Infrastructure）については，後で説明します。

(1) 対象の Identification（識別）

PKI（公開鍵暗号基盤）を用いる。具体的には認証局が証明書発行申請者の実在性を確認し，一意の名前を定義します。ただしこの他に将来動向で後述するように新しい仕組みが提案されています。

(2) 通信の Authentication（認証）

GSI を用いる。

(3) 通信の Confidentiality（秘守性）

GSI を用いる。

(4) 通信の Integrity（完全性）

GSI を用いる。

(5) サービスの Authorization（認可）

基本的には GSI で規定されている Grid-mapfile（証明書の一意の名前とその計算機利用アカウントのマッピングファイル）で利用者ごとの認可を行います。また仮想組織管理で

仮想組織ごとの細かい認可を行うことも可能です。

(6) サービスの Non-repudiation & Auditing（事後否認防止 & 監査）

監査証跡の保存等の運用による対策が必要となります。

(7) 安全の根拠

GSI は信頼できる第三者機関としての認証局により安全性を担保しています。一般的なシステム・ネットワークのセキュリティは別途担保されるという前提です。

3．GSI とは何か？

グリッドのセキュリティ基盤である GSI は Globus Toolkit［2］のセキュリティ層として開発されました。提供する機能は次の通りです。

・通信のセキュリティ
・サービスを行う時の相互認証
・認可の仕組み
・権限委譲
・各レベル（コンテナ・サービス・資源）のセキュリティ設定

GSI に特徴的な機能は，PKI の標準である X.509 公開鍵証明書を用いた権限委譲です。そのためにグリッドではプロキシ証明書と呼ばれる巧妙な方式が発明されました。これは本来利用者自身が行うべき認証のためのメッセージのやり取りを権限委譲したプロセスに代行させるものです。

GSI で実現される機能と利用するプロトコルの関係を“図 2 GSI の実装”に示します。

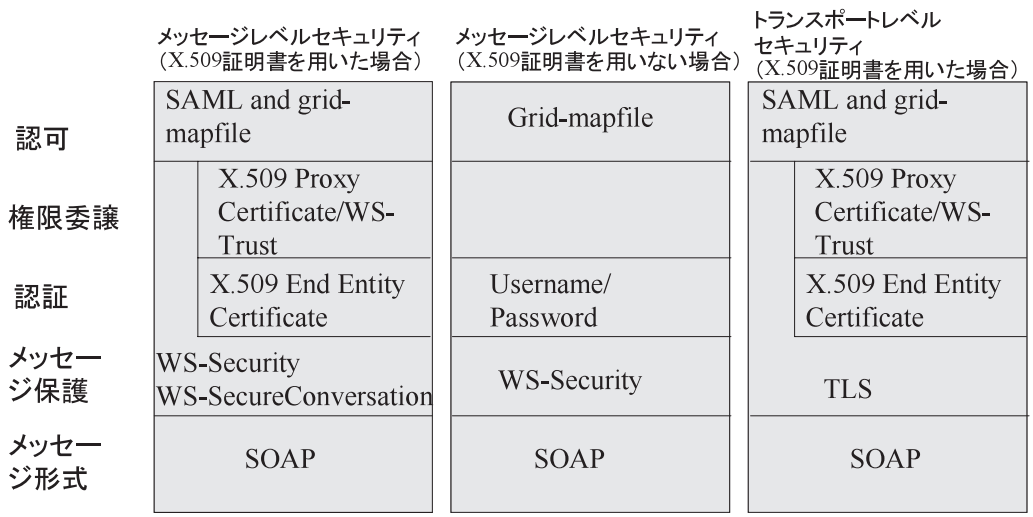

図２　GSI の実装

Ⅳ．仮想組織

参考文献［3］によると，仮想組織（VO: Virtual Organization）とは「同一の目標を達成するために集められた資源とユーザの動的な集合であり，複数の管理ドメインに跨ることが想定されている」と定義されています。ここではまず仮想組織で実現すべき機能を整理し，NAREGI での実装例を紹介します。

1．仮想組織で実現すべき機能

一般的に仮想組織に必要な機能を列挙すれば次の通りとなります。

・セキュリティ機能として，VO の外からの不法なアクセスを排除するためアクセスを管理・制御可能であること。
・ユーザ・資源の管理機能として，プログラムの実行や資源の管理，ロギングなどすべてに及ぶ広範囲な管理機能を持つこと。
・VO ポリシー管理機能として，VO のポリシーに基づいて適切なサービスが提供できること。
・上記の各機能を，管理ドメインを跨いで実現できること。すなわち，現実世界の組織（大学，企業あるいはその部門）や提供されるサービスごとに独立に管理していたユーザとその役割，アクセス権限などを必要に応じて統合し，１つの仮想的なアクセス空間を提供すること。

2．NAREGI での仮想組織管理

NAREGI では仮想組織管理の手段として EU-DataGrid Project で開発された仮想組織管理ミドルウェア VOMS（Virtual Organization Membership Service）を採用しました。その理由は欧州のグリッドプロジェクトである EGEE との間でグリッド環境の相互運用を可能とするためです。

VOMS における VO 関連情報は，プロキシ証明書の X.509v3 拡張情報部分に独自拡張情報として付け加えられ，グリッドのスケジューラや各種計算資源にて参照されます。

仮想組織の基本的な運用ポリシーは次の通りです。

（1） 所有者決定（Ownership Approach）の原則

資源所有者は自分の管理する資源の扱いについてすべての決定権を持ち，また VO 管理者はその VO に属するメンバの登録・削除・属性付与につきすべての決定権を持つ。

（2） VOMS 互換

X.509 属性証明書を利用し，グループ（group），役割（role），資格（capability）に分類した属性定義を行う。

（3） 認可サービスの提供

Globus Toolkit（GT4）コンテナの認可ハンドラから呼び出し可能な認可サービスを提供する。アクセス制御ポリシーの定義は OASIS 標準である XACML（eXtensible Access Control Markup Language）形式で行う。

Ⅴ．将来動向

以下に将来のグリッドセキュリティに関係するいくつかのトピックスを紹介します。

（1） OGSA による標準化の進展

OGSA（Open Grid Services Architecture）は SOAP や WSDL など WEB サービス技術を基盤としてグリッドのすべての機能の Web サービス化を目指しています。セキュリティについても今後の OGSA の動向に注目する必要があります。

（2） IGTF（International Grid Trust Federation）による国際認証連携

グリッドの国際的な相互運用を可能とするためには，グリッドセキュリティの基盤である認証局の相互連携が必要です。そのために IGTF が結成され，ApGrid（アジア太平洋），EUGRID（欧州），TAG（米州）により世界を 3 分割して管理することになりました。日本の認証局は ApGrid PMA の認可を受ければ発行した証明書が世界中で有効となります。

ApGrid PMA とは 2004 年 6 月に設立されたアジア・太平洋地域の PMA（Policy Management Authority：認証局のポリシー及び運用に関する整合性を取る調整機関）です。ApGrid PMA の認可を受けるためには，信頼レベルの高い運用を継続し毎年監査を受けなければなりません。現在認可を受けているのは，NII（NAREGI），AIST，KEK の各認証局であり，NII は将来 NAREGI 認証局を引き継いで“大学間連携のための全国共同電子認証基盤（UPKI）”のグリッド認証基盤として整備して行く予定です。

（3） ID フェデレーションとの連携

グリッドにおける認証には認証局の発行する利用者証明書（X.509 公開鍵証明書）が使われていますが，認証局としてはまず初めに証明書の発行申請をする利用者の認証・認可を行なわなければなりません。そこで，異なる管理ドメイン間で ID 管理を連携させることのできる ID フェデレーションと呼ばれる技術の利用が検討されています。

ID フェデレーションの実現方法にはいろいろな方式が提案されていますが，グリッドの世界で注目されているのが Shibboleth です。Shibboleth とは米国 EDUCAUSE/Internet2 にて 2000 年に発足したプロジェクトであり，SAML，eduPerson 等の標準仕様を利用して認可のための属性交換を行う標準仕様とオープンソースソフトウェアを開発しています。

Shibboleth の基本的な仕組みは“図 3 Shibboleth の動作”の通りです。利用者が① Web サーバにサービスの要求を行うと，② Shibboleth IdP（Identity Provider）にリダイレクトされ，③ Shibboleth IdP が利用者の認証を行い，認証アサーション，属性アサーションと呼ばれる認証結果や認可のための属性に関する情報を発行します。Web サーバはこのアサーションを基に認可判断を行い，④利用者はサービスを受けることができます。

Shibboleth の特徴は，認可判断のために IdP が出す個人情報の内容を利用者自身が制御できることです。すなわちプライバシーを考慮した ID 管理が可能となります。現在 NII では認証局がこの Shibboleth と連携してグリッド用の利用者証明書を発行し，簡易にグ

リッドサービスを受けられるようにする仕組みを検討しています。

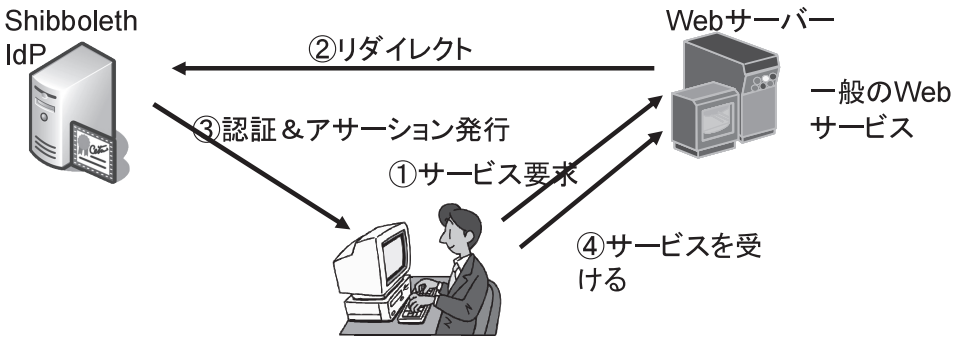

図３　Shibboleth の動作

Ⅵ．まとめ

将来は IT が電気と同じように人間社会にとって必要不可欠なものになると考えられます。その時コンピュータグリッドは，正にその言葉の由来となったパワーグリッド（電力網）と同様に，人間にとってのライフラインとして重要な役割を果たすことになるでしょう。

今後ともより安全なグリッドを目指して研究開発を続けて行かなくてはなりません。“グリッドで KISS!”は“Keep It Simple and Secure.”と読み替えてもよいかもしれません。

参考文献

［1］　“The Physiology of the Grid”, Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke1 (2002)

［2］　The Globus Toolkit 4 Programmer’s Tutorial, http://gdp.globus.org/gt4-tutorial/multiplehtml/index.html

［3］　“The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, I. Foster, C. Kesselman, and S. Tuecke, International Journal of High Performance

（みねお　しんいち：理化学研究所　次世代スーパーコンピュータ開発実施本部　開発グループ
開発研究員，国立情報学研究所　リサーチグリッド研究開発センター　客員教授）