社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－ＱＡＩ－21（２）
2006／10／23

グリッドコンピューティングにおけるセキュリティモデル －ＮＡＲＥＧＩにおける実装例の報告－

峯尾 真一† 四津 匡康‡ 佐賀 一繁§ 佐伯 裕治§ 田中 良夫ξ

†理化学研究所 〒100-0005 東京都千代田区丸の内 2-1-1

‡日本電気株式会社 〒108-8001 東京都港区芝 5-7-1

§国立情報学研究所 〒101-0051 東京都千代田区神田神保町 1-105

ξ産業技術総合研究所 〒101-0021 東京都千代田区外神田 1-18-13

E-mail:†mineo@riken.jp, ‡t-yotsu@bu.jp.nec.com, §{saga, ysaeki}@grid.nii.ac.jp, ξyoshio.tanaka@aist.go.jp

あらまし グリッドセキュリティ基盤（GSI）においては，X.509 ユーザ証明書とプロキシ証明書を用いた権限委譲によるセキュリティシステムが採用されている．NAREGI の開発しているグリッドコンピューティングミドルウェアにおけるセキュリティモデルは GSI を基礎としているが，スーパースケジューラがユーザの代行をしてジョブを実行管理することや，UMS(User Management Server)と呼ばれるユーザの鍵預託機能を持つという特徴を有している．

本論文では，このスーパースケジューラを実現するために開発した権限委譲方式，およびユーザの鍵管理と属性管理を一元的に行うために開発した UMS の仕組みに焦点を当てて NAREGI のセキュリティモデルについて説明する．このシステムを実現するために、二つの独立した MyProxy を用いている．

キーワード グリッドミドルウェア，セキュリティ，プロキシ証明書，ジョブスケジューラ，MyProxy

A Security Model in Grid Computing －An Implementation in the NAREGE project－

Shinichi MINEO†, Tadayasu YOTSU‡, Kazushige SAGA, Yuji SAEKI§ and Toshio TANAKAξ

†RIKEN, 2-1-1, Marunouchi, Chiyoda-ku, Tokyo, 100-0005, Japan

‡NEC Corporation,5-7-1,Shiba,Minato-ku,Tokyo,108-8001,Japan

§National Institute for Informatics, 1-105, Kandajinbocho, Chiyoda-ku, Tokyo, 101-0051, Japan

ξAIST, 1-18-13, Sotokanda, Chiyoda-ku, Tokyo, 101-0021, Japan

E-mail: †mineo@riken.jp, ‡t-yotsu@bu.jp.nec.com§{saga, ysaeki}@grid.nii.ac.jp, ξyoshio.tanaka@aist.go.jp

Abstract X.509 user credentials and proxy certificates are used in Grid Security Infrastructure (GSI). And end-to-end security system is based on the delegation mechanism of proxy certificates. The NAREGI project is developing a Grid Middleware based on GSI, and features that the Super Scheduler works on behalf of users to manage jobs, and that there exists key escrow system named UMS (User Management Server).

In this paper, we present the NAREGI security model focused on the delegation mechanism for the Super Scheduler and the integration of user keys and attributes management by UMS. Two independent MyProxy are used to realize this system.

Keyword Grid Middleware, Security, Proxy Certificate, Job Scheduler, MyProxy

1. はじめに

グリッドコンピューティングを実現するために，これまでに複数のセキュリティモデルが提案されて来ている．Globus におけるセキュリティ基盤である GSI (Grid Security Infrastructure)においては，X.509 ユーザ証明書とプロキシ証明書を用いたユーザの権限委譲が行われる．一方UNICORE においては，NJS と呼ばれるジョブスケジューラがユーザから署名されたジョブ記述を受け取り，ユーザの代わりにジョブの投入を行う方式が採用されている．

NAREGI プロジェクトにおいては，グリッドの標準

仕様策定団体であるOGF（Open Grid Forum）のOGSA-EMS（Open Grid Services Architecture – Execution Management Services）アーキテクチャを基にスーパースケジューラ(SS)と呼ばれるグリッドコンピューティングサービスを開発した．SS はユーザからジョブの集合であるワークフロー記述を受け取り，ユーザに代わってジョブの投入を行う．SSはUNICOREのジョブ署名機構をモデルとしたジョブ投入の仕組みを持つが，ジョブ投入にはGlobus のプロトコルであるGSI を用いるため，従来UNICOREで用いられていたジョブ署名だけではなく，別途ユーザのプロキシ証明書を受け渡す仕組みが必要となる．

そこで NAREGI においては，Globus プロジェクトで開発されたMyProxyと呼ばれるクレデンシャルの預託機能を用いて，２種類のセキュリティプロトコルを結合し，上記のジョブ投入を可能とする仕掛けを開発した．本論文では、通常の MyProxy と区別するためにこれを MyProxy2 と呼ぶ．（図 1-1 参照）

また取り扱いの難しい秘密鍵の保管・操作を利用者自身に行わせることについては従来から問題が指摘されていた．さらに仮想組織管理のための属性証明書の操作を追加することになり，問題の抜本的な解決が求められた．

そこで利用者の秘密鍵を預託し，ユーザプロキシ証明書や属性証明書を一括して作成するためのユーザ管理サーバ(UMS: User Management Server)を開発することにした．

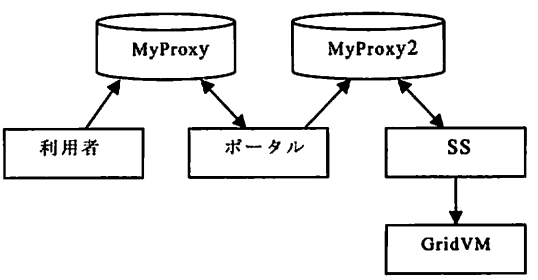

図 1-1　プロキシ証明書の伝播

本論文では，まずSSへの権限委譲について，２で解決すべき課題を定義し，３でMyProxy2を用いた解決方法を示し，４．でグリッドジョブの投入を行うユースケースを，５．でセキュリティモデルとしての考察を記述する．また次にUMSについて，６．でユーザ証明書管理上の課題を，７．でUMSの仕組みを，８．でセキュリティの考察を行う．最後に９．にまとめを記述する．

2. SS 実現のために解決すべき課題

SSはユーザからワークフローの実行を委託され，ユーザに替わってネットワーク上のコンピュータ資源を予約してGlobus ジョブを起動する．そのためには，

１）ユーザのジョブを受け取る．

２）Globus クライアントとしてユーザのプロキシ証明書を入手する．

３）Globus のジョブ投入手順に従い相手先にユーザプロセスの生成を行ってから権限委譲を行う．

という手順が必要となる．プロキシ証明書はその定義により，基になるプロキシ証明書から権限委譲して作成する必要があるが，以下の問題があるためSS においては通常の権限委譲手順を使うことができず，他の方法でプロキシ証明書の作成機能を実現する必要がある．

・ 権限委譲のためにはユーザ認証が必要となるがSS にはユーザの認証機能が無くまた責任分担と負荷の観点から持たせることは好ましくない．

・ 通常の権限委譲では相手先にユーザプロセスを起動させる必要があるが，セキュリティ上SSの稼動するサーバにユーザプロセスを起動することは避けたい．

・ ユーザが指定したワークフローをSSが自動的に実行し，必要ならプロキシ証明書の期間延長も自動的に行いたい．

以上の課題を解決するためにNAREGIではMyProxy2を用いて以下の仕組みを開発した．

3. MyProxy2 を用いたプロキシ証明書の受け渡し

NAREGI におけるソフトウェアアーキテクチャは以下のエンティティのアクションとして定義される．（図3-1参照）

・ ユーザ　：ワークフローの定義を行う．

・ スーパースケジューラ（SS）　：資源の予約とジョブのスケジュール，ワークフローの実行管理を行う．

・ GridVM　：資源の予約，管理とサンドボックス化，ローカルスケジューラとのインターフェース提供を行う．またユーザの認証と認可を行い，ジョブをGRAM経由で受け付ける．

・ 情報サービス(IS)　：ユーザや資源の管理に関する情報を収集し提供する．

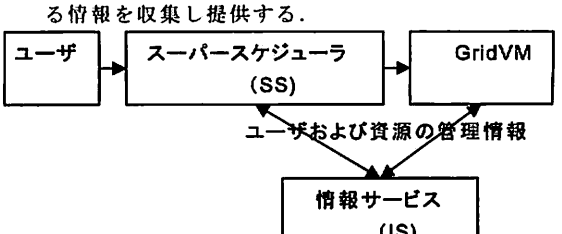

図3-1 NAREGI ソフトウェアアーキテクチャ

そこでSSに必要な機能要件は次の通りとなる．

・ SS はワークフローの真正性のみを検証しユーザの認証を行わない．

・ユーザのプロキシ証明書を安全な方法で受け取ることを可能とする．

・ワークフローの実行に必要なプロキシ証明書を必要に応じて期間延長可能とする．

上記を満たすプロトコルとして，MyProxy2 を用いてユーザのプロキシ証明書を預託し，指定したワークフローのみに限定してSS への払い出し（実際には預託・払い出しともGSI プロトコルによる新規作成）を行う方式を考案した．このMyProxy2はSS と対になって運用されプロキシ証明書の払い出しはSS のみに限定して行う．

MyProxy2を用いたプロキシ証明書の受け渡し手順は以下の通りである．

①ユーザはMyProxy2へプロキシ証明書を預託する．この時のパスフレーズはワークフロー記述のハッシュ値とする．

②ユーザはSS へワークフロー記述を送付する．

③SS はユーザから受け取ったワークフロー記述のハッシュ値を計算し，その値をパスフレーズとしてMyProxy2にユーザプロキシ証明書を要求する．

④MyProxy2はGSI プロトコルによりプロキシ証明書の新規作成を行う．

⑤SSはMyProxy2上のプロキシ証明書を削除する．

上記の手順により，ユーザはSS を利用して，指定したワークフロー記述を実行させることができる．またプロキシ証明書も安全な方法で受け渡すことができる．（図3-2参照）

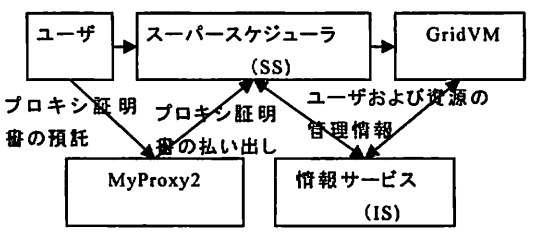

図3-2　MyProxy2によるプロキシ証明書の受け渡し

4. ユースケース

最も基本的なユースケースである「グリッドジョブの投入」においてMyProxy2の利用方法を示す．

1）　Client にて

①まずユーザはglobus ジョブの集合であるワークフローを作成し

Job-WF: Workflow Description

とする．

②次にjob-WF のハッシュ値を

Job-Hash = hash (Job-WF)

と計算する．

③MyProxy2へ登録するパスフレーズを

Pass Phrase = Job-Hash

とする．

④MyProxy2へ登録するためのユーザIDとして一意のワークフロー識別名を作成する．

user-id = 一意のワークフロー識別名

⑤MyProxy2へプロキシ証明書を登録する．

myproxy-init (user-id, Pass Phrase)

⑥ワークフロー記述をSS へ渡す．

Client ->SS :Job-WF

2) SS にて

⑦SS はワークフロー記述を受け取りuser-id（ワークフロー識別名）を取り出す．

⑧ワークフロー記述のハッシュを計算し，パスフレーズを取り出す．

Pass Phrase = hash (Job-WF)

⑨MyProxy2へプロキシ証明書を要求する．

myproxy-get-delegation (user-id, Pass phrase)

使用済みのプロキシ証明書は削除する．

⑩上記により生成されたユーザのプロキシ証明書からワークフローに従い，資源の予約を行ってからGSI 手順によりglobus ジョブをGridVM へ送付する．

⑪ワークフローの実行時にプロキシ証明書の有効期間を監視し，失効が近づいた場合にはSSの持つプロキシ証明書を元にジョブ投入先のプロキシ証明書の期間延長を行う．この延長期間は、MyProxy2に一時的に格納するプロキシ証明書の有効期間により制御することができる．

3)GridVM にて

⑫ GRAMで権限委譲を行う．この時，プロキシ証明書によるユーザの認証およびGridMapFile によるユーザの認可とローカルユーザID へのマッピングが行われる．

⑬Globus ジョブはGridVM により認可ポリシーに従ってローカルスケジューラ経由で計算機へ投入される．

5. SS におけるセキュリティ上の考察

NAREGI のセキュリティモデルにおける信頼の連鎖を示すと図5-1となる．PKI を用いたGSI が前提となっており，信頼の起点は認証局となる．そこでここではNAREGIの特徴であるSS とMyProxyに対してセキュリティ上の脅威に関する考察を行う．

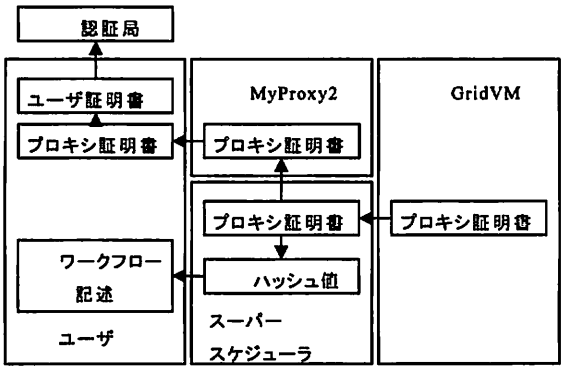

図 5-1 NAREGI における信頼関係の連鎖

5．1 SS に対するセキュリティ上の脅威とその対策

(1) 成りすまし

SSには認証機能が無く、成りすましの対策としてのユーザ認証はMyProxy2に委ね，機能分散している．MyProxy2に預託するプロキシ証明書のパスフレーズをジョブのハッシュ値とすることにより、SSに渡されたジョブが確かに認証されたユーザのジョブであることが確かめられる．

(2) 改ざん

ユーザが記述したジョブが改ざんされた場合には，ジョブのハッシュ値の検証により検出が可能である．

(3) 盗聴

ユーザ－SS 間はSSL／TLS をベースとした盗聴防止を行う．

(4) リプレイ攻撃

MyProxyに預託されたプロキシ証明書の払い出しにはパスフレーズとしてジョブのハッシュ値を用いることからジョブ記述を用いたリプレイ攻撃が考えられる．これを回避するためにSSはユーザプロキシ証明書の入手後に，MyProxy2上の元になった証明書を削除する．

(6) DOS 攻撃

SS はSS クライアント以外からのジョブ投入を受け付けない仕様となっており，DOS 攻撃を回避している．

5．2 MyProxy2に対するセキュリティ上の脅威とその対策

(1) 成りすまし

預託側の手順においてMyProxy2はプロキシ証明書の預託を行う際にユーザまたはホスト認証を行うことにより預託者の成りすましを防止する．また払い出し側の手順においては，払い出しを要求できるのは固定したSS だけと限定することにより成りすましを防止する．

(2) 改ざん

預託されるプロキシ証明書は通常のGSI による権限委譲 の手順により改ざんを防止することができる．

(3) 盗聴

ユーザ－MyProxy2間はSSL／TLS をベースとした盗聴防止を行う．

(4) DOS 攻撃

MyProxy2はユーザまたはホストの認証により不正なサービス要求を拒否することができる．

6. ユーザ証明書管理上の課題

NAREGI ミドルウェアにおいては，GSI のためのプロキシ証明書管理のために MyProxy を，VO 管理のために EGEE が開発した VOMS を用いている．そこで一般的な証明書の取り扱い手順は以下の通りとなる．

①利用者は認証局から自身を証明する X.509 公開鍵証明書を入手する．

②proxy-init で自分のプロキシ証明書を作成する．

③上記で作成したプロキシ証明書を元に voms-proxy-init で VOMS サーバに属性証明書発行を依頼する

④②で作成したプロキシ証明書の拡張部分に③で作成した属性証明書を埋め込む．

⑤上記のプロキシ証明書を myproxy-init で MyProxy にデリゲーションしておく．

⑥ポータルにログイン後，myproxy-get-delegation により MyProxy からプロキシ証明書を作成する．

⑦できたプロキシ証明書を用いてグリッドジョブを起動する．(NAREGI の場合は，SS にワークフローの実行を依頼する．)

以上の手順から次の課題が提起された．

- 一般の利用者にとって証明書の取り扱いが非常に煩雑になり，かつ理解も難しいので実際に操作ができない可能性が高い．
- 一般の利用者に自分の証明書を管理させることは難しく，むしろ危険である．
- 一般の利用者は GT の様な特別の環境が無くてもブラウザのみでグリッドコンピューティングを行える様にしたい．
- MyProxy に格納するプロキシ証明書のファイル名の一意性を確保したい．

これらの課題を解決するために NAREGI では，UMS(user Management Server)と呼ぶ証明書一括管理の仕組みを開発した．

7. UMS の仕組み

UMS は一般の利用者の証明書を安全に管理するためのサーバである．利用者の証明書および秘密鍵の管理を行うという役割上，他のコンポーネントとは独立して利用する．以下に UMS の機能を示す．

(1) X.509公開鍵証明書の取得

UMS 上で鍵生成を行い，認証局から公開鍵証明書の取得を行う．取得した公開鍵証明書は UMS 上の利用者ホームに格納される．

(2) 属性証明書の取得

VOMS から利用者が参加している VO の属性証明書を取得し，プロキシ証明書の拡張部分に埋め込む．

(3) MyProxy へのプロキシ証明書の預け入れ

(2)で作成したプロキシ証明書を MyProxy に預け入れる．MyProxy から Proxy 証明書を取り出すときに必要となるパスフレーズは利用者が設定する．

(4) VO 名の表示

利用者が参加している VO 名の一覧を表示する．

(5) パスワードの変更

UMS 上の利用者アカウントのパスワードを変更する．

UMS では上記機能をメニュー形式で提供することにより，一般の利用者は特別なコマンドを覚えることなく証明書を取り扱うことができる．さらに一般の利用者はメニュー以外の操作を行うことができないため，成りすましや改ざんといった不正な操作を防止することができる．また一般の利用者はポータルを経由して UMS 上の機能を操作することもできるため，直接 UMS にアクセスすることなく，自らの証明書を操作することが可能である．（図 7-1 参照）

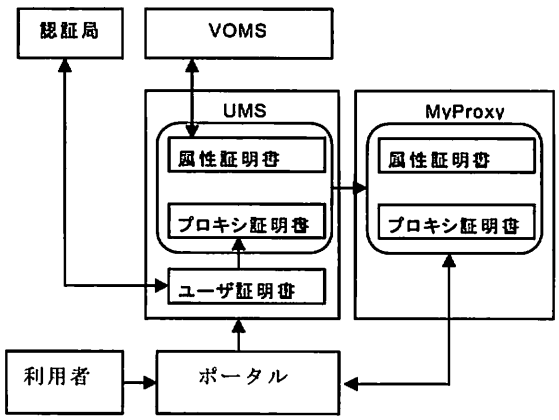

図 7-1 UMS における証明書の処理

MyProxy を利用したグリッドコンピューティングの運用を行うとき，MyProxy に格納されるプロキシ証明書のファイル名は利用者が自由に設定することができるので，ファイル名が重複する可能性がある．仮にファイル名が重複した場合，利用者は MyProxy へプロキシ証明書を格納することができなくなるため，MyProxy に格納されるプロキシ証明書のファイル名は一意性を確保されなければならない．そこで NAREGI では UMS で利用者のアカウント管理を行うことにより，UMS 上の一意的なアカウント名を MyProxy に格納されるプロキシ証明書のファイル名とすることにより，一意性を確保している．

8. UMS におけるセキュリティ上の考察

8．1　X.509 公開鍵証明書の保護

利用者は UMS 上で鍵生成を行い，X.509 公開鍵証明書を取得する．このとき秘密鍵は利用者が設定したパスフレーズで暗号化される．UMS 上には利用者のアカウントがあり，取得した証明書は UMS 上の利用者ホームディレクトリに格納し，ファイルシステムのセキュリティで保護している．

8．2　プロキシ証明書の期間延長

長時間のジョブを実行するためには，長期間有効なプロキシ証明書が必要となる．しかし，長期間有効なプロキシ証明書が計算ノードに存在するとプロキシ証明書が悪用されるリスクが高まるため，プロキシ証明書の期間は短い方がセキュリティ上好ましい[8]．そこで短期間のプロキシ証明書を用いて長期間のジョブを実行する手段として MyProxy を利用したプロキシ証明書の期間延長方法がある．通常行われるプロキシ証明書の期間延長は，利用者が MyProxy に長期間のプロキシ証明書を預け入れし，プロキシ証明書の有効期限が切れる前にポータルにある RenewalService が MyProxy からプロキシ証明書の再取得を行い，計算ノード上に存在するプロキシ証明書の更新を行う仕組みである[9]．しかし NAREGI ではポータルからジョブの投入を行うのではなくＳＳからジョブの投入を行う，MyProxy が 2 段に入る，などの理由により通常の期間延長を行うことができない．このため NAREGI では，利用者が SS 上にジョブ実行に適切な期間の Proxy 証明書の作成を行うことにより，SS から期間延長を行わせる仕組みを採る．

9. まとめ

MyProxy2を用いることにより，セキュリティのレベルを落とすことなくNAREGI で用いられる２種類のセキュリティプロトコルの変換を行うことができる．この場合MyProxy2にあるプロキシ証明書はいわばSSにとってのジョブ実行のチケットとして働く．

但しポータルとSSの間でGSIのチェインをさらに別のMyProxyで繋いでいるため，通常行われている様なユーザ側のMyProxy を用いたプロキシ証明書の期間延長ができない．また現在はSS が世界に唯一の存在として全てのジョブをスケジューリングすることになっているが，将来の拡張性を考えればSS を分散する

ことを目指さざるを得ず，その場合にはこの方法を再検討する必要がある．

将来的には，上記のセキュリティモデルをGT4で開発された権限委譲サービスを用いて再設計することも検討したい．

またUMSを用いることにより複雑な証明書の管理を利用者に行わせることなく証明書を操作できるが，既に取得している証明書を利用してグリッドコンピューティングを利用したい場合には，安全に証明書をUMSへ登録することができる仕組みがない．グリッドコンピューティングをより広く普及させるためにUMSを用いない利用方法についても検討したい．

これらの問題を解決することが今後の課題となる．

謝辞

NAREGIセキュリティモデルの研究開発に関し，日頃から多くのご助言を頂いているNAREGIプロジェクトリーダである国立情報学研究所の三浦謙一教授，サブリーダである大阪大学の下條真司教授，東京工業大学の松岡聡教授，特にセキュリティモデルに関して討論に参加して頂いている日本電気株式会社の森拓也氏，天羽宏嘉氏に深く感謝します．

尚，本研究の一部は，文部科学省「経済活性化のための重点技術開発プロジェクト」の一環として実施している，超高速コンピュータ網形成プロジェクト（National Research Grid Initiative）の成果である．

文　献

[1] Ian Foster, Carl Kesselman, Gene Tsudik, Steven Tuecke, "A Security Architecture for Computational Grids", presented at Proceedings of the 5th ACM Conference on Computer and Communications Security 1998.

[2] Von Welch, Frank Siebenlist, Ian Foster, et al., "Security for Grid Services", Twelfth. International Symposium on High Performance Distributed Computing (HPDC-12), Proceedings, p.48-57, June 2003.

[3] S. Tuecke, V.Welch, D. Engert, L. Pearlman, M. Thompson, "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile", Request for Comments:3820, June 2004.

[4] David F. Snelling, et al. "Explicit Trust Delegation: Security for Dynamic Grids", FUJITSU Sci. Tech. J., 40,2,p.282-294, December 2004.

[5] T. Goss-Walter, et al., "An Analysis of the UNICORE Security Model", GFD.18, http://www.ggf.org/documents/GFD.18.pdf, July 2003.

[6] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich, "The Open Grid Services Architecture, Version 1.0", GFD.30, http://www.ggf.org/documents/GFD.30.pdf,2005

[7] J. Basney, "MyProxy Protocol", GWD-E.054, http://www.ggf.org/documents/GFD.54.pdf, November 26, 2005

[8] Von Welch,Ian Foster,Carl Kesselman,Olle Mulmo,et al. X.509 Proxy Certificate for Dynamic Delegation, 3rd Annual PKI R&D Workshop, 2004.

[9] J.Basney, A Credential Renewal Service for Long-Running Jobs, 6th IEEE/ACM International Workshop on Grid Computing (Grid 2005), Seattle, WA, November 13-14, 2005.