

Grid Security Infrastructure – ein Überblick

Bartol Filipović, Tobias Straub

Fraunhofer-Institut für Sichere Informationstechnologie
Rheinstr. 75
64295 Darmstadt
filipovic@sit.fraunhofer.de
straub@sit.fraunhofer.de

Abstract: Beim Grid Computing gilt es vielfältige Sicherheitsaspekte zu beachten. Ein Teil davon betrifft die Grid Security Infrastructure (GSI), die im Umfeld des Globus Toolkits entstand und deren Konzepte sich mittlerweile auch in weiteren Grid-Middlewares oder anderen Anwendungen wiederfinden. Dieser Beitrag gibt eine Übersicht über die relevanten GSI-Komponenten und stellt ihre Eigenschaften und Funktionen dar.

1 Einleitung

Grid Computing bedeutet, vereinfacht gesagt, die Nutzung verteilter Rechenleistung und Datenspeicherkapazität über das Internet. Dabei geht es aber um weit mehr als nur die Kommunikation zwischen Computern. Ziel ist es vielmehr, aus dem globalen Computernetzwerk eine einheitliche Ressource mit gewaltiger Performanz zu machen, die Nutzern bei Bedarf zur Verfügung stehen kann.

Charakteristisch für Grid Computing ist, dass oftmals heterogene und geografisch verteilte Komponenten integriert oder Teile des Gesamtsystems autonom verwaltet werden – auch über Organisationsgrenzen hinweg. Um die unterschiedlichen Systemressourcen und Komponenten in einem Grid nutzen zu können, kommen verschiedene Middleware-Lösungen zum Einsatz, wie etwa das Globus Toolkit (GT)¹, UNICORE² oder gLite³.

Grid Computing ist als relativ junge Technologie noch in ständiger Entwicklung begriffen. Die verschiedenen Middlewares haben teils identische, teils konkurrierende Eigenschaften. Es gibt Bestrebungen, durch die Schaffung technischer Standards die Entwicklung und Integration zu vereinfachen, etwa auf Grundlage von Web Services und serviceorientierter Architekturen. Die Technologie wird bereits produktiv in einigen Projekten eingesetzt. Rechenintensive Computersimulationen, bei denen auch große Datenmengen verarbeitet werden, etwa in der Astro-, der Hochenergiephysik oder der

¹ <http://www.globus.org/toolkit/>

² <http://www.unicore.org/forum.htm>

³ <http://glite.web.cern.ch/glite/>

Klimaforschung, sind typische Grid-Projekte. Gerade vor dem Hintergrund der zunehmenden praktischen Relevanz der Technologie haben Sicherheitsfragen im Grid eine besondere Bedeutung.

Sicherheit im Grid betrifft durchgängig alle Schichten der Grid-Architektur von den Netzwerkverbindungen und -Protokollen über die Hardwarekomponenten bis zu der Middleware- und Anwendungsebene. Zu einzelnen Teilaspekten gibt es bestimmte – auch unabhängig von Grid Computing einsetzbare – Sicherheitslösungen, etwa Firewalls zur Regulierung und Kontrolle der Netzwerk-Kommunikation oder eine Ressourcen- und Rechtebeschränkung durch das Betriebssystem.

Dieser Beitrag gibt eine Übersicht über die Eigenschaften und Funktionen von GSI (Grid Security Infrastructure). GSI wird als Sicherheitsplattform in den gängigen Grid-Middlewares – etwa Globus Toolkit und gLite – sowie auch unabhängig davon in Anwendungen wie SAMD (Seamless Access to Multiple Datasets⁴) oder SRB (Storage Resource Broker⁵) eingesetzt. Zunächst werden die Sicherheitsanforderungen im Grid präzisiert, bevor in Abschnitt 3 die einzelnen Komponenten des GSI im Mittelpunkt stehen. Anhand des Globus Toolkit erklären wir die Umsetzung der Konzepte in einer weit verbreiteten Grid Middleware.

2 Sicherheit im Grid

2.1 Herausforderung Grid Security

In einem typischen Einsatzszenario für Grid Computing sind *Ressourcen und Benutzer* nicht nur geografisch, sondern auch über Organisationsgrenzen hinweg *verteilt* und in ihrer Zusammensetzung zudem sehr *dynamisch und laufenden Änderungen* unterworfen. In *heterogenen Systemumgebungen ohne einheitliche Security Policy* werden *autonom verwaltete Teilsysteme* integriert, die *unterschiedliche Typen von Credentials*, also Identitäts- und Berechtigungsnachweise, zur Authentifizierung von Nutzern und Diensten verlangen.

Eine weitere Grid-Besonderheit ist die Anforderung, dass Prozesse, wenn sie einmal gestartet wurden, in der Lage sein müssen, sich Dritten gegenüber im Namen des Benutzers zu authentifizieren, um ihre Aufgabe zu erledigen. Dazu gehören auch das autonome, d. h. ohne Benutzerinteraktion erfolgende, Starten weiterer Prozesse und das Vererben von Rechten an diese.

⁴ <http://www.sve.man.ac.uk/Research/AtoZ/SAMD/>

⁵ <http://www.sdsc.edu/srb/>

2.2 Sicherheitsziele

Aus der Vernetzung von Ressourcen ergibt sich die Anforderung nach einer starken *Authentifizierung* der Kommunikationspartner, die insbesondere auch Domänenübergreifend arbeitet. Weiter sollte die Verwaltung von Zugriffsrechten (*Autorisation*) auf Basis lokaler Policies möglich sein, damit Administratoren oder auch einzelnen Benutzer ihre Rechen- und Daten-Ressourcen in eigener Kontrolle anderen zugänglich machen können. Dazu kommen die zu unterstützenden klassischen Schutzziele der *Vertraulichkeit* und *Integrität* von Daten; die *Delegation* von Privilegien wurde bereits im vorigen Abschnitt angesprochen.

Die genannten Anforderungen werden durch GSI-Komponenten realisiert. Darüber hinaus bietet GSI aber keine Mechanismen für Aspekte wie etwa Ressourcen- und Rechteverwaltung oder die Überwachung des Netzwerkverkehrs. Solche Aufgaben werden typischerweise von lokalen Betriebssystemen und mit Hilfe von weiteren Werkzeugen wie etwa Firewalls gewährleistet. Neben bzw. aufbauend auf GSI kommen Mechanismen wie Virtual Organization Membership Service (VOMS) oder MyProxy in Grid-Middlewares zum Einsatz, um einzelne Teilaufgaben zu lösen.

3 GSI-Komponenten

GSI ist eine, im Rahmen von Globus Toolkit entwickelte, Sammlung von Sicherheitsprimitiven, Protokollen und APIs, die Sicherheitsmechanismen für Grids bieten. Diese Sammlung lässt sich grob in drei logische Teile gemäß ihrer Aufgaben gliedern:

1. Sichere, d. h. vertrauliche, authentische und integre, Kommunikation wird mit etablierten Verfahren der symmetrischen und asymmetrischen Kryptografie ermöglicht. Dies kann Kanal- oder Nachrichten-basiert geschehen.
2. Eine Public-Key-Infrastruktur (PKI) bestehend aus unabhängigen Zertifizierungsstellen (CAs) garantiert die Identität der Grid-Teilnehmer (Anwender, Systeme und Dienste).
3. Die temporäre Delegation von Credentials sowie ein benutzerfreundliches Single Sign-On (SSO) an Grid-Diensten wird mit Hilfe so genannter Proxy-Zertifikate und darauf aufbauender Werkzeuge realisiert.

Für den Anwendungskontext Grid wurden im Wesentlichen bewährte Techniken kombiniert und – wo erforderlich – erweitert. Die dazu ursprünglich eingesetzten Techniken PKI (siehe 3.1), *X.509 Version 3-Zertifikate* [ITU97], *Transport Layer Security* (TLS) [DiAl99], *Generic Security Services API* (GSS-API) [Linn00] sind bekannte und etablierte Standards, während mit gewissen Erweiterungen der GSS-API [MWTE04], *Proxy-Zertifikaten* gemäß RFC 3820 [TWE⁺04] und – darauf aufbauend – *Single Sign-On* und *Delegation* [JTE01, ABM04] neue, Grid-spezifische Mechanismen geschaffen worden sind.

GSI ist historisch ein Produkt der Globus Toolkit-Entwicklung. Darüber hinaus fanden die GSI-Bestandteile und -Konzepte auch, wie bereits erwähnt, Eingang in andere Middlewares. Mit dem Aufkommen von Web Services (WS) und serviceorientierten Architekturen gab es auch bei GSI entsprechende Anpassungen. Aktuell bietet etwa Globus Toolkit 4 sowohl die klassischen GSI-Möglichkeiten (als „pre-WS“ bezeichnet) als auch WS-basierte Erweiterungen. Im Folgenden wird zunächst pre-WS GSI besprochen, da dort schon die wesentlichen Konzepte sichtbar sind. Anschließend werden die WS-Erweiterungen in GT 4 vorgestellt.

3.1 Grid-PKI

Für das Grid wurde, koordiniert von der *International Grid Trust Federation* (IGTF), eine PKI aufgebaut. Im Gegensatz zu PKIs in anderen Anwendungsgebieten, etwa für sichere Email, hat sich im Grid-Kontext tatsächlich eine einheitliche globale PKI mit der Grid Policy Management Authority (GridPMA⁶) als oberster Instanz etablieren können. Im Sinne einer Arbeits- und Zuständigkeitsteilung stellen nationale Zertifizierungsstellen jeweils für ihre Teilnehmer (Personen oder Maschinen) Identitätszertifikate aus, mit denen sie die Bindung eines öffentlichen Schlüssels und eines Teilnehmers dokumentieren. Dabei handelt es sich in der Sprechweise von X.509 um *End Entity*-Zertifikate, mit denen die Zertifikatsnehmer also nicht selbst als CA auftreten und damit keine weitere Teilnehmer zertifizieren können. In Abschnitt 3.2 wird beschrieben, wie sich trotz dieser Einschränkung im Grid weitere Zertifikate ausstellen lassen. Daneben werden Zertifikate meist zur Authentifizierung oder teilweise auch zum Verschlüsseln von Nachrichten verwendet.

In jedem Mitgliedsland der IGTF gibt es unterhalb der GridPMA jeweils nur eine nationale Root-CA, die Grid-Zertifikate ausstellt. Somit ist gewährleistet, dass alle Zertifikatsnehmer weltweit eindeutig identifiziert werden können, da durch die unterschiedliche Länderkennung im Distinguished Name (DN) des Zertifikatsinhabers disjunkte Namensräume gewährleistet sind. In Deutschland gibt es als Ausnahme von dieser Regel zwei nationale, gleichgestellte Root-CAs, die von DFN⁷ und dem Forschungszentrum Karlsruhe⁸ betrieben werden. Die ausgestellten Zertifikate sind anhand des DN-Bestandteils „Organisation“ (O=GridGermany bzw. O=GermanGrid) unterscheidbar.

3.2 Proxy-Zertifikate

Proxy-Zertifikate sind von ihrer Syntax, jedoch nicht von ihrer Semantik, her X.509v3-Zertifikate. Zur Unterscheidung werden daher gelegentlich die Begriffe Proxy- bzw. Identitätszertifikat verwendet. Im engen Sinne von X.509 werden nur Zertifikatsketten bei der Validierung als gültig erachtet, die von der obersten Zertifizierungsstelle über möglicherweise eine oder mehrere Zwischen-Zertifizierungsstellen bis zum (Identitäts-)

⁶ <http://www.gridpma.org/>

⁷ <http://www.dfn.de/content/dienstleistungen/dfnpki/grid/>

⁸ <http://www.gridka.de>

Zertifikat einer End Entity führen. Dagegen können mit einem Schlüsselpaar, dessen Public Key in einem Identitätszertifikat enthalten ist, nun im RFC 3820-Gültigkeitsmodell speziell als solche gekennzeichnete Proxy-Zertifikate unterschrieben werden. Mit Proxy-Zertifikaten selbst lassen sich ausschließlich weitere Proxy-, jedoch keine Identitätszertifikate ausstellen.

Zur Kennzeichnung sind Proxy-Zertifikate mit der X.509-Extension ProxyCertInfo⁹ versehen, die zudem als kritisch markiert ist. Letzteres bedeutet, dass Anwendungen, die nicht mit Proxy-Zertifikaten umgehen können, diese generell als ungültig ablehnen.

Prozesse verwenden für die zertifikatsbasierte Authentifizierung gegenüber Grid-Diensten ausschließlich Proxy-Zertifikate. Das zum Zertifikat gehörende Schlüsselpaar wird dem Prozess dabei unverschlüsselt in Form einer PKCS#12-Datei mitgegeben, was bedeutet, dass dieses Softtoken auf der ausführenden Plattform lediglich über die Zugriffsrechte des Dateisystems geschützt ist. Dies ist insbesondere deshalb kritisch, da sich ein Unberechtigter, der Zugriff auf diese Datei erlangt, während der Gültigkeitsdauer des zugehörigen Proxy-Zertifikats für den rechtmäßigen Anwender ausgeben kann.

Im Gegensatz zu Zertifikaten, die von einer CA ausgestellt werden, gibt es aber keine Möglichkeit, Proxy-Zertifikate direkt mit Hilfe von X.509-Sperrlisten zurückzurufen¹⁰. Auf die Gültigkeit von Proxy-Zertifikaten lässt sich aber auch jetzt schon Einfluss nehmen, indem das übergeordnete End Entity-Zertifikat gesperrt wird. Allerdings ist dies für den Benutzer mit weit reichenden Konsequenzen verbunden, da sich sämtliche in seinem Namen laufenden Prozesse nicht mehr authentifizieren können und er sich außerdem ein neues Identitätszertifikat ausstellen lassen muss (was ggf. eine aufwändige Neuentifizierung durch die CA erfordert).

Eine Alternative zur Verwendung von Sperrlisten für Proxy-Zertifikate ist es, sie mit einer relativ kurzen Gültigkeit zu verwenden und diese nach Bedarf zu verlängern. Somit behält der Anwender die Zertifikate der unter seinem Namen laufenden Prozesse zeitnah zu kontrollieren. Allerdings würde die manuelle Verlängerung der Zertifikate einen erheblichen Aufwand bedeuten und wäre bei zu kurzen Intervallen wenig praktikabel.

Das Werkzeug *MyProxy* [BHW05] erlaubt es nicht nur, eigene Identitätszertifikate zentral zu verwalten und im Sinne eines Single Sign-On Proxy-Zertifikate auf einfache Weise bereit zu stellen. MyProxy ist vielmehr auch in der Lage, Proxy-Zertifikate automatisch zu verlängern, sofern die Berechnung des entsprechenden Prozesses andauert und kein Sperrgrund vorliegt. Der entfernte Prozess authentifiziert sich dabei mit dem aktuellen Proxy-Zertifikat dem MyProxy-Server gegenüber und stellt bei ihm eine Zertifizierungsanfrage. Auf diese Weise werden Benutzbarkeit und Sicherheit gleichermaßen gewährleistet. Ein Beispiel für ein Job-Verwaltungssystem, das das so genannte Proxy-Renewal unterstützt, ist Condor-G [Con06].

⁹ OID: 1.3.6.1.5.5.7.1.14

¹⁰ Dies wird aber möglicherweise zukünftig ein Feature der GSI werden [WFK+04].

3.3 Security Policy-Mechanismen

GSI bietet zwei Mechanismen der Rechtevergabe, die im Folgenden näher beschrieben werden. Zum einen lassen sich bei der Delegation die Privilegien des Proxys festlegen, die er vom Delegierenden übernimmt. Zum anderen kann der Administrator desjenigen Systems, das den Prozess ausführt, Rechte auf Dateisystemebene festlegen. Am Ende dieses Abschnitts wird das Zusammenwirken der GSI-Bestandteile nochmals schematisch dargestellt.

Proxy-Zertifikate verfügen, wie bereits erwähnt, über die X.509 Extension ProxyCertInfo. In diesem Feld ist die Angabe der *ProxyPolicy* verpflichtend, wobei prinzipiell eine beliebige existierende Policy-Sprache zum Einsatz kommen kann, die jedoch von allen beteiligten Systemen auch verstanden werden muss. In diesem Fall, der als „Restricted Delegation“ bezeichnet wird, werden allerdings die Implementierungen aufwändiger. Relying Parties müssen nicht nur die Semantik der Delegations-Policy verstehen, sondern auch die entsprechenden Beschränkungen durchsetzen können.

Da diese Policies oft anwendungsspezifisch sind, ist es für die darunter liegende Security Library, die die Authentifizierung über Proxy-Zertifikate abwickelt, schwierig zu erkennen, welche Beschränkungen die Anwendung tatsächlich kennt und durchsetzen kann. Ohne diese Gewähr kann kein Proxy-Zertifikate zuverlässig akzeptiert werden. Dieses Problem wurde durch eine Erweiterung der GSS-API adressiert (siehe unten). Da der Ansatz aber aufgrund der Anwendungsabhängigkeit kompliziert ist, wird diese Technik in der Praxis nur wenig eingesetzt [WFK+04].

Im RFC 3820 sind jedoch bereits zwei spezielle „Sprachen“ definiert, die alle Systeme verarbeiten können müssen. Zum einen ist dies die „inheritAll“-Policy, bei der sämtliche Rechte vom Aussteller an den Proxy delegiert werden¹¹, zum anderen eine „independent“-Policy, bei der überhaupt keine Rechte delegiert werden. Die Independent-Policy hat dennoch ihre Berechtigung, da diese Proxy-Zertifikate alleine zwar nicht für Berechnungen im Grid, jedoch aber z. B. für GridFTP eingesetzt werden können [WSF+03]. Darüber hinaus lassen sich über ein Attribut-Zertifikat (nach RFC 3281 [FaHo02]) oder auf anderem Wege dem Proxy doch noch und unabhängig von dem Proxy-Zertifikat wieder Rechte zuweisen [WFK+04]. Ein Nachteil dieser Variante ist aber die fehlende Protokollunterstützung für Attribut-Zertifikate in den Anwendungen.

Prozesse, die über ein Proxy-Zertifikat verfügen, können ihrerseits wiederum Prozesse starten und diesen ein Proxy-Zertifikat ausstellen. Dieser Schritt lässt sich beliebig iterieren, sofern keine entsprechende Einschränkung vom Anwender selbst oder von einem Prozess festgelegt wird. Dazu dient die Option, in der Extension ProxyCertInfo auch eine Längenbegrenzung der Kette aufeinander folgender Proxy-Zertifikate anzugeben.

Ein von einem Anwender remote gestarteter Prozess wird auf dem Zielsystem stets unter der Kennung eines lokalen Benutzerkontos ausgeführt. Dieser Mechanismus gestattet es dem lokalen Administrator, vollständige Kontrolle über die Rechte von Grid-Diensten zu

¹¹ Dies wird auch als „Impersonation Mode“ bezeichnet [WFK+04].

behalten. Grid-Anwender und ihre Prozesse lassen sich über den DN im Zertifikat global eindeutig identifizieren. Für die Zuordnung der globalen Identität zu einer lokalen gibt es in den Middlewares dann verschiedene Mechanismen.

Im Globus Toolkit 2 erfolgt diese Zuordnung über ein so genanntes *Grid-Mapfile*. Dabei handelt es sich um eine Konfigurationsdatei der Middleware-Installation auf dem Zielsystem, die eine (partielle) Funktion von der Menge der Distinguished Names in die Menge der lokalen Benutzerkonten beschreibt. Auf diese Weise ist es möglich, selektiv globale Benutzer auf dem System zuzulassen und diesen individuelle oder auch gemeinschaftlich genutzte lokale Accounts zuzuweisen.

Mehr Flexibilität bieten die Ansätze *Local Center Authorization Service* und *Local Credential Mapping Service* (LCAS/LCMAPS¹²). Damit lassen sich etwa virtuelle Organisationen und Gruppenmitgliedschaften, White- und Blacklists oder Pool-Accounts realisieren. Für weitere Details siehe etwa [DEF⁺06].

Beispiel: Abbildung 1 zeigt ein Anwendungsbeispiel, in dem die Grundoperationen von GSI sichtbar sind. Ein Benutzer, der über ein End Entity-Zertifikat (EEZ) verfügt, erstellt damit ein User Proxy-Zertifikat (UPZ) mit kurzer Gültigkeitsdauer (schraffiert dargestellt). Beim Zugriff auf Ressourcen der Domäne 1 ist eine beidseitige Authentifizierung von Host und Zielrechner erforderlich. RZ₁ bezeichnet hier das Zertifikat der Ressource. Nach erfolgreicher Authentifizierung wird der globale Benutzer mit

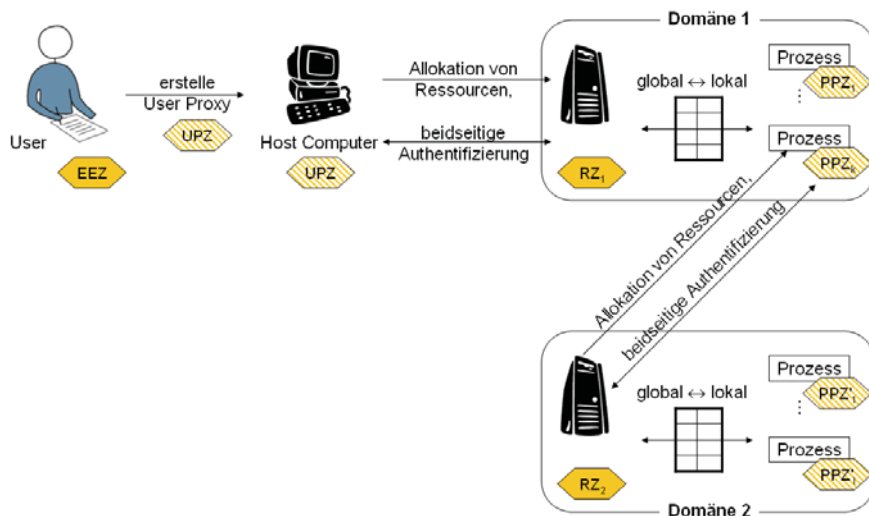


Abbildung 1: Beispiel für das Zusammenwirken der GSI-Komponenten

¹² <http://www.dutchgrid.nl/DataGrid/wp4/>

dem im EEZ angegebenen DN einem lokalen Benutzer zugeordnet. Im Beispiel geschieht dies über ein Grid-Mapfile (als Tabelle dargestellt).

Der globale Benutzer startet – ausgestattet mit den Rechten des entsprechenden lokalen Subjekts in Domäne 1 – Prozesse, die wiederum Proxy-Zertifikate erhalten (PPZ₁ bis PPZ_k), diesmal ausgestellt von UPZ. Mit diesen Zertifikaten kann sich ein Prozess auch gegenüber der Domäne 2 authentifizieren und dort ebenfalls Prozesse starten, die entsprechende Credentials erhalten (Proxy-Zertifikate PPZ'₁ bis PPZ'_l). Auch hier findet wiederum eine Abbildung des durch EEZ bestimmten globalen Subjekts auf ein lokales Subjekt der Domäne 2 statt. Wie oft sich dieser Schritt der Delegation wiederholen lässt, kann durch entsprechende Angabe der „Kettenlänge“ in UPZ beschränkt werden. Zwischen Proxy-Zertifikaten, die auf dasselbe EEZ zurückgehen, besteht im Grid implizites Vertrauen, d. h. dass die zugehörigen Prozesse (hier: UPZ, PPZ₁,..., PPZ_k, PPZ'₁,..., PPZ'_l) kooperieren und untereinander vertraulich und authentifiziert Daten austauschen können.

3.4 Erweiterungen der GSS-API

Das Generic Security Service Application Programming Interface (GSS-API, RFC 2743/2744) unterstützt die einfache Programmierung von Sicherheitsmechanismen für Netzwerk-Anwendungen. Es folgt hier nur eine kurze Einführung in die Konzepte (siehe etwa [Sun00] für eine ausführlichere Darstellung).

Die GSS-API ist insofern generisch, als dass sie die Implementierung unabhängig von einem bestimmten Sicherheitsmechanismus (z. B. in Bezug auf das Credential-Format), dem Transportprotokoll (möglich sind etwa RPC oder Sockets) oder auch einer bestimmten Plattform hält. Dies fördert in besonderem Maße die Portabilität. GSS-API implementiert jedoch selbst keine Sicherheitsdienste, sondern stellt nur ein standardisiertes Framework bereit, über das Anwendungen generisch Mechanismen wie etwa Kerberos oder PKI nutzen können. Die Auswahl der konkreten zugrunde liegenden Verfahren kann der Programmierer GSS-API überlassen und Vorgabewerte wählen oder selbst einzelne Verfahren auswählen (so genannte Quality of Protection – QOP). In letzterem Fall kann allerdings die Portabilität beeinträchtigt werden.

GSS-API erlaubt es einerseits Anwendungen, einen vertrauenswürdigen „Security Context“ herzustellen, in dem sie untereinander Daten austauschen können. Andererseits stellt GSS-API in jedem Fall Authentifizierungsverfahren und – sofern dies die zugrunde liegende Technologie unterstützt – Integritätsschutz und Verschlüsselung bereit.

Die GSI-Erweiterungen von GSS-API sind in [MWTE04, Eng04] beschrieben. Diese relativ geringfügigen Änderungen betreffen den

1. *Import und Export von Credentials* zwischen Prozessen (auch GSS-API-fremden Anwendungen),
2. Das Ermöglichen der *Delegation zu einem beliebigem Zeitpunkt* (GSS-API in der ursprünglichen Version erlaubt dies nur während des Aufbaus des Security

Context; diese Funktion kann nun dafür verwendet werden, Credentials zu aktualisieren),

3. eine Erweiterung des *Handlings von Credentials* um die Möglichkeit, Certificate Extensions oder Policy-Informationen zu verarbeiten,
4. die *Parameter-Übergabe an den Security Context*, z. B. um Verschlüsselung auszuschalten oder Beschränkungen für die Delegation festzulegen.

Anwendungen, die auf der GSS-API aufsetzen und GSI unterstützen, sind etwa SSH (OpenSSH¹³ und Putty¹⁴, beide unterstützen über die GSS-API auch Kerberos anstelle von GSI), FTP (GSIFTP¹⁵, GridFTP¹⁶) sowie CVS (GridCVS¹⁷).

3.5 Absicherung der Kommunikation zwischen Grid-Komponenten

Es werden nun pre-WS-Mechanismen von GSI zur Absicherung der Kommunikation zwischen Grid-Knoten beschreiben, die im Wesentlichen auf dem TLS-Protokoll basieren. TLS ist weit verbreitet, da es beliebige TCP-basierte Protokolle tunneln kann und eine Reihe von Sicherheitsfunktionen schon von Haus aus unterstützt. Während des Verbindungsaufbaus, dem so genannten TLS Handshake, können die Kommunikationsendpunkte flexibel die verwendeten Kryptoverfahren und -parameter und somit das Sicherheitsniveau aushandeln. Auch ist die Möglichkeit einer starken Authentifizierung mittels eines kryptografischen Challenge-Response-Verfahrens vorgesehen. Diese Form der Authentifizierung ist wesentlich stärker als etwa ein Passwort-basierter Mechanismus. Oft authentifiziert sich in der Praxis bei HTTPS-Verbindungen nur der Server mit einem kryptografischen Verfahren, während der Client (wenn überhaupt) ein Passwort über die verschlüsselte Leitung sendet. Im Gegensatz dazu authentifizieren sich im Grid in der Regel beide Kommunikationspartner mit ihrem Zertifikat. An dieser Stelle wird vom TLS-Standard insofern abgewichen, als dass nicht nur Identitäts-, sondern auch Proxy-Zertifikate für die Authentifizierung zugelassen werden.

4 GSI in Globus Toolkit 4

In diesem Abschnitt wird dargestellt, welche Erweiterungen GSI in der Entwicklung von GT 2 zu GT 4 erfuhr. Dies betrifft zum einen die über TLS hinausgehenden Möglichkeiten der Absicherung von Kommunikation zwischen Grid-Komponenten, zum anderen Varianten bei der Authentifizierung und Autorisierung.

¹³ <http://grid.ncsa.uiuc.edu/gssapi-mechglue/openssh/>

¹⁴ <http://meta.cesnet.cz/cms/opencms/en/docs/environment/tokens/globus/>

¹⁵ <http://www.globus.org/toolkit/docs/2.4/datagrid/deliverables/gsiftp-tools.html>

¹⁶ http://www.globus.org/grid_software/data/gridftp.php

¹⁷ <http://www.dutchgrid.nl/Admin/GridCVS/>

	GSI Secure Conversation	GSI Secure Message	GSI Transport
<i>Basis</i>	Nachrichten	Nachrichten	Kanal
<i>Technologie</i>	WS-SecureConversation	WS-Security	TLS
<i>Vertraulichkeit</i>	Ja	Ja	Ja
<i>Integritätsschutz</i>	Ja	Ja	Ja
<i>Delegation</i>	Ja	Nein	Nein
<i>Performanz</i>	gut bei vielen Nachrichten	gut bei wenigen Nachrichten	am besten

Tabelle 1: Vergleich der drei Protection Schemes in GSI

4.1 Schutz der Kommunikation mit Web Services

Seit Globus Toolkit 4 kommen aus dem Bereich der Web Services die Standards *WS-Security*, *WS-Trust* und *WS-SecureConversation* zum Einsatz [Wel05]. Damit ist GSI auch in der Lage, die Kommunikation auf der Ebene einzelner Nachrichten abzusichern. Es wird hierbei nicht der gesamte Datenverkehr, sondern nur die Nutzinhalt (Payload) der Nachrichtenpakete verschlüsselt. Im Vergleich zur Kanal-basierten Absicherung der Kommunikation mittels TLS, ist die Performanz geringer, denn zum einen müssen für jede einzelne Nachricht aufwändige asymmetrische kryptografische Berechnungen erfolgen, zum anderen entsteht durch den Gebrauch von Web Service ein gewisser Overhead. Allerdings erreicht man durch die Nutzung von Public-Key-Verfahren, dass die Authentizität einzelner Nachrichten auch durch Dritte überprüfbar und eine individuelle Verschlüsselung einzelner Nachrichten möglich wird. GSI Secure Conversation baut zunächst einen Security Context zwischen Client und Server auf, welcher von folgenden Nachrichten wieder verwendet wird. Daraus resultiert eine höhere Performanz als bei GSI Secure Message, falls der Overhead für den Aufbau des Security Contexts vernachlässigbar ist

Auf Transportebene kann per TLS die komplette Kommunikation verschlüsselt werden (Kanal-basierte Absicherung). Eine (gegenseitige) Authentifizierung erfolgt mittels Public Key-Verfahren, anschließend kommt Hybridverschlüsselung zum Einsatz. Allerdings gibt es hierbei keinen durch Dritte überprüfbaren Nachweis der Authentizität für einzelne Nachrichten. Tabelle 1 stellt die Eigenschaften der so genannten Protection Schemes in der Übersicht dar.

4.2 Authentifizierung und Autorisierung

Starke Authentifizierung mit X.509-Zertifikaten ist mit allen der drei oben genannten Protection Schemes möglich. Daneben lässt sich prinzipiell auch eine Authentifizierung mit Benutzername und Passwort gemäß dem Standard WS-Security realisieren – wenngleich diese Variante jedoch nur geringe praktische Relevanz besitzt, gibt es doch keine Unterstützung von Datenverschlüsselung bzw. einen Integritätsschutz oder die Möglichkeit zur Delegation. Es besteht außerdem die Möglichkeit einer so genannten anonymen

oder unauthentifizierten Kommunikation; ein Mechanismus, der nützlich sein kann, wenn mehrere Protection Schemes gemeinsam genutzt werden, etwa GSI Secure Conversation (mit X.509-Zertifikat) und anonymer GSI Transport (Verzicht auf redundante Authentifizierung).

Eine weitere Erweiterung in GT 4 betrifft die Mechanismen der Autorisation, die Client und Server zur Verfügung stehen, die über das bereits beschriebene Grid-Mapfile hinausgehen [Sot06]. Der Server kann etwa Aufrufe abhängig davon zulassen, ob die Identität des Besitzers des Services (also der Ressource) und jene des Clients übereinstimmen (so genannte „Self Authorization“). Auch können Services eine einzelne Identität festlegen, deren Zugriff gewährt werden soll („Identity Authorization“). Eine Abschwächung dieses Prinzip besteht darin, den Hostnamen anzugeben, von dem aus Anfragen gestellt werden dürfen („Host Authorization“). Der mächtigste Mechanismus besteht in der Verwendung eines OGSA¹⁸-Authorization-kompatiblen Autorisationsdienstes, der z. B. Aufrufe via SAML (Security Assertion Markup Language) erlaubt. Clientseitig sind einige dieser Mechanismen vorhanden, nämlich wiederum Self Authorization, Identity Authorization sowie Host Authorization. Schließlich gibt es auf beiden Seiten die Möglichkeit, überhaupt auf eine Autorisierung zu verzichten. Es besteht die Möglichkeit mit dem Authorization Framework in GT 4 eigene bedarfsgerechte Mechanismen zur Autorisation einzubinden [LFS+06]. An dieser Stelle lassen sich insbesondere externe Autorisierungsdienste wie VOMS oder Shibboleth anbinden.

5 Zusammenfassung

GSI hat sich nicht nur im Umfeld von Grids etabliert. Einige der darin enthaltenen Konzepte haben auch anderweitig Verbreitung gefunden. Obwohl GSI natürlich keine umfassende Sicherheitslösung für das Grid Computing darstellt, bietet es doch praktikable und nützliche Lösungen und Werkzeuge für alle einige der relevanten Sicherheitsaspekte.

Der Schritt von pre-WS GSI hin zu WS GSI war hauptsächlich von der Vereinheitlichung im Zuge der Integration der Web Services beim Versionssprung von GT 2 zu GT 3 bzw. GT 4 getrieben. Dass sich gerade durch den Einsatz von WS-Mechanismen auch einige Möglichkeiten für die Einbindung neuer Sicherheitsmechanismen ergeben, wurde in Abschnitt 4 gezeigt.

Danksagung Dieser Beitrag baut auf unseren Arbeiten in der Fachgruppe „Netze und Sicherheit“ des D-Grid Integrationsprojekts auf, welches mit Mitteln des BMBF unter dem Kennzeichen 01AK800B gefördert wird. Wir bedanken uns bei den Kollegen der Fachgruppe für Diskussionen über die Thematik GSI und ihre Kommentare zu unseren Beiträgen in [DEF+06].

¹⁸ <http://www.globus.org/ogsa/>

Literaturverzeichnis

- [ABM04] M. Ahsant, J. Basney, O. Mulmo. Grid Delegation Protocol, Workshop on Grid Security Experiences, 2004. <http://www.ncsa.uiuc.edu/~jbasney/Grid-Delegation-Protocol.pdf>
- [BHW05] J. Basney, M. Humphrey, V. Welch. The MyProxy Online Credential Repository. Software: Practice and Experience, Volume 35, Issue 9, July 2005.
- [Con06] Condor Team. Condor-G v6.7 Manual. University of Wisconsin-Madison, Februar 2006.
- [DEF+06] T. Dussa, U. Epting, B. Filipovic, G. Foest, J. Glowka, J. Götze, C. Grimm, M. Hillenbrand, C. Kohlschütter, R. Lohner, S. Makedanz, P. Müller, M. Pattloch, S. Piger, T. Straub, J. Wiebelitz. Aufbau einer AA-Infrastruktur für das D-Grid - Analyse von AA-Infrastrukturen in Grid-Middleware, 2006.
- [DiA199] T. Dierks, C. Allen. The TLS Protocol Version 1.0, RFC 2246, 1999.
- [FaHo02] S. Farrell, R. Housley. An Internet Attribute Certificate Profile for Authorization, RFC 3281, 2002.
- [ITU97] ITU-T Recommendation X.509 (1997 E): Information Technology – Open Systems Interconnection – The Directory: Authentication Framework, June 1997.
- [JTE01] K. Jackson, S. Tuecke, D. Engert. TLS Delegation Protocol, Internet Draft, draft-ietf-tls-delegation-01.txt, 2001.
- [LFS+06] B. Lang, I. Foster, F. Siebenlist, R. Ananthkrishnan, T. Freeman. A Multipolicy Authorization Framework for Grid Security, Proc. Fifth IEEE Symposium on Network Computing and Application, 2006
- [Lin00] J. Linn. Generic Security Service Application Program Interface Version 2, Update 1, RFC 2743, 2000.
- [MWTE04] S. Meder, V. Welch, S. Tuecke, D. Engert. GSS-API Extensions, Grid Security Infrastructure WG, 2004. <http://www.ggf.org/documents/GFD.24.pdf>
- [Sun00] Sun Microsystems. GSS-API Programming Guide, 2000.
- [Sot06] B. Sotomayor. The Globus Toolkit 4 Programmer's Tutorial, 2006.
- [TWE+04] S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, RFC 3820, 2004.
- [Wel05] V. Welch. Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective. <http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf>
- [WFK+04] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, F. Siebenlist. X.509 Proxy Certificates for Dynamic Delegation, 2004.
- [WSF+03] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke. Security for Grid Services, Proc. 12th IEEE Int'l Symposium on High Performance Distributed Computing, 2003.