September 2001

Entwurf, Implementierung und Bewertung eines Kryptographiemoduls für Client-Server Handelssysteme

Stephan Ehren
IBM e-Financial Solutions, stephan.ehren@de.ibm.com

Norbert Ludwig
IBM e-Financial Solutions, norbert.ludwig@de.ibm.com

Boudewijn Haverkort
RWTH Aachen, haverkort@cs.rwth-aachen.de

Rachid El Abdouni Khayari
RWTH Aachen, rachid@cs.rwth-aachen.de

In: Buhl, Hans Ulrich, u.a. (Hg.) 2001. *Information Age Economy*; 5. Internationale Tagung Wirtschaftsinformatik 2001. Heidelberg: Physica-Verlag

ISBN: 3-7908-1427-X

Entwurf, Implementierung und Bewertung eines Kryptographiemoduls für Client-Server Handelssysteme

Stephan Ehren, Norbert Ludwig
IBM e-Financial Solutions

Boudewijn Haverkort, Rachid El Abdouni Khayari
RWTH Aachen

Zusammenfassung: Valuta-Direct® ist ein Client-Server-System für den Online-Handel mit Finanzprodukten. Bisher wurden geschäftskritische Informationen in diesem System ungeschützt über Netzwerke verbreitet und konnten abgehört und manipuliert werden. Die Möglichkeit solcher Sicherheitsverletzungen verhinderte den Einsatz dieses Systems in Internet-Anwendungen. In diesem Beitrag wird eine kryptographische Erweiterung des Kommunikationsmoduls entworfen und implementiert, um die Kommunikationskanäle zwischen Clients und Server zu sichern. Die beschriebene Erweiterung benutzt dazu die netzwerkprotokoll-unabhängige DSNET-Schicht von Valuta-Direct®. Durch die Implementierung des Internet-Standards Transport Layer Security (TLS) in dieser Schicht und die Benutzung neuer kryptographischer Algorithmen werden sichere Kommunikationskanäle eingerichtet. Tests der Erweiterung zeigen, dass deutliche Leistungeinbußen durch die Verwendung kryptographischer Verfahren entstehen.

1 Einführung

Die IBM e-Financial Solutions entwickelt elektronische Client-Server-Handelssysteme für große Kreditinstitute. Diese Systeme verbinden Filialen, Firmen und Privatkunden über Firmennetzwerke oder über das Internet mit der Unternehmenszentrale.

In diesem Beitrag berichten wir über Untersuchungen, Design und Implementierung eines Prototyps eines kryptographischen Moduls, womit sensible Geschäftsdaten vor Dritten vertraulich zu halten und gleichzeitig vor Manipulation zu schützen sind. Durch Einbeziehung digitaler Signaturen kann darüber hinaus die Authentizität der Informationen gesichert werden, um in Streitfällen beweisen zu können, welche Transaktionen von wem getätigt wurden.

Nach [Schn96; Bozo99] sind die folgenden Anforderungen an ein sicheres Kommunikationssystem zu identifizieren: (i) *Authentifizierung*, d.h. die Identitäten der Kommunikationspartner müssen gegenseitig verifizierbar sein; (ii) *Vertraulichkeit* der übermittelten Informationen, d.h. ein Abhören darf nicht möglich sein; (iii) *Datenintegrität*, um Veränderungen der Informationen während der Übertragung auszuschließen; (iv) *Verbindlichkeit*, d.h. der Sender einer Information kann nicht abstreiten, diese zu einem bestimmten Zeitpunkt geschickt zu haben; (v) *Zugangskontrolle*, so dass nur berechtigte Nutzer Zugang zu den Kommunikationsresourcen haben.

Im Bereich der Entwicklung von Finanzhandelssystemen sind auch andere Hersteller tätig. Das System TIBMercury™ der Firma TIBCO Finance Technology bietet verschlüsselte Kommunikation über die Secure Socket Layer (SSL) und eine Authentifikation über Smartcards [Tibco01]. Im System AutoDeal LITE™ von Cognotec™ basiert die Sicherheitsfunktionalität auch auf der SSL und erfordert die Installation von Client-Zertifikaten [Cogn01]. Das Handelssystem GATE™ der Firma Dene ist internet-basiert und nutzt das Protokoll HTTP über SSL (HTTP/S). Der Server authentifiziert sich über Zertifikat, während auf Client-Seite der Benutzer durch einen Benutzernamen und ein Kennwort gegenüber dem System identifiziert wird [Dene01].

Das Papier gliedert sich wie folgt. In Abschnitt 2 wird kurz die Systemarchitektur und Kommunikation von Valuta-Direct® beschrieben. Es folgen in Abschnitt 3 eine Problemanalyse und mögliche Lösungsansätze. Abschnitt 4 konkretisiert die Lösung und es wird die Einbettung eines Kryptographiemoduls in die System- und Kommunikationsarchitektur erläutert. In Abschnitt 5 wird der erstellte Prototyp des beschriebenen Moduls hinsichtlich seiner Leistung bewertet. Abschnitt 6 faßt die erhaltenen Ergebnisse zusammen.

2 Die Valuta-Direct® Systemarchitektur

Mit Hilfe des elektronischen Handelssystems Valuta-Direct® [IBM01] können institutionelle Anleger zeitnah mit standardisierten Finanzprodukten, wie z.B. Aktien, Optionen, Futures, Devisen und Renten handeln. Ebenso können angeschlossene Banken und durch die Internet-Erweiterung NET.Trader® auch Privatkunden das System nutzen. Das Client-Server-Design von Valuta-Direct® stellt ein verteiltes Echtzeit-Informations- und automatisiertes Handelssystem dar. Der Valuta-Direct® Server ist der zentrale Anwendungsserver für die Verarbeitung aller Operationen, die von den Clients im Internet oder Intranet initiiert werden.

Die Hauptmerkmale von Valuta-Direct® sind: (i) *Übernahme von Echtzeit-Informationen über Kurse und Preise* aus angeschlossenen Kurs-Systemen und Weitertransferierung an angeschlossene Clients; (ii) *Online-Handel mit Transaktionsverarbeitung*; (iii) *Integrierte Sicherheit* durch die individuelle Vergabe von Zugriffs-

rechten durch die Zentrale der Bank für jeden Benutzer und Benutzergruppen, sowie zum Schutz von Internet-Transaktionen Benutzung von Verfahren wie PIN- (Persönliche Identifikationsnummer) und TAN (Transaktionsnummer) benutzt. (iv) *Vielfältige Kommunikationsmöglickeiten* durch Middleware-Software, die Integration verschiedener Banksysteme und die Anbindung von unterschiedlichen Netzwerken.

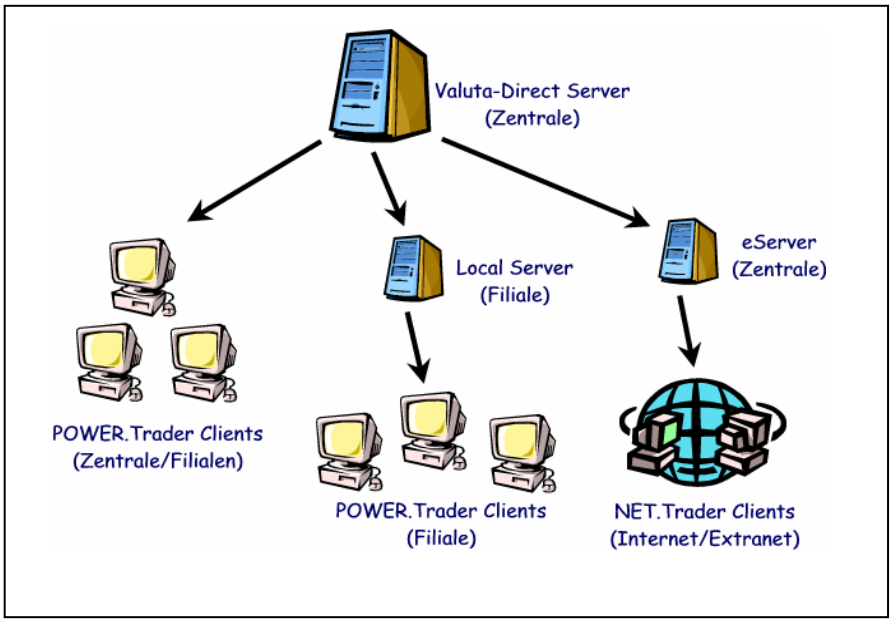

Abbildung 1: Die Valuta-Direct® Systemkomponenten

Die folgenden Komponenten bilden das Valuta-Direct® System (s. Abb. 1): (i) Der zentrale *Valuta-Direct® Datenbank- und Anwendungsserver*, wobei die Datenbank- und die Anwendungsserver-Komponenten auf verschiedene Maschinen verteilt werden können; (ii) die *POWER.Trader™ Clients*, welche über ein Kommunikationsnetzwerk (LAN oder WAN) an den Valuta-Direct® Server angeschlossen sind, werden im zentralen Handelsraum einer Bank und in deren Filialen benutzt; (iii) ein *Local Server* wird optional in Filialen eingesetzt, um den Netzwerkverkehr zwischen den POWER.Trader Clients in der Filiale und dem zentralen Valuta-Direct® Server zu reduzieren; (iv) der (hier nicht betrachtete) *NET.Trader® Client* wird im Online-Handel via Internet eingesetzt; (v) der *eServer* verbindet die NET.Trader® Clients mit dem System.

Für die **Kommunikation** wird TCP/IP als Standard-Protokoll benutzt, aber auch NetBIOS und APPC (IBM SNA LU 6.2) können benutzt werden.

POWER.Trader™ Clients bauen zwei logische Kommunikationskanäle zum Server auf, eine für Benutzeranfragen und die andere für den Empfang von nicht explizit angeforderten Messages (Broadcasts) vom Server (z.B. Kursaktualisierungen).

Es werden zwei Kommunikationsarten zwischen POWER.Trader™ Client und den Servern unterschieden: (i) Die *synchrone Kommunikation* ist für alle vom Client initiierten Message-Transfers zwischen Client und Server verantwortlich. Dies umfaßt die Vermittlung von allen "synchronen" Client-Requests und den sich daraus ergebenden Server-Antworten. Jeder POWER.Trader™ Client ist mit dem Server über eine einzelne Netzwerk-Session verbunden, die für den Transfer von allen Messages zwischen Client und Server benutzt wird. (ii) Die *Broadcast-Kommunikation* ist zuständig für die Verteilung von nicht explizit angeforderten Messages vom Server an die Clients, z.B. zur Übertragung von aktualisierten Kursdaten und Datenbank-Updates, die an die Client weiterverteilt werden.

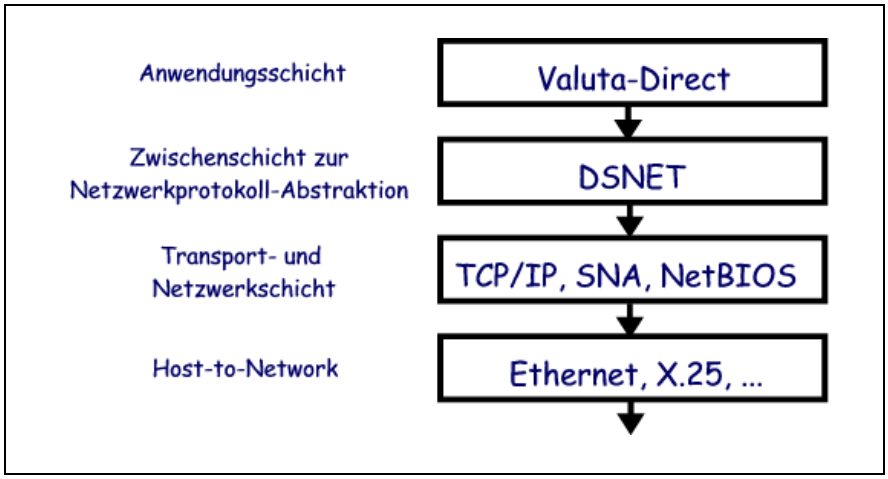

Abbildung 2: Das Valuta-Direct® Kommunikationsschichtenmodell

Alle Kommunikationsprozesse auf POWER.Trader™ Clients und Valuta-Direct® Server basieren auf der hauseigenen Softwarebibliothek DSNET Library (Data Sciences NETwork), die die Netzwerkfunktionalität in einer Zwischenschicht zwischen dem Anwendungsprotokoll von Valuta-Direct® und dem jeweils verwendeten Netzwerkprotokoll kapselt (s. Abb. 2). Die DSNET bietet allgemeine Funktionen für Verbindungsinitialisierung und -abbau, sowie für Datenübertragung mittels verbindungsorientierter und verbindungsloser Kommunikation an. Diese Funktionalität abstrahiert vom eigentlich benutzten Netzwerkprotokoll, welches über die Konfiguration festgelegt wird, und derzeit TCP/IP, SNA oder NetBIOS sein kann.

3 Problemstellung und Lösungsansätze

3.1 Sicherheitstechnische Analyse von Valuta-Direct®

Sämtliche Kommunikation zwischen POWER.Trader™ Clients, den Local Servern und dem Valuta-Direct® Server findet ungeschützt statt. Ein möglicher Angreifer des Systems kann sich durch Abhören der Kommunikation Kenntnis über gemachte Geschäfte verschaffen und eventuell durch geschickte Manipulation falsche Geschäftsinformationen in das System einschleusen. Durch Verwendung des Systems in geschlossenen Netzwerken (LAN und WAN des Systembetreibers) war bisher die Sicherheitsfrage sekundär. Die Nutzung des Internets zur Kosteneinsparung und zur schnelleren Kundenanbindung, vor allem im Business-to-Business-Bereich (B2B), macht dies nun aber zu einem primären Problemfeld.

Das Valuta-Direct® System ist an mehreren Stellen angreifbar: (i) Die Authentifizierung vor der Benutzung des POWER.Trader™ Clients schützt nur vor unautho-risierten Gebrauch der Client-Software, aber durch Simulation des Messageverkehrs Server und Client kann das System manipuliert werden, ohne dass eine Benutzerauthentifizierung stattgefunden hat; (ii) durch Einschleusen falscher Kursinformationen können Benutzer getäuscht werden, da die Authentität einer Message nicht zweifelsfrei bestimmt werden kann; (iii) eine Vertraulichkeit der übermittelten Informationen ist nicht gegeben; (iv) mittels Abfangen und Verändern von Messages (aktiver Angriff) sind beispielsweise Geschäftsabschlüsse unter anderem Namen möglich. ; (v) die Zugangskontrolle zum System kann mittels Abhören der Zugangsinformationen (Username, Passwort) übergangen werden.

3.2 Lösungsansätze

Um ein sicheres Kommunikationssystem zu etablieren ist ein Protokoll nötig, welches sichere Kommunikation ermöglicht. Gemeinsam ist dabei allen entwickelten *sicheren Kommunikationsprotokollen*, dass sie kryptographische Verfahren benutzen, um ihre Sicherheitsmerkmale zu erreichen. Unterschiede finden sich in der Anordnung im Netzwerk-Schichtenmodell und in der Benutzung der kryptographischen Verfahren. Protokolle für sichere Kommunikation sind z.B. IPSEC [Atk95], Secure Sockets Layer (SSL) [FrKa96], und dessen Nachfolger Transport Layer Security (TLS) [DiAl99].

Symmetrische Verschlüsselungmethoden [Schn96] können Daten vor Abhören und Manipulation schützen, bieten sich also an, um die Vertraulichkeit und Datenintegrität im Valuta-Direct® System sicherzustellen. Problematisch ist die initiale Einigung auf einen gemeinsamen Schlüssel zwischen Client und Server, denn dieser muß auf einem sicheren Kommunikationsweg übertragen werden.

Public-Key-Verfahren benutzen zwei verschiedene Schlüssel und lösen das Problem der Schlüsselübergabe: Einen öffentlich bekannten Schlüssel - den Public Key - zum Verschlüsseln von Daten und einen nur dem Empfänger von Nachrichten bekannten geheimen Schlüssel - den Private Key - zum Dechiffrieren von Daten. Von Nachteil ist, dass asymmetrische Verfahren deutlich langsamer (etwa um den Faktor 1000) sind. Außerdem müssen zur Authentizitätsicherung der Public Keys Zertifikate eingesetzt werden.

Darüberhinaus ermöglichen viele Public-Key-Verfahren digitale Signaturen, die die Authentität, Integrität und Verbindlichkeit von Dokumenten gewährleisten können. In Valuta-Direct® könnten sie z.B. zur Unterzeichnung von Geschäftsaufträgen verwendet werden. In den für den Schlüsselaustausch wichtigen Zertifikaten werden die öffentlichen Schlüssel eines Kommunikationspartners von einem vertrauenswürdigen Dritten (*Trustcenter* oder *Certification Authority (CA)*) digital signiert.

Kryptographische Hashfunktionen (auch Einweg-Hashfunktionen genannt) dienen dazu "Fingerabdrücke" von Nachrichten zu erzeugen, die als Eingabe für eine digitale Signatur dienen können. Da sie mit hinreichender Sicherheit eine Nachricht identifizieren, können diese anstatt der gesamten Nachricht signiert werden, was den Signaturprozess insbesondere bei großen Nachrichten wesentlich beschleunigt. Soll nur der Empfänger einer Nachricht deren Integrität überprüfen können, kommt ein sogenannter *Message Authentication Code (MAC)* ins Spiel, dies ist eine kryptographische Hashfunktion mit der Ergänzung eines geheimen Schlüssels. Der Hashwert ist dann eine Funktion der Nachricht und des Schlüssels, und nur die Inhaber des Schlüssels können die Nachricht verifizieren. Im Gegensatz zu digitalen Signaturen werden die MACs aber nicht zu späteren Prüfzwecken mitabgespeichert.

4 Einbettung in die Systemarchitektur

4.1 Einbettung in das Valuta-Direct® Schichtenmodell

Die Einbettung einer Erweiterung für sichere Kommunikation (Kryptographie-modul) in Valuta-Direct® hängt zu einem großen Teil von der Frage ab, wo es im Schichtenmodell angesiedelt wird. Mögliche Lösungen liegen in der Netzwerkschicht (z.B. IPSEC) oder zwischen Transport- und Anwendungsschicht (z.B. SSL/TLS). Auch eine Verschlüsselung und Signatur auf der Anwendungsschicht von Valuta-Direct® ist möglich, dies würde auch viele Kontrollmöglichkeiten erlauben. Beispielsweise könnte die Signatur eines Benutzers zu einem Geschäftsabschluß überprüft werden. Der Zugriff auf die Informationen, dass es sich um eine

Message des Typs Geschäftsabschluß handelt, und die Kontrollmöglichkeit steht nur den Prozessen der Geschäftslogik zur Verfügung, und diese arbeiten auf der Valuta-Direct® Anwendungsschicht. Nachteil der Implementierung auf der Anwendungsschicht ist, dass jeder Prozess der Geschäftslogik geändert werden muß, was aufgrund der Vielzahl der dazu nötigen Modifikationen nicht praktikabel ist.

Kryptographielösungen auf der Netzwerk- und der Transportschicht haben den Vorteil, dass an der Valuta-Direct® Software keine Änderung vorgenommen werden muss, allerdings sind sie netzprotokollabhängig und erlauben keine Steuerung, welche Datenpakete verschlüsselt oder signiert werden sollen, sondern behandeln alle Pakete gleich, was z.B. beim Einsatz von hochfrequenten Broadcasts durch die entstehende Systemlast für kryptographischen Operationen problematisch werden kann.

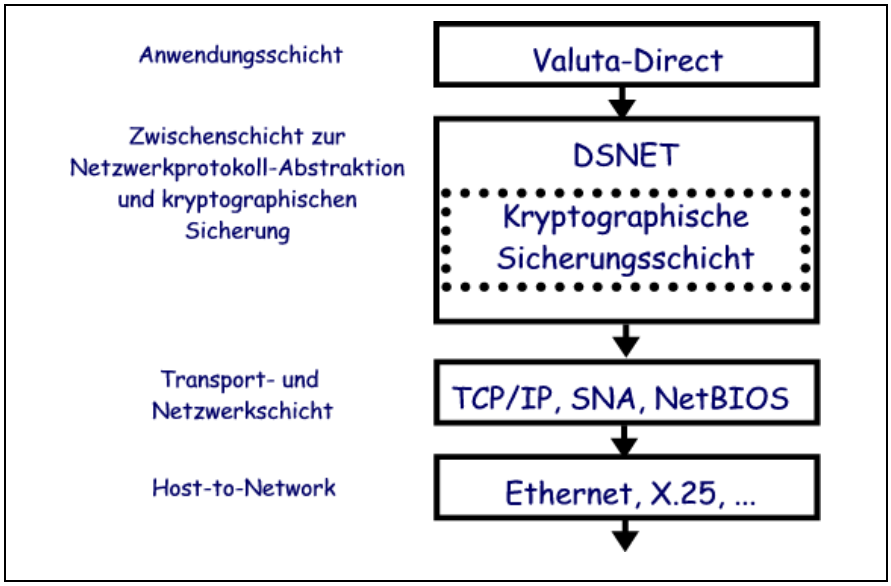

Abbildung 3: Einbettung eines Kryptographie-Moduls in die DSNET-Schicht

Der Einsatz eines Kryptographiemoduls zwischen Anwendungs- und Transportschicht, also bei Valuta-Direct® auf der DSNET-Schicht, scheint am sinnvollsten zu sein, denn damit beschränkt sich der Änderungsaufwand auf die Softwaremodule, die diesen Dienst implementieren und auf die Prozesse, die die Schnittstellen dieser Schicht benutzen (s. Abb. 3).

Die digitale Signatur von Geschäfts-Transaktionen ist auf dieser Schicht aber nicht möglich, da dort die unterschiedlichen Messagetypen nicht unterschieden werden können. Eine digitale Signatur macht nur Sinn unter einem “elektronischen Dokument” und nicht unter dem Datenstrom, mit dem sich die DSNET-Schicht befaßt.

Daher muß eine digitale Signatur unter Geschäftsabschlüsse o.ä. von der Anwendungsschicht verarbeitet werden, wie auch in [Schm98] festgestellt wird.

Für die kryptographische Sicherung zwischen der Anwendungs- und Transportschicht gibt es bereits das TLS-Protokoll, welches in [DiAl99] spezifiziert wird. TLS ist keine bereits fertige Lösung, sondern stellt ein Protokoll-Framework zur Verfügung, welches von den Softwareherstellern für ihr Produkt implementiert werden kann. Durch die Unabhängigkeit von der Wahl eines Transportschicht-Protokolls ist TLS für den Einsatz außerhalb von TCP/IP-Netzen geeignet.

4.2 Einbettung in das Valuta-Direct® System

Wie im letzten Unterkapitel gezeigt wurde, kann die DSNET-Schicht mit Hilfe von TLS um kryptographische Verfahren erweitert werden, und es werden nur an wenigen Stellen in dem Kommunikationssubsystem Änderungen nötig.

Die DSNET-Schicht wird durch die DSNET-Softwarebibliothek implementiert. Hier ist die Funktionalität von TLS zu ergänzen. Die DSNET-Library wird sowohl vom Valuta-Direct® Server als auch vom POWER.Trader™ Client benutzt, so dass keine getrennte Implementation für Client und Server stattfinden muß. Darüberhinaus kann ein durch Point-to-Point-Verbindung initiierter Kryptographie-Status, d.h. die ausgehandelten kryptographischen Verfahren und Schlüssel, auch für die Broadcast-Kommunikation verwendet werden.

Da die Kommunikationsprozesse zur Netzwerkkommunikation nur Methoden der DSNET-Softwarebibliothek aufrufen, sind an diesen keine Änderungen nötig. Die Steuerung der kryptographischen Parameter (Art der Verschlüsselung, Schlüssellängen, usw.) kann über eine Erweiterung der bereits vorhandenen DSNET-Konfiguration erfolgen.

Für den Einsatz eines Kryptographiemoduls mit den geforderten Eigenschaften starke Verschlüsselung, digitale Signatur auf der Anwendungsschicht, sowie dem sicheren Schlüsselaustausch über Public-Key-Verfahren, ist eine Certification Authority (CA) für die Verwaltung von Server- und Client-Zertifikaten notwendig. Da prinzipiell die CA unabhängig von Valuta-Direct® sein sollte, können deren Dienstleitungen auch von Dritten betrieben werden.

Die Clients und der Server benötigen das sogenannte Root-Zertifikat der CA, um die von ihr signierten Zertifikate des Server bzw. der Clients zu verifizieren. Dieses Zertifikat sollte bei der Installation und Konfiguration mitinstalliert werden. Mit Hilfe des vorkonfigurierten Zertifikats der CA können alle Transaktionen verifiziert werden, die von einem von ihr herausgegebenen Zertifikat gemacht werden.

Darüberhinaus sind noch private Schlüssel bei Client und Server notwendig, damit sich diese gegenüber dem jeweiligen Kommunikationspartner authentifizieren können, und evtl. in Zukunft digitale Signaturen möglich werden. Diese privaten

Schlüssel sind geheim zu halten und bestmöglich zu schützen, z.B. durch Speicherung auf Smartcards bei den Clients.

TLS selber beinhaltet keine kryptographischen Verfahren, sondern definiert nur einige bewährte Methoden (z.B. Triple-DES mit RSA-Schlüsselaustausch), die benutzt werden können. Um den inzwischen verbesserten Verschlüsselungs-algorithmen wie z.B. AES Rechnung zu tragen, wird die TLS-Implementation innerhalb der DSNET um diese neuere Verfahren erweitert. Da die Ciphersuite (= kryptographische Bibliothek) in TLS als "Blackbox" gesehen wird, ist dies problemlos machbar.

4.3 Details der Implementierung

Das in der DSNET-Schicht lokalisierte Kryptographiemodul wird in zwei Untermodule aufgeteilt: (i) Einem TLS-Modul, welches die Funktionalität von TLS in Form einer Klassenbibliothek zur Verfügung stellt; (ii) das Ciphersuite-Modul, auch in Form einer Klassenbibliothek, welches die kryptographischen Verfahren kapselt.

Beide Klassenbibliotheken werden in weitesgehend plattformunabhängigen C++ implementiert und für alle Zielplattformen übersetzt.

Die Implementation beschränkt sich allerdings auf ein Prototyp-Modul, welches bisher nur die folgenden Verfahren in das System integriert: (i) Das Kommunikationsprotokoll TLS in der aktuellen Version 1, angepaßt an neuere symmetrische Kryptographieverfahren wie AES und Twofish; (ii) die symmetrischen Verschlüsselungsverfahren AES und Twofish in Form der Referenzimplementationen ihrer Entwickler; (iii) die Hashalgorithmen SHA und MD5, ebenfalls in Form der Referenzimplementationen; (iv) das Public-Key-Verfahren RSA (Schlüssellänge von 768 bis 4096 Bit).

5 Bewertung des neuen sicheren Systems

Um die Leistung des Prototyps des Kryptographiemoduls zu bewerten wird zuerst das Vorgehen bei der Bewertung vorgestellt. Danach wird mit Hilfe der Testergebnisse überprüft, ob alle an das Modul gestellten Anforderungen befriedigt werden.

5.1 Vorgehen bei der Bewertung

Neben den während und nach der Implementierung erfolgten funktionellen Tests, werden mehrere Datendurchsatzmessungen durchgeführt, um die Leistungsfähig-

keiten des Kryptographiemoduls zu bewerten. Dazu bietet sich die Messung der *Latenzzeit*, d.h. der Verweildauer einer Message im Kryptographiemodul und der DSNET-Schicht, sowie der *Übertragungsrate*, d.h. der Menge der Daten, die pro Zeiteinheit durch das Modul gehen können, an.

Es werden dazu drei Betriebsmodi verglichen: (i) nur Verwendung der DSNET, d.h. dem Ausgangspunkt; (ii) zusätzliche Verwendung des Kryptographiemoduls (aktivierte TLS-Schicht) aber ohne Verwendung von Verschlüsselung und MAC, d.h. der Nullcipher, (iii) mit Benutzung des Kryptographiemoduls mit aktivierter Verschlüsselung und MAC (asymmetrische Verschlüsselung mit RSA, symmetrische Verschlüs-selung mit Twofish (128 Bit), MAC konstruiert mit dem Secure Hash Algorithm).

Durch unterschiedliche Messagegrößen (100, 250, 500, 1000, 2500, 5000, 10000, 25000, 50000 Bytes) werden die in der Praxis vorkommenden Kommunikationssituationen simuliert.

Jeweils 10 Millionen Bytes an Netto-Daten (vor der Messung generiert) werden pro Messung von der Client-Anwendung an den Server geschickt. Die Zeit für die Übertragung der 10 Millionen Bytes Netto-Daten wird von der Testanwendung auf Client-Seite gemessen, die Server-Anwendung bestätigt den Erhalt und führt eine Kontrollzeitmessung vom Beginn bis zum Ende der Übertragung durch.

Durch die Messung wird die Latenzzeit (in s) bestimmt. Desweiteren wird die Übertragungsrate (in MBit/s) bestimmt. Der durch die TLS-Schicht zusätzliche Overhead zu den Daten wird *nicht* zu den 10^7 Bytes in die DSNET-Schicht eintretenden Daten hinzugezählt. Jede Messung wird 10 mal wiederholt, um möglichst zuverlässige Werte zu ermitteln und Fehler durch Messungsartefakte zu vermeiden. Schwankungen lagen typischerweise unterhalb von 1%.

Die Testumgebung für das Kryptographiemodul besteht aus den folgenden Komponenten: (i) zwei identische PCs mit Intel Pentium 200, 64MB RAM, 10MBit-Netzwerkkarte; (ii) eine kleine Client/Server-Anwendung, die die DSNET mit der Kryptographie-Erweiterung benutzt und die Testwerte mißt; (iii) ein dediziertes 10MBit-Ethernet-Netzwerk.

5.2 Testergebnisse

Betrachten wir zuerst die Latenzzeiten in Abbildung 4. Bei größeren Messages (ab 2500 Bytes) stößt die DSNET-Schicht an die Grenzen des 10MBit-Netzwerkes, welche bei einem Datendurchsatz von ungefähr 8 MBit/s liegen. Bei kleinen Messages ist der Overhead so groß, daß der Durchsatz bis auf unter 20% des Maximalwertes sinkt bzw. die Latenzzeit um mehr als das fünffache ansteigt.

Bei Verwendung der TLS-Schicht ohne aktivierte Verschlüsselung machen sich sehr kleine Verzögerungen durch den zusätzlichen Daten-Overhead von 7 Bytes

pro Message und durch die zusätzlichen Funktionsaufrufe bemerkbar. Die Leistungsverluste bewegen sich im Rahmen von bis zu 4% bei einer Messagegröße von 100 Bytes. Bei den großen Messages ist kein Unterschied zu bemerken, da man sich hier an der Leistungsgrenze des Netzwerks bewegt.

Die aktivierte Verschlüsselung bremst die Datenübertragung im Kryptographiemodul stark. Während bei kleinen Messages die Latenzzeit schon um 44% gegenüber der bei alleiniger Benutzung der DSNET-Schicht ansteigt, ist dies bei großen Paketen schon 60%, da hier mehr zu verschlüsseln und zu hashen ist. Insbesondere ist der verwendete Hashalgorithmus ein Bremsfaktor, denn er muss für das HMAC-Konstrukt, welches in TLS verwendet wird, 2 mal über der gesamten Message ausgeführt werden. Auch der Overhead vergrößert sich durch den Hashalgorithmus gegenüber alleiniger Benutzung von TLS um weitere 20 Bytes pro Message, die für den Hashwert benötigt werden.

Durchsatzverluste von 2 Größenordnungen beim Einsatz von TLS auf Webservern, wie sie in [ApPe99] angesprochen werden, haben wir nicht feststellen können.

Abbildung 5 visualisiert die erhaltenen Durchsätze. Erkennbar ist eine obere Begrenzung der maximalen Übertragungsrate in Höhe von ca. 8 MBit/s bei alleiniger Benutzung der DSNET oder des Kryptographiemoduls mit Nullcipher. Entsprechend ist eine untere Begrenzung bei der minimalen Latenzzeit in Höhe von ca. 10 Sekunden pro 10^7 Bytes zu finden. Bei Benutzung des Kryptographiemoduls mit Verschlüsselung und Hashfunktion geht die Latenzzeit bei größeren Messages gegen 16 Sekunden, während sich die Übertragungsrate 5 MBits/s nähert.

Die Leistungseinbußen durch das Kryptographiemodul werden sich hauptsächlich bei Verwendung in einem LAN herausstellen, denn hier wird die maximale Übertragungsrate deutlich eingeschränkt. Bei Verwendung im WAN mit den im Kundenumfeld üblichen Bandbreiten bis 2 MBit/s werden nur die Latenzzeiten im Modul leicht spürbar sein, denn selbst die maximale Übertragungsrate des Kryptographiemoduls bietet hier noch genug Reserven.

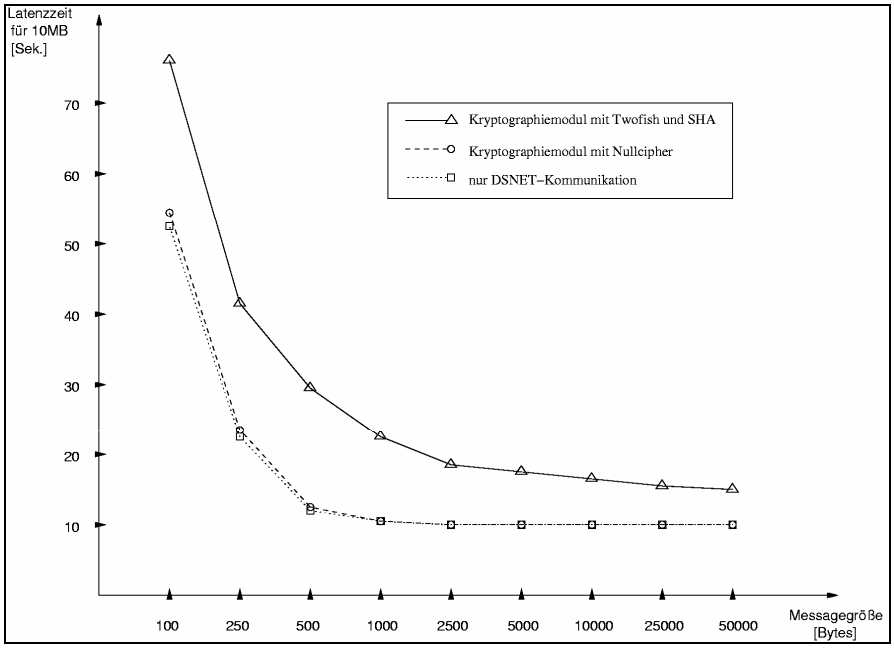

Abbildung 4: Latenzzeiten

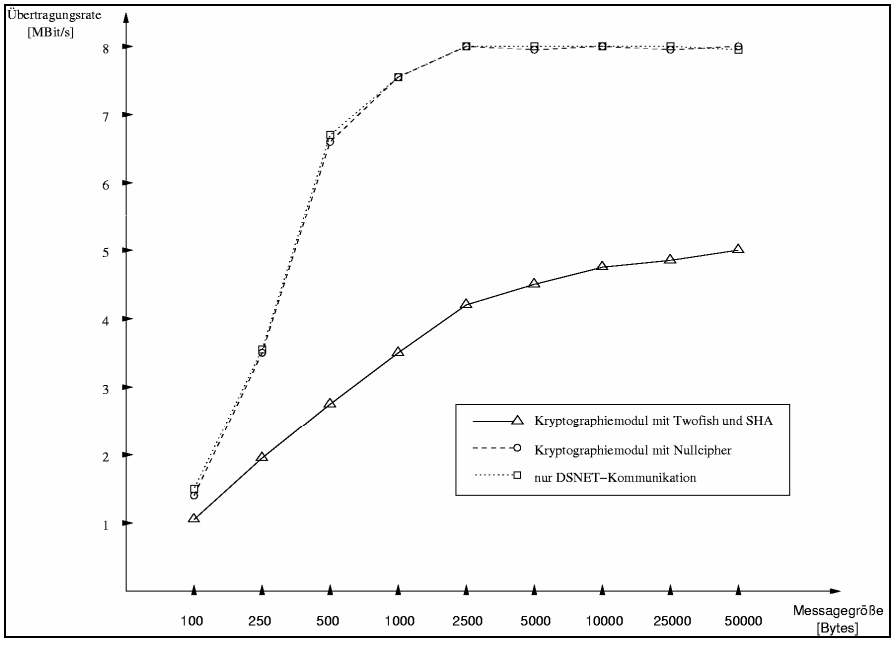

Abbildung 5: Durchsatzraten

6 Zusammenfassung und Ausblick

Die Anforderungen an das Kryptographiemodul werden durch den Prototyp zu einem großen Teil befriedigt. Der *Schutz von Kommunikationsdaten durch Kryptographie* wird durch die kryptographische Sicherungsschicht mittels TLS im Kryptographiemodul und durch die Integration in die DSNET gewährleistet. Die Forderung nach *Systemtransparenz* wird durch die Integration des Kryptographiemoduls innerhalb der DSNET-Schicht befriedigt. Der *Datendurchsatz* des Gesamtsystems wird durch die zusätzliche kryptographische Sicherung zwar verschlechtert, dies ist allerdings durch den Einsatz zusätzlicher Hardware für die kryptographischen Operationen ausgleichbar.

Der *administrative Aufwand beim Kunden* wird allenfalls durch die aufzubauende Certification Authority für die Zertifikate vergrößert. Soweit möglich wurden *Lösungen von Drittanbietern* in der Entwicklung des Kryptographiemoduls miteinbezogen, was insbesondere bei dem Ciphersuite-Modul Sinn macht.

Im Vergleich zu bereits bestehenden Lösungen auf dem Markt, wie z.B. der Open Source Kryptographielösung OpenSSL [OSSL01], ist das Kryptographiemodul besser auf die Erfordernisse von Valuta-Direct® eingestellt. Durch die Integration in die DSNET wird eine weitgehende Netzprotokolltransparenz erreicht, während OpenSSL auf TCP/IP festgelegt ist.

Im Vergleich der kryptographischen Verfahren ist das Kryptographiemodul gegenüber anderen Lösungen fortschrittlich, denn es werden die sehr sicheren und leistungsfähigen Methoden aus der AES-Ausschreibung benutzt. Bei den asymmetrischen Verfahren wird im Prototyp noch der renommierte RSA-Algorithmus verwendet.

Literatur

[ApPe99] Apostolopoulos, G.; Peris, V.; Saha, D.: TLS : How much does it really cost ?. In: Proceeding of INFOCOM'99, New York März 1999.

[Atk95] Atkinson, R.: RFC 1825: Security Architecture for the Internet Protocol. http://www.ietf.org/rfc/rfc1825.txt, Abruf am 2001-02-28.

[Bozo99] Bozoki, E.:: IP Security Protocols. In: Dr. Dobb's Journal (1999) 12, S. 42-55.

[Cogn01] Cognotec Ltd.: Internet Website. http://www.cognotec.com, Abruf am 2001-02-28.

[Dene01] Dene International: Internet Website. http://www.dene.com, Abruf am 2001-02-28.

[DaRi99] Daemen, J.; Rijmen, V.: The Rijndael Page. http://www.esat.kuleuven.ac.be/~rijmen/rijndael, Abruf am 2001-02-28.

[DiAl99] Dierks, T.; Allen, C.: RFC 2246: The TLS Protocol Version 1. http://www.ietf.org/rfc/rfc2246.txt, Abruf am 2001-02-28.

[FrKa96] Freier, A.; Karlton, P.; Kocher P.: The SSL Protocol Version 3.0. ftp://ftp.netscape.com/pub/review/ssl-spec.tar.Z, Abruf am 2001-02-28.

[IBM01]IBM Corp.: Informationen zu Valuta-Direct®. http://www.ibm.com/services/de/e-business/financial_trading.html, Abruf am 2001-02-28.

[KrBe97] Krawczyk, H.; Bellare, M.; Canetti, R.: RFC 2104: HMAC: Keyed-Hashing for Message Authentication. http://www.ietf.org/rfc/rfc2104.txt, Abruf am 2001-02-28.

[OSSL01] The OpenSSL Project: The Open Source toolkit for SSL/TLS. http://www.openssl.org, Abruf am 2001-02-28.

[Schm98] chmeh, K.: Einer paßt – Krypto-Protokolle für das Internet. In: iX Magazin (1998) 12, S. 113-117.

[Schn96] Schneier, B.: Applied Cryptography. Wiley, New York 1996.

[Tibco01] Tibco Finance Technology Inc.: Internet Website. http://www.tibcofinance.com, Abruf am 2001-02-28.