Marcel Kronberg

Implementierung einer Iris-Biometrik in ein „Client-Server-Authentisierungssystem“

Diplomarbeit

Universität Hamburg
Fachbereich Informatik
Arbeitsbereich Anwendungen der Informatik in Geistes und Naturwissenschaften (AGN)

Begutachtet durch:
Prof. Dr. rer. nat. Klaus Brunnstein
Dr. Martin Lehmann

Betreuer: Dipl.Inform. Arslan Brömme

Juni 2002

Erklärung:
Ich versichere, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe und keine außer den angegebenen Quellen und Hilfsmitteln benutzt habe.

Hamburg, den 27.5.2002 **Marcel Kronberg**

Danksagung:

Ich danke Herrn Prof. Dr. Klaus Brunnstein für die mir gegebene Möglichkeit das Thema zu bearbeiten und die Betreuung der Diplomarbeit. Weiterhin danke ich Herrn Dr. Martin Lehmann und Herrn Dipl.-Inform. Arslan Brömme, für die mir zuteil gewordene Unterstützung.

Inhaltsverzeichnis

Abbildungsverzeichnis

ABBILDUNGSVERZEICHNIS

Abkürzungsverzeichnis

AP	Authentisierungspaket
API	Application Programming Interface, Anwendungsprogrammierschnittstelle
AS	Authentisierungs-Server
BFM	Biometrisches Funktionsmodul
CSRSS.EXE	Windows 2000 Systemprozess des Client-Server-Subsystems (Win32 Umgebungssubsystem)
CS Austausch	Client-Server-Austausch
DTM	Datenübertragungsmodul, Datentransfermodul
EXPLORER.EXE	Windows 2000 Systemprozess, der die grafische Benutzershell zur Verfügung stellt
FAR	False Acceptance Rate
FRR	False Rejection Rate
GINA	Graphical Identification and Authentication Interface, Grafische Identifikations- und Authentisierungsschnittstelle
HAL.DLL	Hardware Abstraction Layer, Hardwareabstraktionsschicht
ID	Identifier, Identifikator
IRINA	Iris-Identifikations- und Authentisierungsschnittstelle/ modul
KERBEROS.DLL	Kerberos Authentisierungspaket
KsecDD	Kernel Security Device Driver, Kernel Sicherheitsgerätetreiber
KSECDD.SYS	Komponente, die KsecDD zur Verfügung stellt
LPC	Local Procedure Call, Lokaler Prozeduraufruf
LSA	Local Security Authority, Lokale Sicherheitsautorität
LSASRV.DLL	Dienst der lokalen Sicherheitsautorität

ABKÜRZUNGSVERZEICHNIS

LSASS.EXE	Windows 2000 Systemprozess der lokalen Sicherheitsautorität
MIT	Massachusetts Institute of Technology
MSGINA.DLL	Microsoftversion der GINA.DLL
MSDN	Microsoft Development Network, Microsoft Entwicklungsnetzwerk
MSDOS	Microsoft Disk Operating System
MSV1_0.DLL	NTLM Authentisierungspaket
NTDSA.DLL	Active Directory Verzeichnisdienstagent
NETLOGON.DLL	Anmeldedienst
NT	New Technology
NTDLL.DLL	Windows 2000 Bibliothek, die den Komponenten des Benutzer-Modus die Funktionen des Betriebssystemkerns zugänglich macht.
NTLM	New Technology Lan Manager
NTOSKRNL.EXE	Windows 2000 Betriebssystemkern
OS2SS.EXE	Windows 2000 Systemprozess des OS/2 Umgebungssubsystems
PIN	Persönliche Identifikationsnummer
PSXSS.EXE	Windows 2000 Systemprozess des POSIX Umgebungssubsystems
REFDBM	Referenzdatenbank- managementmodul
REGEDIT.EXE	Registrierungseditor
SAM	Security Account Manager, Sicherheitskonten-Manager
SAMSRV.DLL	Komponente, die Zugang zum SAM bereitstellt
SAS	Secure Attention Sequence
SDK	Software Development Kit, Software Entwicklungs-Kit
SERVICES.EXE	Windows 2000 Systemprozess des Dienststeuerungs-Managers
SID	Security Identifier
SSP	Security Support Provider
SMSS.EXE	Windows 2000 Systemprozess des Sitzungs-Managers
SSP	Security Support Provider

SSPI	Security Support Provider Interface
ST	Server Tickets
TGS	Ticket-Granting-Server
TCSEC	Trusted Computer System Evaluation Criteria
TGT	Ticket-Granting-Ticket
UID	User-Identifier
WIN32K.SYS	Komponente des Win32 Umgebungssubsystems, die im Kernel-Modus ausgeführt wird.
WINLOGON.EXE	Interaktiver Anmeldeprozess

Kapitel 1

Einleitung

Die Bestimmung der Identität von Personen findet heute überwiegend in Client-Server Umgebungen mit Hilfe von Authentisierungsservern statt, die mit bestimmten Verfahren die Identität einer Person feststellen und prüfen. Hier kommen in der Regel Kennwortmechanismen im Zusammenhang mit kryptografischen Verfahren zur Anwendung, bei denen die Identität einer Person durch eine Benutzerkennung festgestellt und mit einem geheimen Kennwort überprüft wird. Diese Mechanismen gewährleisten jedoch nur, dass eine Person über das Wissen eines Kennworts verfügt, aber nicht ob es sich tatsächlich um die entsprechende Person handelt, da das Merkmal zur Überprüfung der Identität zwar auf eine bestimmte Person bezogen, aber nicht an diese Person gebunden ist. Mit dem Wissen von Benutzerkennungen und den dazugehörigen Kennwörtern ist es leicht möglich, in Computersystemen eine andere Identität anzunehmen und auf vertrauliche Information zuzugreifen.

Einen Ausweg bieten hier die biometrischen Verfahren, die zur Identitätsbestimmung fest an Lebewesen gebundene biometrische Merkmale benutzen, wobei ein biometrisches Merkmal eine messbare physiologische oder verhaltensbasierte Charakteristik eines Lebewesens ist. Insbesondere die Iris eines Menschen, die sich geschützt, aber dennoch sichtbar im Auge befindet, eignet sich gut als Merkmal zur Identitätsbestimmung, da sie eine besonders große Merkmalscharakteristik aufweist.

Im Rahmen der *Biometrik Research Group* des Arbeitsbereichs Anwendungen der Informatik in Geistes- und Naturwissenschaften des Fachbereichs Informatik der Universität Hamburg wird an einem neuen biometrischen Verfahren zur Identitätsbestimmung mit Hilfe der menschlichen Iris geforscht. Neben der Entwicklung und Implementierung eines neuen Verfahrens zur Erkennung einer mensch-

lichen Iris, welches im weiteren Kontext *Iris-Biometrik* genannt wird, spielt die Implementierung von biometrischen Verfahren in die Sicherheitssysteme bestehender Computersysteme eine große Rolle.

Diese Arbeit stellt die Implementierung einer Iris-Biometrik in ein „Client-Server-Authentisierungssystem“ dar, wobei das Client-Server-Authentisierungssystem als Komponente eines Sicherheitssystems die Zugangskontrolle realisiert und die Prüfung der Identität einer Person zur Verfügung stellt. Dabei soll vor allem die Frage beantwortet werden, ob und inwieweit die biometrischen Verfahren und insbesondere die Iris-Biometrik in vorhandene Systeme zur Prüfung der Identität einer Person (Authentisierungssysteme) implementiert werden können und welche Wechselwirkungen dabei mit den bestehenden System entstehen. Anhand von Konzepten sollen die Integrationsmöglichkeiten dargestellt und die Vor- und Nachteile der Integrationsmöglichkeiten betrachtet werden. Darüber hinaus soll die Umsetzung einer Integrationsmöglichkeit anhand einer prototypischen Implementierung in Windows 2000 gezeigt werden.

Dazu stellt Kapitel 2 allgemein die Identitätsbestimmung in Computersystemen dar. Dabei wird der technische Vorgang der Authentisierung und die Verfahren zur Identitätsbestimmung mit Hilfe von personenbezogenen Merkmalen und personengebundenen biometrischen Merkmalen erläutert. Anhand der Iris-Biometrik wird beispielhaft dargestellt, wie eine Identitätsbestimmung mit Hilfe der Iris stattfindet. Anschließend wird im Kapitel 3 auf die technischen Grundlagen der Authentisierung im Netzwerk eingegangen. Dort wird im Abschnitt „Client-Server-Authentisierungssysteme“ eine Architektur beschrieben, mit Hilfe derer die Authentisierung im Netzwerk erfolgt. Darüber hinaus wird das Challenge-Response-Verfahren als häufig eingesetztes Protokoll zur Authentisierung und das Kerberos-Authentisierungssystem beschrieben. Kapitel 4 stellt das Windows 2000 Betriebssystem vor und erläutert die Funktionsweise der Anmeldearchitektur, um später als Basis einer Implementation zu dienen. Im Kapitel 5 werden mit dem, in den vorhergehenden Kapiteln erarbeiteten Basiswissen Konzepte zur Implementation erarbeitet. Ausgehend von einem Grobkonzept wird ein Modularisierungskonzept erstellt, in dem die Schnittstellen und Funktionen der Komponenten eines Iris-Biometrik Authentisierungssystems definiert sind. Im Anschluss daran werden Konzepte vorgestellt, die zeigen, wie das biometrische Authentisierungssystem in ein nichtbiometrisches implementiert werden kann. Kapitel 6 geht dann speziell auf die Implementierung der Konzepte in Windows 2000 Betriebssystem Umgebungen ein und stellt dar, wie die im Kapitel 5 definierten Module in die Komponenten der Windows 2000 Anmeldearchitektur eingebunden werden können um eine biometrische Authentisierung zu ermöglichen. Die Implementierungsmöglichkeiten werden diskutiert und Vor- und Nachteile aufgezeigt. Abschließend erfolgt in Kapitel 7 die Zusammenfasssung der Ergebnisse und ein Ausblick auf künftige Arbeiten.

Kapitel 2

Identitätsbestimmung in Computersystemen

Die Bestimmung der Identität von Personen ist ein bedeutender Vorgang in Computersystemen. Sicherheitsmechanismen wie Zugriffskontrolle und Überwachung basieren auf eindeutig bestimmten Identitäten und gewährleisten nur Schutz vor Angriffen, wenn es nicht möglich ist, unter falscher Identität ein Computersystem zu nutzen.

Dieses Kapitel stellt grundlegende Konzepte zur Identitätsbestimmung von Personen in Computersystemen dar.

Dazu wird in Abschnitt 2.1. die Authentisierung als eine Form der Identitätsbestimmung in Computersystemen erläutert und dargestellt. Darüber hinaus wird auf den Authentisierungsvorgang eingegangen und die einzelnen Phasen der Authentisierung beschrieben.

Danach erfolgt die Darstellung von speziellen Verfahren zur Identitätsbestimmung, wobei in 2.2. die Identitätsbestimmung mit Hilfe personenbezogener Merkmale und in 2.3. die Identitätsbestimmung mit Hilfe von biometrischen Merkmalen erläutert wird.

Der vierte Abschnitt 2.4 stellt die Iris-Biometrik als spezielles biometrisches Verfahren vor, bei dem die Identität einer Person anhand der Iris erkannt wird. Neben der Betrachtung der Iris als biometrisches Merkmal und der Einordnung der Iris-Biometrik als physiologisches biometrisches Verfahren wird weiterhin auf den Ablauf der Identitätsbestimmung mittels Iris-Biometrik eingegangen.

2.1 Authentisierung

Der Begriff Authentisierung bezeichnet den Vorgang, bei dem eine angenommene (oder behauptete) Identität festgestellt und verifiziert wird [Gollmann 1999, S.19].

Das Festellen der Identität (*Identifikation*) ist das Erkennen einer Person anhand eines *Identifikationsmerkmals*, z.B. Benutzerkennung. Die Benutzerkennung[1] ist eine leicht merkbare eindeutige Zeichenkette, mit der sich eine Person bei der Anmeldung am Computersystem zu erkennen gibt.

Die Identifikation anhand eines Identifikationsmerkmals wie der Benutzerkennung stellt jedoch nicht sicher, dass es sich bei der erkannten Identität tatsächlich um die zu einer Person dazugehörigen Identität handelt, da das Identifikaftionsmerkmal ein öffentliches Merkmal ist. Personen können mit dem Wissen oder dem Besitz des öffentlichen Merkmals verschiedene Identitäten annehmen. Darum muss die anhand des Identifikationsmerkmals festgestellte Identität zusätzlich verifiziert (geprüft) werden.

Das Prüfen der Identität (*Verifikation*) erfolgt ebenfalls anhand von Merkmalen (*Authentisierungsmerkmale*), die jedoch bei den heutigen Verfahren im Gegensatz zu Identifikationsmerkmalen geheim sind.

Nach erfolgreicher Verifikation wird der sich am Computersystem anmeldenden Person ein *Berechtigungsnachweis* (Credentials) zur Nutzung des Computersystems ausgestellt. Der Berechtigungsnachweis enthält einen eindeutigen benutzerspezifischen *Identifikator*, der im Computersystem die Identität des Benutzers kennzeichnet und ein bestimmtes Format aufweist[2].

Beispiele für Identifikatoren sind die Benutzer-IDs (User-ID, UID) in UNIX-Betriebssystemen und die Sicherheitsidentifikatoren (Security Identifier, SID) in den Betriebssystemen der Windows NT-Betriebssystemfamilie[3].

Nach der Anmeldung am Computersystem wird eine Person computersystemintern dann anhand des, während der Anmeldung zugewiesenen und im Berechti-

1 Für Benutzerkennung wird auch oft synonym der Begriff Benutzername verwendet.

2 Identifikatoren sind beispielsweise in den Zugriffskontrolllisten des Sicherheitssystems enthalten, in denen die Zugriffsrechte von Benutzern auf eine Betriebssystemressource definiert sind.

3 Zu den Betriebssystemen der Windows NT Betriebssystemfamilie gehören alle Windows NT Betriebssysteme, sowie die Windows 2000 und Windows XP Betriebssysteme.

gungsnachweis enthaltenen, Identifikators erkannt. Die Authentisierung als Phase des Feststellens und Verifizierens einer Identität stellt die Grundlage zur Gewährleistung der Sicherheit in Computersystemen dar und kann als vierphasiger Vorgang, der aus Datenaufnahme, Vorverarbeitung, Identifikation und Verifikation besteht, betrachtet werden.

2.1.1 Datenaufnahme

Während der *Datenaufnahme* werden die zur Identitätsbestimmung notwendigen Daten (Identifikations- und Authentisierungsmerkmal) erfasst. Die Erfassung findet dabei mit Hilfe von Dialogen statt, die die Rahmenbedingungen zur sicheren Erfassung zur Verfügung stellen.

Dazu gehört die Bereitstellung von Eingabefeldern, deren Inhalt verschleiert dargestellt wird, sodass der Inhalt bei der Eingabe nicht von anderen Personen mitgelesen werden kann. Darüber hinaus sollten die Dialoge zur Datenaufnahme sicherstellen, dass andere Programme nicht auf die Elemente wie Eingabefelder und Buttons zugreifen können, um das unrechtmäßige Verfolgen des Datenaufnahmevorgangs und das damit verbundene Kopieren der Identifikations- und Authentisierungsmerkmale zu verhindern.

2.1.2 Vorverarbeitung

In der *Vorverarbeitungsphase* werden die während der Datenaufnahme erfassten Daten zur Weiterverarbeitung aufbereitet. Hier findet z.B. eine Verschleierung von Kennworten mit kryptografischen Verfahren oder die Verschlüsselung aller erfassten Daten statt. Die Verfahren, die während der Vorverarbeitungsphase zur Anwendung kommen, sind stark von dem verwendeten Verfahren zur Identitätsbestimmung abhängig.

2.1.3 Identifikation

In der Identifikationsphase erfolgt das Feststellen der Identität anhand eines eindeutigen *Identifikationsmerkmals*. Als Identifikationsmerkmal dient dabei in den heutigen Betriebssystemen die Benutzerkennung, mit der sich ein Benutzer gegenüber dem Computersystem zu erkennen gibt.

Es findet ein eins zu n Vergleich des erfassten Identifikationsmerkmals mit den in einer Benutzerkontendatenbank gespeicherten Identifikationsmerkmalen der registrierten Benutzer statt, um festzustellen, ob der Benutzer im Computersystem registriert ist oder nicht.

2.1.4 Verifikation

In einer weiteren Phase, der Verifikation findet mit Hilfe eines bestimmten Verfahrens die Prüfung der während der Identifikation festgestellten Identität statt. Die Überprüfung geschieht wieder anhand eines Merkmals (*Authentisierungsmerkmal*), welches die mit der Benutzerkennung bzw. dem Identifikationsmerkmal vorher kenntlichgemachte Identität einer Person bezeugt.

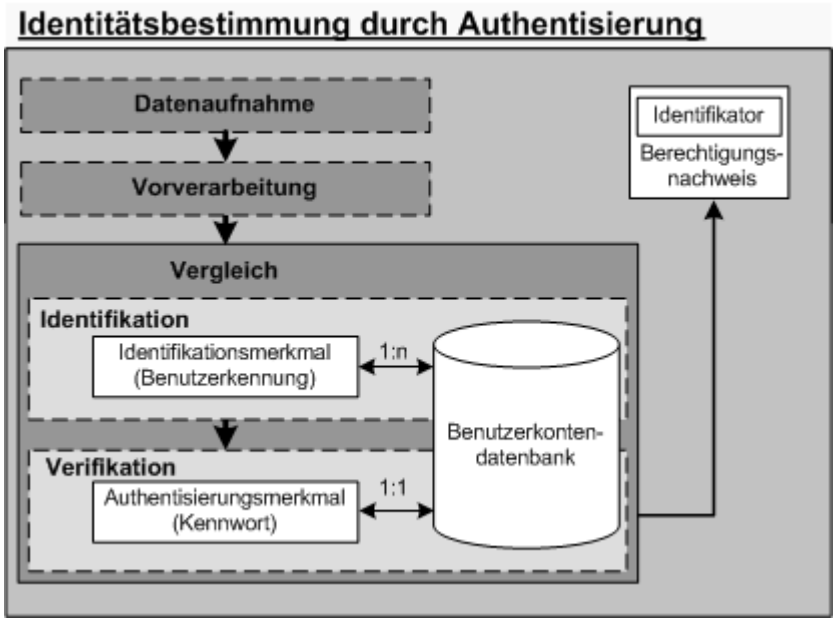

Abbildung 1: Phasen der Identitätsbestimmung durch Authentisierung

Im Gegensatz zur Identifikation findet bei der Authentisierung ein eins zu eins Vergleich statt, bei dem das in der Datenaufnahme erfasste und während der Vorverarbeitung aufbereitete Authentisierungsmerkmal mit dem zum Benutzer zugehörigen gespeicherten Authentisierungsmerkmal verglichen wird.

2.2 Identitätsbestimmung mit personenbezogenen Merkmalen

Das Feststellen und Prüfen der Identität von Personen in Computersystemen geschieht anhand von Merkmalen (Identifikationsmerkmale und Authentisierungsmerkmale). Die heute verwendeten Verfahren zur Identitätsbestimmung benutzen überwiegend personenbezogene Merkmale wie Wissen oder Besitz, die Personen beliebig zugeordnet werden können. In den folgenden Abschnitten wird daher die Identitätsbestimmung durch Wissen und Besitz kurz erläutert.

2.2.1 Identitätsbestimmung durch Wissen

Die Bestimmung der Identität durch Wissen basiert auf der Kenntnis eines gemeinsamen Geheimnisses zwischen Computersystem und Benutzer, wobei die Identität als nachgewiesen gilt, falls ein Benutzer das gemeinsame Geheimnis kennt und es vom Computersystem erfolgreich verifiziert wurde.

Technisch gestaltet sich dieser Vorgang folgendermaßen: Das gemeinsame Geheimnis wird in Form eines Kennworts mit Hilfe eines Einweg-Verfahrens verschlüsselt und zusammen mit einer Benutzerkennung in einer Benutzerkontendatenbank gespeichert. Kommt es zu einer Identitätsbestimmung, so wird das vom Benutzer angegebene Kennwort ebenfalls mit dem unumkehrbaren Verfahren verschlüsselt und während einer Verifikationssphase mit dem gespeicherten verglichen. Stimmen die verschlüsselten Kennworte überein, so gilt die Identität eines Benutzers als nachgewiesen. Aufgrund der einfachen Implementierung und Handhabung ist dieses Verfahren in Computersystemen weit verbreitet und wird heute hauptsächlich verwendet, um die Identität von Benutzern zu bestimmen.

Die Sicherheit dieses Verfahrens hängt allerdings stark von der Disziplin der Benutzer ab, die nur bedingt beeinflusst und kontrolliert werden kann. So stellt die Wahl des Kennworts eine große Schwachstelle des Verfahrens dar [Morris 1979, S.596][4]. Weiterhin kann die Geheimhaltung eines Kennworts oft nicht gewährleistet werden, da die Kennworte wegen Vergesslichkeit aufgeschrieben werden und die Verwahrung des aufgeschriebenen Kennworts nicht mit verantwortungsvoller Sorgfalt erfolgt. Die Sicherheit der Identitätsbestimmung wird praktisch aufgehoben, wenn die aufgeschriebenen Kennworte sich auf den Unter-

4 Morris und Thomson haben schon 1979 mit Angriffen auf gespeicherte Kennworte festgestellt, dass eine große Anzahl von Nutzern unsichere Kennworte zur Authentisierung wählen. Ein Bericht über die damaligen Angriffe befindet sich in [Morris 1979, S.594].

seiten der Tastaturen befinden oder sofort ersichtlich am Monitor befestigt wurden. Darüber hinaus können Kennworte bekannt werden, ohne dass der Besitzer es merkt, indem die verschlüsselten Kennworte aus der Benutzerkontendatenbank kopiert werden und mit Hilfe von Brute Force oder Wörterbuchangriffen geknackt werden.

2.2.2 Identitätsbestimmung durch Besitz

Eine Alternative oder Ergänzung zur Identitätsbestimmung durch Wissen stellt die Identitätsbestimmung durch Besitz dar, bei der mit Hilfe eines Gegenstands, dem *Token* (Abbildung 2), die Identität einer Person bestimmt wird. Das Token kann dabei in verschiedenen Formen wie Schlüssel, Magnetkarte oder Chipkarte[5] auftreten.

In der Regel werden diese Verfahren mit den Kennwortverfahren oder persönlichen Identifikationsnummern (PINs), also Wissen kombiniert, da ein Diebstahl oder das Duplizieren des Tokens nicht ausgeschlossen werden kann.

Während hierbei zur Identifikation das Token benutzt wird, erfolgt die Verifikation wieder mit Hilfe von Wissen. Durch die Kombination beider Verfahren wird eine höhere Sicherheit bei der Identitätsbestimmung gewährleistet, da ein Diebstahl des Tokens feststellbar ist und die Identitätsbestimmung mit Hilfe des Tokens und des Kennworts erfolgt.

Abbildung 2: Token - a) USB Schlüssel b) Chipkarte

Das unbemerkte Duplizieren eines Tokens und das Knacken des zum Token gehörenden Kennworts kann aber nicht ausgeschlossen werden, auch wenn es unter Umständen mit einem erheblichen technischen Aufwand verbunden ist.

5 Eine Einführung in die Chipkartentechnologie wird in [Rankl 1999] gegeben.

2.3 Biometrische Identitätsbestimmung

Personenbezogene Merkmale zur Identitätsbestimmung wie Wissen und Besitz können verloren, gestohlen, vergessen und imitiert werden. Darüber hinaus garantieren sie nur, dass ein Benutzer zu einem entsprechenden Zeitpunkt im Besitz eines Identifikations- und Authentisierungmerkmals ist.

Der berechtigte Besitz kann mit diesen Merkmalen jedoch nicht geprüft werden [Wirtz 1999, S.129]. Die Merkmale zur Bestimmung der Identität können auf andere Personen übertragen werden und sind nicht fest an eine bestimmte Person gebunden.

Darum ist es zweckmäßig personengebundene Merkmale zur Identitätsbestimmung einzusetzen, um sicherzustellen, dass die entsprechende Person auch tatsächlich zum Zeitpunkt der Identitätsbestimmung anwesend ist. Hierzu eignen sich die biometrischen Merkmale von Lebewesen, die in biometrischen Verfahren zur Identitätsbestimmung genutzt werden.

Ein *biometrisches Merkmal* ist eine messbare physiologische oder verhaltensbasierte Charakteristik eines Lebewesens [Zhang 2000, S.2]. Neben der Personengebundenheit können einige physiologische oder verhaltensbasierte Merkmale ebenfalls für eine Lebenderkennung verwendet werden.

Mit einer Lebenderkennung ist es möglich festzustellen, ob eine Person tatsächlich physisch anwesend und lebendig ist. Die Prüfung kann hier durch die Messung der Körpertemperatur oder das Aussenden zufälliger Reizungen und der Beobachtung der Reaktion auf diese geschehen. Wird beispielsweise die Iris für eine Lebenderkennung verwendet, so kann durch eine Veränderung der Lichtverhältnisse eine Änderung der Pupillengröße beobachtet werden. Das zufällige Aussenden von Lichtreizen und die damit verbundene Reaktion beweisen die Anwesenheit der Person.

2.3.1 Eigenschaften biometrischer Merkmale

Biometrische Merkmale müssen bestimmte Eigenschaften aufweisen, damit sie zuverlässig zur Identitätsbestimmung eingesetzt werden können. Die einzigartige Charakteristik eines bestimmten Merkmals reicht nicht aus, um sich als Merkmal zur Identitätsbestimmung zu eignen.

Biometrische Merkmale sollten folgende Eigenschaften aufweisen, damit sie zur technischen Identitätsbestimmung eingesetzt werden können:

Universalität, Verbreitung: Das biometrische Merkmal sollte bei möglichst vielen Person vorhanden sein [Behrens Roth 2000, S.327], wobei nach [Laßmann 1999, S.5] zu beachten ist, dass es kleine Bevölkerungsgruppen geben kann, die gewisse Merkmale nicht aufweisen oder durch eine körperliche Schädigung nicht in der Lage sind ihre Identität mit Hilfe dieses biometrischen Merkmals nachzuweisen. Beispielsweise ist die Identitätsbestimmung mit Hilfe der Iris für Blinde unter Umständen ungeeignet, da sie nicht in der Lage sind, ihr Auge vor der Kamera richtig zu positionieren.

Einzigartigkeit: Das biometrische Merkmal muss einzigartig in dem Sinne sein, dass es für verschiedene Menschen hinreichend unterschiedlich ausgeprägt ist [Laßmann 1999, S.5].

Konstanz: Das biometrische Merkmal sollte sich während der gesamten Lebensdauer möglichst wenig ändern [Laßmann 1999, S.5].

Erfassbarkeit: Das biometrische Merkmal muss mit der vorhandenen Technik quantitativ erfassbar sein [Behrens Roth 2000, S.327].

Biometrische Merkmale erfüllen diese Eigenschaften in unterschiedlicher Güte, sodass die biometrischen Verfahren, die mit dem Merkmal verbundenen Schwächen gegebenenfalls ausgleichen müssen. Ändert sich beispielsweise ein biometrisches Merkmal während der Lebensdauer eines Individuums geringfügig, so muss das biometrische Verfahren diese Änderungen stets aufzeichnen, um eine permanente Erkennung zu gewährleisten.

Andere Merkmale wie die Muster einer Iris sind zwar erfassbar, aber nicht sofort für die Identitätsbestimmung brauchbar und müssen innerhalb einer Normalisierung mit Hilfe von mathematischen Verfahren aufbereitet werden, um zur Extraktion von Merkmalen geeignet zu sein.

2.3.2 Ablauf einer biometrischen Identitätsbestimmung

Unabhängig vom genutzten biometrischen Merkmal ist den biometrischen Verfahren ein allgemeiner Ablauf gemein, der in [Petzel 1997], [Wirtz 1999, S.130], [Zhang 2000, S.8] und [Brömme et al 2001-2] beschrieben wird. Grundlegend

stimmen die Abläufe alle miteinander überein, unterscheiden sich jedoch im Abstraktionsgrad voneinander. Nachfolgend wird der Ablauf eines biometrischen Verfahrens basierend auf [Zhang 2000] dargestellt.

Enrollment
Jeder biometrischen Identitätsbestimmung geht ein *Enrollment* voraus, bei dem die biometrischen Merkmale einer Person erfasst, verarbeitet und als biometrisches Referenztemplate in einer Datenbank abgelegt werden. Danach kann eine Identitätsbestimmung erfolgen.

Identitätsbestimmungsvorgang
Zhang teilt den Ablauf einer biometrischen Identitätsbestimmung in drei Phasen ein: Datenaufnahme, Merkmalsextraktion, Vergleich.

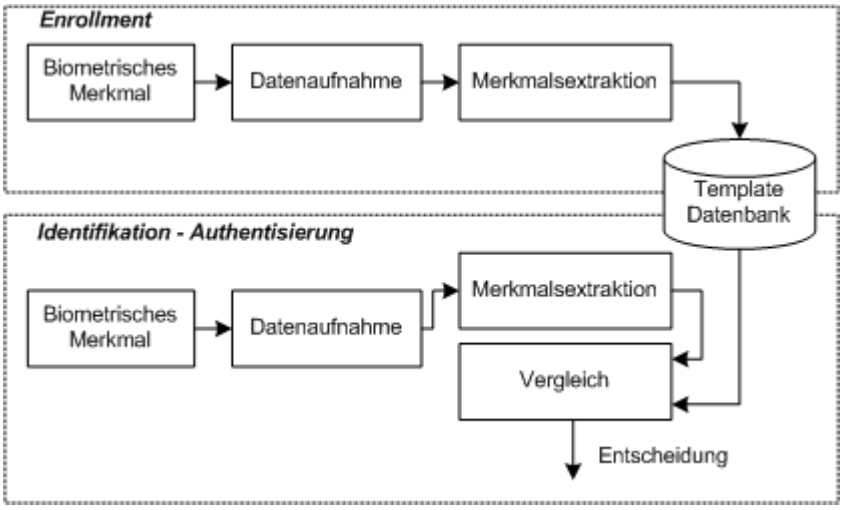

Abbildung 3: Ablauf einer biometrischen Identitätsbestimmung basierend auf [Zhang 2000]

Datenaufnahme: Während der Datenaufnahme werden die biometrischen Merkmale einer Person mit Hilfe eines Sensors erfasst und digitalisiert. Der Sensor kann dabei, je nach Verfahren, eine Kamera, ein Mikrofon, eine Tastatur oder ein spezielles Gerät (z.B. Fingerabdruckscanner) sein, das zur Erfassung der biometrischen Merkmale notwendig ist.

Merkmalsextraktion: Die während der Datenaufnahme digitalisierten biometrischen Merkmale (biometrisches Sample) werden während der Merkmalsextraktion extrahiert, um ein biometrisches Template zu erzeugen. Das Template stellt dabei eine Kenngröße des biometrischen Merkmals einer

Person dar und wird später zur Erkennung einer Identität genutzt. Die Templateerzeugung kann nach zwei unterschiedlichen Ansätzen erfolgen. Im ersten Ansatz werden die biometrischen Merkmale sofort aus den biometrischen Daten extrahiert, um daraus ein Template zu erstellen. Im zweiten Ansatz kann nicht sofort ein Merkmal aus den digitalen biometrischen Daten extrahiert werden, sodass die digitalen Daten aufbereitet werden müssen[6]. Dazu werden sie mit Hilfe von Transformationsverfahren wie Fourier- oder Wavelettransformation transformiert, damit Merkmalsausprägungen gemessen werden können und ein biometrisches Template erstellt werden kann.

Vergleich: Anschließend findet ein Vergleich des während der Merkmalsextraktion erzeugten Template mit den gespeicherten Referenztemplates statt. Dies kann nicht wie bei Kennwortverfahren durch einen einfachen Vergleich erfolgen, da die erzeugten Templates durch die unterschiedliche Positionierung des biometrischen Merkmals vor einem Sensor und die unterschiedlichen optischen Verhältnisse nicht exakt mit dem Referenztemplate übereinstimmen. Darum kommen hier Schwellwertverfahren zum Einsatz, die eine Toleranzbreite zum Vergleich einbeziehen müssen, um zu entscheiden, ob das erzeugte biometrische Template im Näherungsbereich des Referenztemplate liegt. Die Wahl einer geeigneten Toleranzbreite muss für jedes biometrische Verfahren speziell bestimmt werden und ist nicht unproblematisch. Eine hohe Toleranzbreite erhöht die Wahrscheinlichkeit, dass unberechtigte Personen als berechtigt anerkannt werden. Dieser Fehlertyp, auch *False Acceptance Rate* (FAR) genannt, ist ein maßgebliches Qualitätsmerkmal für biometrische Verfahren, sagt aber allein nicht genug aus, um das biometrische Verfahren ausreichend beurteilen zu können. Hierzu bedarf es einer weiteren Fehlerrate, der *False Rejection Rate* (FRR), die angibt, wie oft eine berechtigte Person nicht erkannt wird. Weiterhin ist die Art und Weise wie das erzeugte Template mit den Referenztemplates verglichen wird abhängig davon, ob der Vergleich eine biometrische Identifikation oder eine biometrische Verifikation zum Ziel hat. Erfolgt eine reine biometrische Identifikation, so wird das erzeugte Template mit allen in der Template-Datenbank gespeicherten Referenztemplates verglichen (1 zu n Vergleich) um die Identität einer Person zu bestimmen. Wird das biometrische Verfahren zur Authentisierung einer Identität (also zur Verifikation) eingesetzt, so wird das erzeugte Template mit dem gespeicherten Template der zu authentisierenden Person verglichen (1 zu 1 Vergleich).

6 [Zhang 2000] integriert hier die Vorverarbeitungsphase in die Phase der Merkmalsextraktion

Im Rahmen der *Biometric Research Group* des Arbeitsbereichs Anwendungen der Informatik in Geistes- und Naturwissenschaften des Fachbereichs Informatik der Universität Hamburg wurde der allgemeine Ablauf eines biometrischen Verfahrens untersucht um zu bestimmen, wie man biometrische Verfahren unter reallen Bedingungen testen kann. FAR und FRR können zwar etwas über die Qualität eines biometrischen Verfahrens aussagen, jedoch nicht über die Ursache, die zu diesem Ergebnis geführt hat. Darum ist es wichtig, die Arbeitsweise des biometrischen Systems zu überwachen und die in den einzelnen Phasen erzeugten Daten aufzuzeichnen. Dabei wurde festgestellt, dass der Ablauf nach [Zhang 2000] sich zwar gut eignet, um die einzelnen Module eines biometrischen Systems abstrakt darzustellen, aber nicht differenziert genug ist, um Daten über die Arbeitsweise des biometrischen Systems zu sammeln und eventuelle Schwächen des Systems herauszufinden. Darum wurde in [Brömme et al 2001-2] der abstrakte Ablauf um mehrere Phasen ergänzt, um den genauen Ablauf des Verfahrens unter realen Bedingungen beobachten und auswerten zu können. Die Innovation war hierbei die Defintion einer allgemeinen Vorverarbeitungs- und Quallitätskontrollphase, die in biometrischen Verfahren wiederzufinden sind.

Vorverarbeitung: Die Vorverarbeitungsphase schließt sich dabei direkt der Datenaufnahme an und dient zur Untersuchung der digitalen Daten nach bestimmten Merkmalen, die zur Weiterverarbeitung und insbesondere zur Normalisierung benötigt werden. Andere Autoren wie [Wirtz 1999, S.132] betrachten ebenfalls eine Vorverarbeitungsphase, benutzen diese aber nur zur Normalisierung der digitalen biometrischen Daten. Die Normalisierung findet in [Brömme et al 2001-2] aber erst in einer späteren Phase statt.

Qualitätskontrolle: Die Phase der Qualitätskontrolle schließt sich der Vorverarbeitungsphase an und prüft mit Hilfe der während der Vorverarbeitung ermittelten Daten, ob die digitalen biometrischen Daten zur Weiterverarbeitung geeignet sind. Ist das der Fall, so findet je nach Verfahren eine Normalisierung der Daten statt, um eine Extraktion von Merkmalen zu ermöglichen.

Diese Phasen sollten beachtet werden, um die erzeugten Daten eines biometrischen Verfahrens ausreichend aufzuzeichnen. Es ist jedoch nicht immer notwendig, sie auch als Module eines biometrischen Systems wie in [Broemme et al 2001-2] zu betrachten, da sie mit der Merkmalsextraktion in einem Modul gekapselt sein können.

2.3.3 Physiologische biometrische Verfahren

Biometrische Verfahren können, je nachdem welche biometrischen Merkmale sie verwenden, in physiologische und verhaltensbasierte Verfahren unterschieden werden.

Physiologische biometrische Verfahren nutzen nach [Wirtz 1999, S.130] physiologische biometrische Merkmale, die sich während des gesamten Lebens nicht oder nur geringfügig ändern, wie Gesichtsmerkmale, Fingerabdrücke, Handgeometrie oder die Muster der Iris eines Auges. Aufgrund der Merkmalskonstanz werden diese Merkmale auch statische Merkmale [Bleumer 1999, S.156] und die damit verbundenen Verfahren statische Verfahren [Probst 2000, S.323] genannt.

Diese besitzen den Vorteil, dass über Häufigkeitsverteilungen der zugrundeliegenden statischen physiologischen Merkmale die erzielbaren Erkennungsgenauigkeiten theoretisch abgeschätzt werden können [Wirtz 1999, S.130]. Durch unterschiedliche Positionierung des biometrischen Merkmals an der Mensch-Sensor-Schnittstelle können jedoch Variationen entstehen, die eine theoretische Erkennungsrate stark von der tatsächlichen abweichen lässt. Die theoretisch ermittelte Erkennungsgenauigkeit ist somit nur bedingt aussagekräftig, da sie ideale Bedingungen wie die wiederholte, exakt gleiche Positionierung des biometrischen Merkmals vorraussetzt.

Ein Nachteil physiologischer Verfahren ist, dass statische biometrische Merkmale nicht beliebig oft ausgetauscht werden können [Tönnesen 1999, S.161]. Gelingt es einem Angreifer durch das Imitieren eines biometrischen Merkmals das System zur Identitätsbestimmung zu überlisten, so kann er die Identität anderer Personen annehmen.

In Systemen, die Merkmale wie Wissen oder Besitz zur Identitätsbestimmung benutzen, können diese beliebig oft ausgetauscht werden, sodass nach einem bemerkten Angriff durch Wechsel von Passwörtern oder Chipkarten ein Vortäuschen einer Identität mit Hilfe eines gestohlenen Merkmals zukünftig verhindert werden kann.

Physiologische biometrische Merkmale sind jedoch nicht unbegrenzt vorhanden, sodass die Möglichkeit des Austauschs durch ein physiologisches biometrisches Merkmal der gleichen Klasse begrenzt ist. So besitzt der Mensch nur zehn Finger und zwei Augen. Noch problematischer stellt sich die Situation dar, wenn zur Identitätsbestimmung das Gesicht benutzt wird, da es beim Gesicht keine Ausweichmöglichkeit gibt.

2.3.4 Verhaltensbasierte biometrische Verfahren

Eine weitere Klasse biometrischer Verfahren sind die verhaltensbasierten biometrischen Verfahren, die auf der Untersuchung verhaltensbasierter Merkmale, wie die Tastaturanschlagsdynamik, Unterschriftsdynamik oder Sprachproben basieren. Diese werden nach [Wirtz 1999, S.130] auch als personencharakteristische Anteile menschlicher Aktionen bezeichnet und unterliegen natürlichen Variationen, sodass die erfassten Charakteristika nie komplett identisch sein können.

Aufgrund der dynamischen Eigenschaft verhaltensbasierter Merkmale wird im Zusammenhang mit verhaltensbasierten biometrischen Verfahren auch von dynamischen Verfahren gesprochen. Dennoch enthalten verhaltensbasierte Verfahren immer eine physische Komponente. So werden die Sprachproben einer Person zu einem bestimmten Anteil durch die Struktur des Mund- und Rachenraums geprägt [Colsman 2001, S.10]. Die Unterschriftsbewegung ist größtenteils durch das motorische System der jeweiligen Person beeinflusst.

Probst differenziert in [Probst 2000, S. 323] die verhaltensbasierten dynamischen Verfahren noch weiter als Wirtz, wobei er sie in dynamisch physiologische und dynamisch wissensbasierte Verfahren unterteilt. Dabei sind dynamische physiologische Verfahren diejenigen, die nur durch physiologische Komponenten mitbestimmt werden. Dynamisch wissensbasierte biometrische Verfahren kombinieren personenbezogenes Wissen mit einem biometrischen Verfahren. Die Erkennung einer Lippenbewegung oder einer Stimme kann mit einem Kennwort verbunden sein, das während der Identitätsbestimmung angegeben werden muss. Das biometrische Muster bildet sich nicht nur durch eine individuelle Sprechweise, sondern auch durch das gesprochene Wort, wodurch sich mit dem Wechsel des Wortes eine fast unbegrenzte Anzahl unterschiedlicher biometrischer Templates erzeugt werden können, die für den Sprechenden charakteristisch sind [Tönnesen 1999, S.161].

Dynamisch wissensbasierte biometrische Verfahren besitzen nach Tönnesen gegenüber den physiologischen Verfahren Vorteile. Nach einem erfolgreichen Angriff auf das System zur Identitätsbestimmung kann durch die Kombination mit Wissen ein neues, personenspezifisches, biometrisches Template aus der gleichen biometrischen Merkmalsklasse erzeugt werden. Der Merkmalsträger kann jederzeit selbst das biometrische Referenztemplate ändern und drückt durch das Nennen des Kennworts im Identitätsbestimmungsvorgang eine Willenserklärung aus. Im Fall einer Erpressung kann der Merkmalsträger mit einem speziellen Kennwort Alarm auslösen und das Computersystem so vom Erpressungsversuch informieren.

2.4 Iris-Biometrik

Unter Iris-Biometrik werden im Folgenden biometrische Verfahren verstanden, die zur Bestimmung der Identität einer Person die Muster der Iris benutzen. Die Irismuster besitzen zwischen verschiedenen Personen eine sehr große Mustervariablilität und eignen sich daher hervorragend zur Identitätsbestimmung einer Person, da jede Iris eine eigene spezielle Charakteristik aufweist, deren wiederholtes Vorhandensein bei einer anderen Person vernachlässigbar ist.

Geeignete Verfahren zur Abbildung und Erkennung einer Iris in Computersystemen wurden von John Daugman 1993 vorgeschlagen und 1994 patentiert [Daugman 1994]. Die Algorithmen wurden in Form von ausführbaren Programmen auf den Markt gebracht und bilden seitdem die Grundlage für Iriserkennungssysteme, die für Versuche in der Öffentlichkeit eingesetzt werden.

Projekte, bei denen der Daugman Algorithmus eingesetzt wurde, fanden bei der British Telecom, US Sandia Labs, UK International Physical Laboratory, NCR, Oki, IriScan, Iridian, Sensar und Sarnoff statt.

Dabei wurde festgestellt, dass die Fehlübereinstimmungsquote bei der Identitätsbestimmung mit Hilfe der Iris gleich Null ist. Dies ist als ein Erfolg zu werten, da andere biometrische Verfahren wie z.B. die Gesichtserkennung nach kurzer Zeit Fehlerquoten von 43% [Daugman 1998, S.2 zitiert nach [Phillips et al 2000]] bis 50 % [Daugman 1998, S.2 zitiert nach [Pentlant et al 2000]] aufweisen und deshalb als Verfahren zur Identitätsbestimmung nur bedingt geeignet sind .

Im Folgenden wird eine kurze Einführung in die Iris-Biometrik als biometrisches Verfahren gegeben, die als Wissensgrundlage für die spätere Konzeption der Implementierungsmöglichkeiten in Computersysteme dienen soll. Dazu wird zuerst die Iris als biometrisches Merkmal dargestellt und anschließend die Phasen des Ablaufs einer biometrischen Identitätsbestimmung mit Hilfe der Iris-Biometrik näher erläutert.

2.4.1 Die Iris als biometrisches Merkmal

Die Iris, auch Regenbogenhaut genannt, umgibt die Pupille und ist ein sichtbarer Bestandteil des Auges (siehe Abbildung 4). Ihre Entwicklung beginnt ab dem dritten Schwangerschaftsmonat und ist bis zum 8. Monat mit ihren Strukturen größtenteils abgeschlossen [Daugman 1998, S.2 zitiert nach [Kronfeld 1962]],

obwohl sich die Pigmentierung in den ersten Lebensjahren noch ändern kann. Die Gestalt der Iris wird dabei nicht nur genetisch, sondern auch von Umwelteinflüssen bestimmt, sodass es sich bei der Iris um ein phänotypisches Merkmal handelt, das selbst bei eineiigen Zwillingen eine unterschiedliche Charakteristik aufweist und somit die Eigenschaft der Einzigartigkeit erfüllt. Nach abgeschlossener Entwicklung ändert sich das Aussehen der Iris nicht mehr und bleibt während des gesamten Lebens konstant.

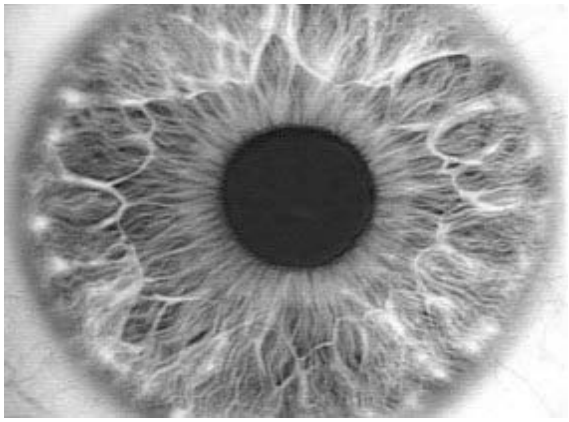

Abbildung 4: Iris eines Menschen

Vollständig ausgebildet besitzt jede Iris eine bestimmte Farbe, die von einer Anzahl von Melanin Pigmenten bestimmt wird, wobei eine blaue Iris das Resultat fehlender Pigmente ist und eine hohe Pigmentanzahl eine dunkle Augenfarbe zur Folge hat [Daugman 1998, S.2 zitiert nach [Chedekel 1994].

Weiterhin besitzt jede Iris bestimmte Muster, die ihr eine einzigartige Charakteristik verleihen und sie zur Identitätsbestimmung so interessant machen. Diese können sich aus mehreren verschiedenen Formen zusammensetzen wie bogenförmige Ränder, Furchen, Stege, Gruften, Ringe, Kronen, Tüpfel und Zackenkragen.

Die Muster sind mit Hilfe von Sensoren in Form von Digitalkameras technisch erfassbar, sodass neben Universalität, Einzigartigkeit und Konstanz ebenfalls die Erfassbarkeit als Anforderung an ein biometrisches Merkmal erfüllt ist. Darüber hinaus besteht bei der Iris eine geringe Verletzungsgefahr, da sie durch die Linse und die Hornhaut geschützt wird und ein inneres Organ darstellt, das aber den-

noch sichtbar ist. Durch die Fähigkeit sich bei Heilligkeitsveränderungen zusammenziehen und ausdehnen zu können kann die Iris ebenfalls für eine Lebenderkennung[7] genutzt werden, bei der zufällige Änderungen der Helligkeit erzeugt werden und die Reaktion der Iris beobachtet wird.

2.4.2 Ablauf einer biometrischen Identitätsbestimmung mit Hilfe der Iris Biometrik

Der Ablauf einer biometrischen Identitätsbestimmung mit Hilfe der Iris-Biometrik kann grob in vier Phasen unterteilt werden: Datenaufnahme, Vorverarbeitung, Merkmalsextraktion und Vergleich.

2.4.3 Datenaufnahme

Während der Datenaufnahme wird mit Hilfe eines Sensors in Form einer Digitalkamera (CCD-Kamera) ein biometrisches Sample des Auges in Form eines hochaufgelösten digitalen Bildes erstellt. Um die Einzelheiten der Irismuster zu erfassen, sollte die Auflösung des Bildes so gewählt werden, dass der Iris-Radius mindestens 50 Pixel entspricht.

Abbildung 5: Digitalkamera zur Erfassung einer Iris (Panasonic Authenticam)

In den bisherigen Feldversuchen wurden zur Erfassung der Augenoberfläche monochrome CCD-Kameras mit einer Auflösung von 640 x 480 Pixel eingesetzt. Dadurch konnten Abbildungen der Augenoberfläche erzeugt werden, in denen die Iris einen Radius zwischen 100 und 140 Pixel aufwies [Daugman 2001, S.132]. Mittlerweile sind Kamerasysteme zur Erfassung der Iris in Handel erhältlich. Ein Beispiel dafür ist die Authenticam von Panasonic (Abbildung 5).

7 Der Begriff der Lebenderkennung wird in 2.2 erläutert.

2.4.4 Vorverarbeitung

Das während der Datenaufnahme erfasste biometrische Sample des Auges in Form eines digitalen Bildes enthält neben der Iris andere Bestandteile, die für den Erkennungsvorgang nicht von Bedeutung sind. Dazu gehören beispielsweise das Augenlid oder die Pupille, die zwar bei der Datenaufnahme erfasst, danach jedoch nicht mehr verwendet werden. Darüber hinaus kann die Größe und Qualität der Augenbilder des gleichen Auges bei jeder Datenaufnahme variieren, sodass eine Vorverarbeitung notwendig ist, um ein biometrisches Template erstellen zu können.

Die Vorverarbeitungsphase setzt sich aus drei Teilphasen zusammen und besteht aus: Lokalisierung der Iris, Normalisierung und Aufbereitung. (Abbildung 8, 9)

Lokalisierung der Iris im biometrischen Sample

Die Lokalisierung der Iris bezeichnet den Vorgang, bei dem versucht wird, die Iris im biometrischen Sample des Auges aufzufinden, damit später eine Analyse der Irismuster stattfinden kann.

Im Lokalisierungsvorgang werden dabei die innere (Pupillenrand) und die äußere Begrenzung der Iris (Hornhautrand) gesucht. Diese können abstrakt als Kreise dargestellt werden, die jedoch häufig nicht konzentrisch angeordnet sind.
So kann der Pupillenmittelpunkt näher zur Nase und tiefer als der Irismittelpunkt liegen. Der Pupillenradius kann einen Bereich von 0,1 bis 0,8 des Irisradius ausmachen. Die Begrenzungen der Iris werden im Daugman Verfahren mit Hilfe eines integrodifferentialen Operator gefunden (Abbildung 6).

$$max_{(r,x_0,y_0)}\left|G_\sigma(r) * \frac{\partial}{\partial_r}\oint_{r,x_0,y_0}\frac{I(x,y)}{2\pi r}ds\right|$$

Abbildung 6: integrodifferentialer Operator zum Finden der inneren und äußeren Begrenzung der Iris aus [Daugman 1998, S.2]

Der Operator sucht im Abbildungsbereich (x,y) nach dem Maximum bezüglich eines zunehmenden Radius r der unscharfen partiellen Ableitungen, für das normierte Konturenintegral von *I*(x,y) entlang eines Kreisbogens ds mit dem Radius r und den Mittelpunktkoordinaten (x_0, y_0) [Daugman 2001, S.132]. Das Symbol *G*

ist eine Glättungsfunktion (wie z.B. eine Gaußfunktion) und das Symbol * steht für eine Faltungsfunktion. Bei dem Suchen der Begrenzungen wird mit dem Suchen des Pupillenrandes begonnen, da zwischen Iris und Pupille ein größerer Kontrast auftritt als zwischen Iris und Hornhaut. Nachdem der Pupillenrand und der Hornhautrand gefunden wurden, werden die Begrenzungen der Augenlieder gesucht, die die Iris teilweise oben und unten bedecken.

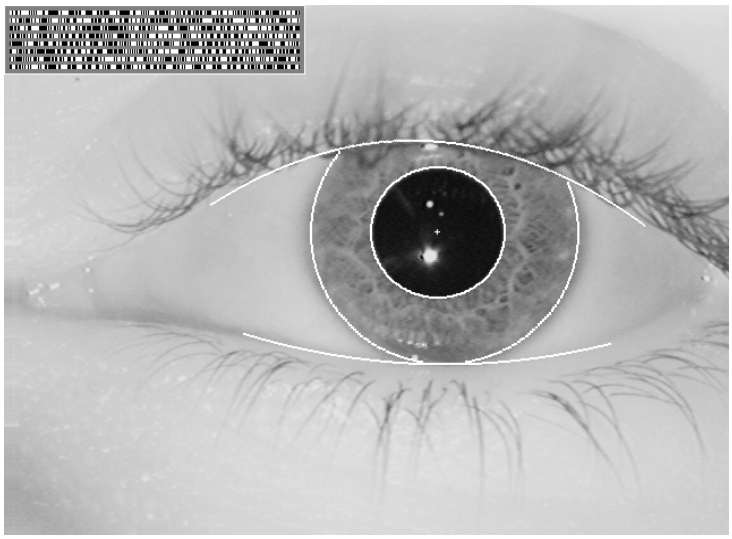

Abbildung 7: Abbild eines Auges mit den gefundenen Begrenzungen der Iris aus [Daugman 1998, S.1]

Das Auffinden der oberen und unteren Lidkante erfolgt ebenfalls mit Hilfe des integrodifferentialen Operators aus Abbildung 6, jedoch wird nun der Konturenintegrationsweg von kreisförmig auf bogenförmig umgestellt. Das Ergebnis dieser Verfahren ist die Begrenzung der Iris unter Ausschluss aller weiteren Bildbereiche des Auges (Abbildung 7).

Normalisierung der Irisdaten

Die Größe der Pupille kann durch unterschiedliche Beleuchtung variieren und hat eine Deformation (Stauchung oder Streckung) der Irismuster zur Folge. Um die Muster der Iris zu analysieren und gleiche Iriden erkennen zu können, ist eine Kompensation der Deformationen notwendig, die in der Normalisierungsphase

stattfindet. Dazu wird das Irisabbild in ein Polarkoordinatensystem[8] projiziert. Da der Radialkoordinatenbereich von der inneren Begrenzung bis zur äußeren Begrenzung der Iris ein Einheitsintervall ist ([0,1]), erfolgt hier automatisch eine Korrektur für die Musterverzerrung, die bei der Veränderung der Pupillengröße entsteht [Daugman 2001, S.145].

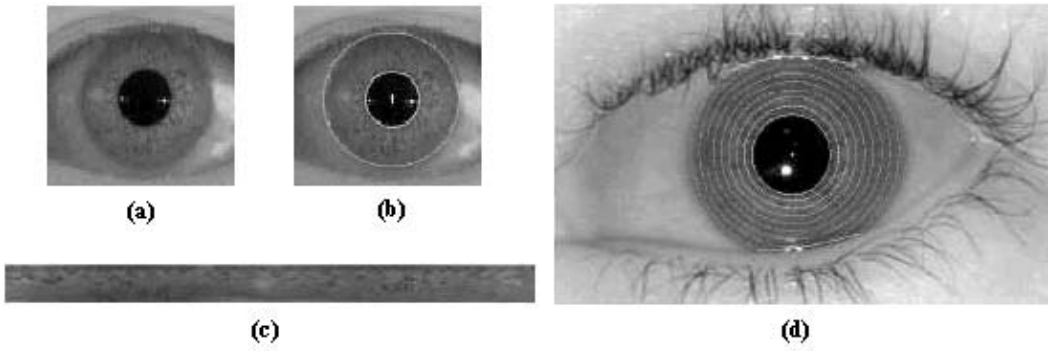

Abbildung 8: Vorverarbeitung: (a) Bild des Auges, (b) Bild nach Lokalisierung der Iris, (c) Iris nach Normalisierung, (d) Visualisierung des Polarkoordinatensystems in der Iris

Bild (a), (b), (c) aus [Yong et al 1999, S.2]

(d) aus [Johns 2002]

Aufbereitung

In der Phase der Aufbereitung kann mit mathematischen Verfahren eine Datenaufbereitung des normalisierten Irisabbilds erfolgen, um die Muster später besser extrahieren zu können. Yong et al beschreiben in [Yong et al 1999, S.2] eine Aufbereitungsphase, in der die Helligkeit und der Kontrast des Irisabbilds korrigiert werden.

Abbildung 9: Normalisiertes Irisabbild nach Aufbereitung aus [Yong et al 1999, S.2]

In den Dokumentationen zum Daugman Algorithmus wird allerdings von keiner Phase der Aufbereitung gesprochen. Durch die Glättungsfunktion während der Lokalisierungsphase findet aber auch eine Art Datenaufbereitung statt.

8 Das Polarkoordinatengitter muss dabei nicht konzentrisch sein, da die Pupille in den meisten Augen nicht zentriert in der Iris liegt. Hier ist eine Verschiebung bis zu 15 % zur Nase nicht ungewöhnlich.

2.4.5 Merkmalsextraktion

Während der Merkmalsextraktion werden Informationen über Irismuster aus dem normalisierten und aufbereiteten Irisabbild extrahiert und ein Iris-Template[9] erstellt. Die Extraktion erfolgt dabei im Daugman Algorithmus durch das sequenzielle Abtasten des normalisierten und aufbereiteten Irisabbilds mit Hilfe zweidimensionaler Quadratur-Gabor-Wavelets.

$$h_{\{Re,\,Im\}} = sgn_{\{Re,\,Im\}} \int_{\rho} \int_{\phi} I(\rho,\phi)\, e^{-i\omega(\theta_0-\phi)}\, e^{-(r_0-\rho)^2/\alpha^2}\, e^{-(\theta_0-\phi)^2/\beta^2}\, \rho\, d\rho\, d\phi$$

Abbildung 10: Formel zur Extraktion der Informationen über Irismuster mittels zweidimensionaler Quadratur-Gabor-Wavelets aus [Daugman 1998, S.3]

Hierbei kann $h_{\{Re,\,Im\}}$ als komplexwertiges Bit betrachtet werden, dessen reelle und imaginäre Komponente in Abhängigkeit des Vorzeichens des zweidimensionalen Integrals nur die Werte 0 oder 1 annehmen kann.

$I(\rho, \Phi)$ ist die Abbildung der Iris in einem Polarkoordinatensystem (normalisiertes Irisabbild), um dem Irisabbild die Eigenschaft der Maßstabs- und Translationsinvarianz zu verleihen.

Weiterhin sind α und β die mehrfach skalierten Größenparameter der zweidimensionalen Wavelets, die einen Bereich von 0,15 mm bis 1,2 mm auf der Iris überdecken. Die Waveletfrequenz wird mit ω dargestellt und erstreckt sich über 3 Oktaven invers proportional zu β.

(r_0, θ_0) sind die Polarkoordinaten, von den Regionen der Iris, für die die Phasen-Koordinaten $h_{\{Re,\,Im\}}$ berechnet werden. Insgesamt werden 2048 Phasenbits für jede Iris errechnet, sodass sich eine 256 Byte große Kenngröße für eine Iris ergibt [Daugman 2001, S.135].

Darüber hinaus werden gleich viele Maskierungsbits berechnet, um anzugeben, ob eine Irisregion von Augenlidern überdeckt wird, Augenbrauen enthält, Reflexionen aufweist, Grenzeffekte von Kontaktlinsen zeigt oder mit schlechtem Signal-Rausch-Verhältnis behaftet ist, sodass diese Bereiche im Iriscode als evidente Bereiche verworfen werden sollten.

9 Iriscode entspricht Iristemplate

2.4.6 Vergleich

Während der Vergleichsphase wird das in der Merkmalsextraktionsphase erzeugte Iristemplate mit einem Referenztemplate verglichen, um festzustellen, ob diese übereinstimmen. Als Maß für die Unterschiedlichkeit zwischen dem Iristemplate und einem Referenztemplate kommt dabei im Daugman Algorithmus der Hamming-Abstand zum Einsatz, der die Anzahl unterschiedlicher Bits zweier gleichlanger Binärwerte angibt.

Daugman verwendet hier jedoch eine normalisierte Form des Hamming-Abstands, der Bitmasken miteinbezieht, die bei der Iristemplateerstellung erzeugt wurden, um nur unverdeckte Stellen der Iris vergleichen zu können.

$$HD = \frac{\left\| (code\,A \otimes code\,B) \cap mask\,A \cap mask\,B \right\|}{\left\| mask\,A \cap mask\,B \right\|}$$

Abbildung 11: Normierter Hamming-Abstand zum Vergleich zweier Iristemplates aus [Daugman 1998, S.4]

Der XOR-Operator ermittelt hierbei nichtübereinstimmende Bitstellen. Der AND-Operator sorgt dafür, dass nur solche Bitstellen dem Korrelationsvergleich unterzogen werden, die nicht durch Augenbrauen, Augenlieder, Lichtreflexionen oder anderen Störungen ungültig sind.

Der normierte Hamming-Abstand entsteht aus dem Quotienten der Normen der resultierenden Bitvektoren und der in der AND-Logik kombinierten Maskenvektoren, wobei „code A“ und „code B“ die Phasenbitvektoren und „mask A“ und „mask B“ die Bitmaskenvektoren bezeichnen.

Der sich ergebene normierte Hamming-Abstand ist ein Maß für die Unähnlichkeit zweier Iristemplates. Ein Wert von 0 entspricht hierbei einer perfekten Übereinstimmung.

Kapitel 3

Authentisierung im Netzwerk

Die Vernetzung von Computern ermöglicht eine Authentisierung mit Hilfe von entfernten Rechnern, die Dienste zur Prüfung der Identität im Netzwerk zur Verfügung stellen. Der Datenaustausch zwischen den Computern findet mit Hilfe von Netzwerkprotokollen statt, die regeln wie die Daten zwischen den Kommunikationspartnern ausgetauscht werden.

Im Bereich der Netzwerkprotokolle und ihrer Implementationen sind Schwächen aufgezeigt worden [Morris 1985, Bellovin 1989, Jonchery 1995], sodass die Systeme zur Authentisierung im Netzwerk oft spezielle Protokolle zum Datenaustausch einsetzen.

Diese Systeme besitzen einen ähnlichen Aufbau und werden im Folgenden Client-Server-Authentisierungssysteme genannt.

Abschnitt 3.1. beschreibt die allgemeine Architektur von Client-Server-Authentisierungssystemen und stellt deren Aufbau und Funktionsweise dar. Dabei wird speziell auf das Problem der sicheren Datenübertragung eingegangen, um später einschätzen zu können, ob die eingesetzten Verfahren zur Authentisierung mit Hilfe biometrische Verfahren geeignet sind.

Abschnitt 3.2 stellt ein spezielles Client-Server-Authentisierungssystem vor. Die Wahl fiel dabei auf Kerberos als ein System, das durch die Unterstützung in Betriebssystemen wie Windows 2000 eine immer weitere Verbreitung erfährt. Dabei werden hauptsächlich technische Abläufe dargestellt, um im Zusammenhang mit biometrischen Verfahren zu einer Bewertung zu kommen.

3.1 Client-Server-Authentisierungssysteme

Client-Server-Authentisierungssysteme sind Systeme, die eine zentrale Authentisierung im Netzwerk zur Verfügung stellen und einen Zugang zu Computersystemen und Diensten ermöglichen. Sie sind sowohl in Verbänden von Netzwerkbetriebssystemen als auch im Internet wiederzufinden, wo sie jeweils eine Funktionseinheit zur Zugangskontrolle darstellen.

3.1.1 Architektur eines Client-Server-Authentisierungssystems

Client-Server-Authentisierungssysteme besitzen eine grundlegend identische Architektur, mit der sie allgemein dargestellt werden können.

Hauptkomponenten sind hierbei Clienten, die Daten zur Authentisierung erfassen, gegebenenfalls vorverarbeiten[10] und zentrale Server (Authentisierungsserver) auf denen Dienste zur Identifikation und Verifikation (Authentisierungsdienste) ausgeführt werden.

Die Kommunikation zwischen dem Client und dem Authentisierungsdienst erfolgt über ein Protokoll, das regelt, wie der Datenaustausch zwischen Client und Authentisierungsdienst stattfindet[11], um eine Authentisierung über das Netzwerk zu ermöglichen.

Nach erfolgreicher Authentisierung erhält der Client vom Authentisierungsserver einen Berechtigungsnachweis mit einem spezifischen Identifikator[12], der die Identität des Benutzers darstellt, und mit dem er ein Computersystem oder einen Dienst im Netzwerk nutzen kann.

Zur Identifikation und Authentisierung benutzt der Authentisierungsdienst eine zentrale Benutzerkontendatenbank, wodurch eine Verwaltung von Benutzerkonten sinnvoll geschehen kann. Bei einer Dezentralisierung der Benutzerkonten

10 Die Vorverarbeitung auf dem Clienten ist systemabhängig. In vielen Client-Server-Authentisierungssystemen findet nach der Datenerfassung keine Vorverarbeitung statt. Beispiele hierfür sind häufig Dienste, die im Internet zur Verfügung gestellt werden. Dort werden die erfassten Merkmale zur Authentisierung ohne Vorverarbeitung im Klartext übertragen.

11 Hierbei werden oft spezielle Protokolle wie das Challenge-Response-Protokoll oder das Kerberos-Protokoll benutzt.

12 Der Begriff des Identifikators wird im Kapitel 2 definiert.

wären auf mehreren Computersystemen Benutzerkontendatenbanken mit identischen Benutzerkonten vorhanden. Die Verwaltung und Pflege der Benutzerkonten könnte bei Systemstörungen mit Anomalien in den Datenbeständen verbunden sein, die durch die redundante Datenhaltung entstehen.

Darüber hinaus hätten die mit der Administration verbundenen Operationen auf den Datenbeständen der Benutzerkonten einen hohen Datenverkehr zur Folge, da Änderungen an jedes Computersystem mit einer Benutzerkontendatenbank übertragen werden müssten, um einen gemeinsamen aktuellen Zustand aller Benutzerkontendatenbanken sicherzustellen.

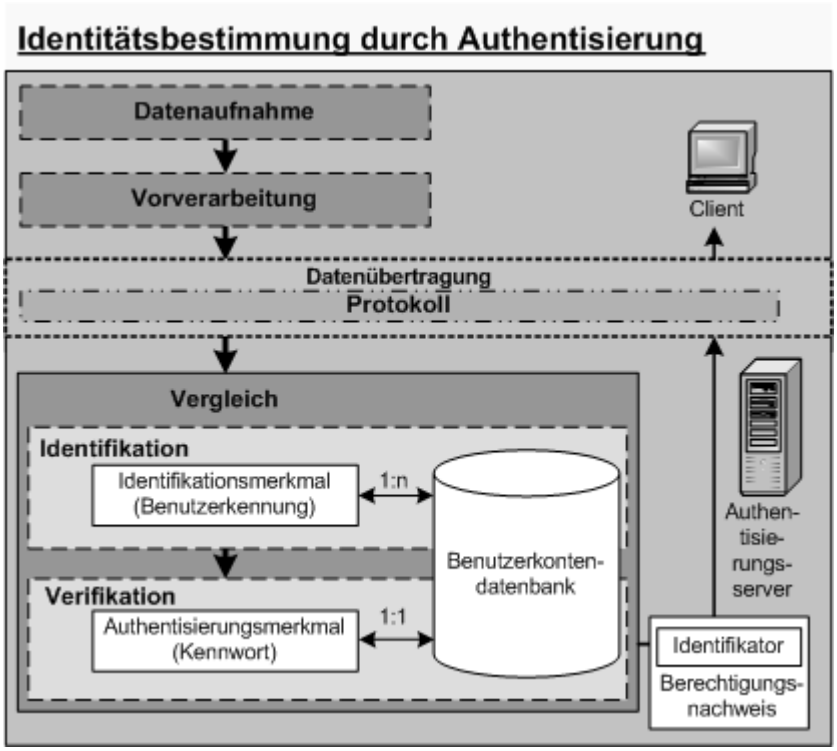

Abbildung 12: Architektur eines Client-Server-Authentisierungssystems

Weiterhin ergäben sich durch das dezentrale Vorhandensein mehrerer gleicher Benutzerkontendatenbanken eine Menge von Angriffspunkten, bei denen eine zuverlässige Absicherung nicht gewährleistet werden kann. Hier besteht an mehreren Orten die Möglichkeit die Benutzerkontendatenbanken anzugreifen und die Daten der Benutzerkonten zu kopieren, um sie später für das Eindringen in die Computersysteme des Netzwerks zu missbrauchen.

3.1.2 Übertragung der Authentisierungsdaten

Ein allgemeines Problem bei der Authentisierung im Netzwerk ist die Datenübertragung der Authentisierungsmerkmale, da diese geheim, die Verbindungen im Netzwerk aber offen und nicht sicher sind.

Die heute weit verbreitet eingesetzten Protokolle, wie TCP/IP, besitzen keine Sicherheitsmechanismen, die einer Kommunikationsbeziehung Eigenschaften wie Vertraulichkeit, Integrität, Authentizität und Unabstreitbarkeit verleihen.

Die Daten sind daher während der Übertragung vor Angriffen nicht geschützt und können mitgelesen und interpretiert werden (Angriff auf die Vertraulichkeit – z.B. durch Sniffing). Darüber hinaus ist nicht zweifelsfrei feststellbar, ob die Daten während der Übertragung verändert wurden (Verletzung der Integrität – z.B. durch Man in the Middle Angriff) und ob sie tatsächlich vom erwarteten Kommunikationspartner (Verletzung der Authentizitätseigenschaft – z.B. durch Spoofing) stammen [Gellert 1999, S.23].

Um so unverständlicher ist es, dass in einigen Client-Server-Authentisierungssystemen (z.B. bei Diensten im Internet) die Daten zur Authentisierung im Klartext übertragen werden. Eine solche Vorgehensweise ist nicht akzeptierbar.

Daher gibt es für die Authentisierung im Netzwerk spezielle Protokolle, die mit Hilfe von kryptografischen Verfahren die Angriffsmöglichkeiten einschränken. Diese enthalten eine Menge von Regeln für den Nachrichtenaustausch zwischen den Kommunikationspartnern, an deren Ende jeweils die Authentisierung mindestens eines Kommunikationspartners steht [Helden 1995, S.24].

Häufig kommt dabei das Challenge-Response-Verfahren zum Einsatz. Das Challenge-Response-Verfahren, ist ein Verfahren, bei dem das zur Authentisierung benutzte gemeinsame Geheimnis (z.B. Kennwort) durch die Nutzung kryptografischer Verfahren nie direkt über das Netzwerk übertragen wird. Stattdessen wird aus dem gemeinsamen Geheimnis ein Schlüssel abgeleitet, der zur Verschlüsselung einer Nachricht benutzt wird.

Während einer Authentisierungsanfrage sendet der Authentisierungsserver dem Client eine Klartextnachricht (Challenge). Der Client verschlüsselt diese mit einem aus dem gemeinsamen Geheimnis (z.B. Kennwort) abgeleiteten Schlüssel und sendet die verschlüsselte Nachricht zurück an den Server (Response). Der Server kennt das gemeinsame Geheimnis, leitet daraus ebenfalls einen Schlüssel ab und entschlüsselt die Nachricht.

Stimmt der Inhalt der Klartextnachricht mit dem Inhalt der entschlüsselten Nachricht überein, so kennt der Client das gemeinsame Geheimnis und gilt als authentisiert.

Als Nachrichteninhalte kommen meist Zufallszahlen zum Einsatz. Dadurch werden Angriffe durch Wiedereinspielung (Replay-Angriffe) erschwert, da bei jeder Authentisierung andere Zufallszahlen genutzt werden. Es ist jedoch möglich den Nachrichtenverkehr abzuhören, aufzuzeichnen und danach mit Hilfe von Brute-Force Angriffen nach dem gemeinsamen Geheimnis zu suchen.

Durch das Challenge-Response-Verfahren kann weiterhin eine gegenseitige Authentisierung (mutual authentication) stattfinden. Hierbei prüft nicht nur der Server die Authentizität des Clients, sondern der Client auch die Authentizität des Servers. Damit wird gewährleistet, dass mit dem gewünschten Authentisierungsserver eine Verbindung aufgebaut wird.

Neben dem Challenge-Response-Verfahren gibt es eine weitere Möglichkeit einen sicheren Datenaustausch zwischen dem Client und dem Server zu ermöglichen. Diese besteht in der Etablierung eines sicheren Kanals mit Hilfe kryptografischer Verfahren[13].

Während des Verbindungsaufbaus werden dabei symmetrische Schlüssel errechnet, mit denen der nachfolgende Datenverkehr geschützt wird, wobei für jede weitere Verbindung neue symmetrische Schlüssel errechnet werden. Die Etablierung eines symmetrischen Schlüssel mit dem der Datenverkehr geschützt wird erfolgt hierbei durch Protokolle zum Schlüsselaustausch.

Diese Methode hat den Nachteil, dass die Merkmale zum Authentisierungsserver direkt übertragen werden. Der Vorteil von Challenge-Response-Verfahren gegenüber der Etablierung eines sicheren Kanals ist es, dass keine direkte Übertragung der Merkmale stattfindet.

Viele Authentisierungssysteme setzen eigene Protokolle zur Authentisierung ein. Diese basieren häufig auf dem Challenge-Response-Verfahren. Die Qualität dieser Verfahren ist von der Erzeugung der verwendeten Zufallszahlen abhängig.

13 Ein Protokoll zur Etablierung eines sicheren Kanals ist z.B. das TLS Protokoll [RFC 2246].

3.2 Kerberos

Kerberos ist ein Authentisierungssystem für offene Netzwerke und wurde im Rahmen des Athena Projekts am Massachusetts Institute of Technology (MIT) entwickelt [Steiner et al 1988, S.8].

Es basiert auf dem *Needham-Schroeder Key Distribution Protokoll* [Needham 1978] und wurde entwickelt, um unbekannte Einheiten in einem unsicheren Netzwerk von nicht vertrauenswürdigen Rechnern zu authentisieren und nachzuweisen, dass eine Entität (Benutzer, Server, Rechner oder Dienst) wirklich diejenige ist, welche sie vorgibt.

Mit Kerberos ist eine gegenseitige Authentisierung (mutual authentication) möglich, sodass sich Clienten und Server gegenseitig ihre Identität prüfen können. Weiterhin kann mit Kerberos festgestellt werden, ob eine Entität in einem Netzwerk eine bestimmte Ressource nutzen darf.

Bisher wurden zwei Kerberos Versionen veröffentlicht: Version 4 und 5. Die Version 4 wurde getestet und wies Schwächen auf, die in [Bellovin Merrit 91] dokumentiert sind. Die bekannten Schwächen nahmen Einfluss auf die Entwicklung der Version 5, die in [RFC 1510] dokumentiert ist.

3.2.1 Komponenten und allgemeines Funktionsprinzip

Kerberos nutzt zur Authentisierung eine zentrale dritte vertrauenswürdige Instanz, die mit Hilfe von symmetrischen Verschlüsselungsverfahren Dienste zur Authentisierung von Clienten und Servern (Prinzipalen[14]) zur Verfügung stellt. Die Authentisierung erfolgt durch die Kenntnis eines gemeinsamen geheimen Schlüssels, der nur dem Prinzipal und dem Kerberossystem bekannt ist, wobei die Fähigkeit eine Nachricht zu entschlüsseln als Identitätsnachweis gilt. Darüber hinaus kommen *Tickets*[15] zum Einsatz, die zur Nutzung eines bestimmten Dienstes im Netzwerk berechtigen und Informationen enthalten mit denen eine Authentizitätsprüfung erfolgen kann. Diese sind mit einem geheimen Schlüssel des Netzwerkdienstes verschlüsselt und werden von den Diensten des Kerberossystems vergeben.

14 Der Begriff Prinzipal bezeichnet einen Kerberos Client. Der Begriff wurde eingeführt, um Kerberos Clients im Netzwerk von den Clients anderer Dienste unterscheiden zu können. [Steiner et al 1988, S.2]

15 Tickets werden immer nur von den Kerberos Diensten erzeugt.

Die Authentisierung in einem Kerberossystem verläuft in drei Phasen und wird durch das Zusammenwirken mehrerer Komponenten zur Verfügung gestellt. Zentrale Komponenten eines Kerberossystems sind die Kerberos-Datenbank, der Authentisierungs-Server, der Ticket-Granting-Server und die dienstanbietenden Server.

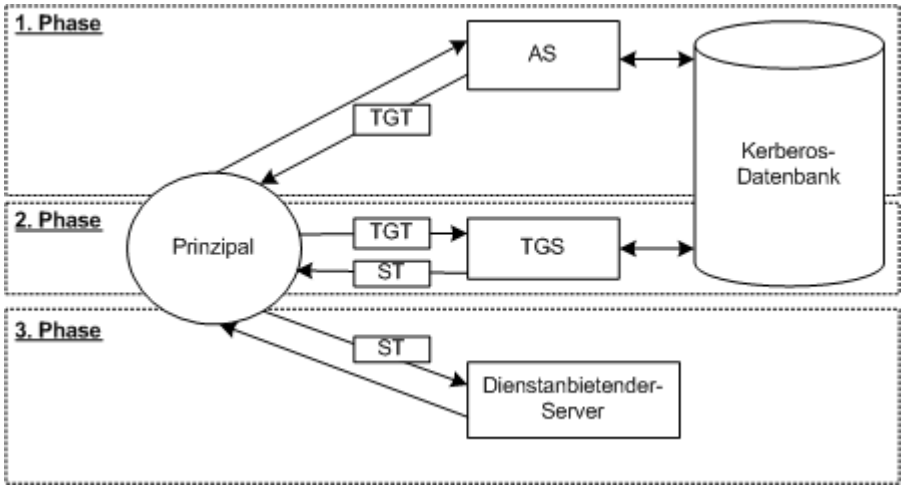

Abbildung 13: Komponenten eines Kerberossystems

Kerberos-Datenbank: Die Kerberos-Datenbank dient zur Speicherung von Prinzipalnamen und den dazugehörigen geheimen Schlüsseln und wird vom Authentisierungs- und Ticket Granting Server als Datenbasis und zur Ausgabe von Tickets verwendet.

Authentisierungsserver: Der Authentisierungsserver[16] dient der initialen Authentisierung eines Prinzipals. In einem Kerberos Authentisierungssystem meldet sich ein Prinzipal[17] an einem zentralen Authentisierungsserver (AS) an und erhält bei erfolgreicher Authentisierung ein Ticket (siehe Abbildung 13), das Ticket-Granting-Ticket (TGT) genannt wird und zum Bezug von dienstspezifischen Tickets (Server-Tickets, ST) bei einem Ticket-Granting-Server dient.

16 Der Authentisierungsserver wird in der Kerbrosliteratur wie [Steiner et al1988] auch Kerberosserver genannt.

17 Der Terminus Prinzipal wird nach [Steiner et al 1988] verwendet um Kerberos Clienten von anderen Clienten im Client Server Netzwerk zu unterscheiden. Ein Prinzipal ist eine Entität, die einen Kerberos Dienst nutzt und kann ein Benutzer oder ein Dienst sein.

Ticket-Granting-Server: Die Nutzung eines bestimmten Dienstes im Netzwerk erfordert ein dienstspezifisches Server-Ticket. Während der zweiten Phase (siehe Abbildung 13) fordert der Prinzipal nun mit Hilfe des Ticket-Granting-Tickets ein Server-Ticket für einen bestimmten Dienst vom Ticket-Granting-Server an.

Dienstanbietende Server: In der dritten Phase kann nun der Prinzipal durch Vorlage des Server-Tickets (ST) einen bestimmten Dienst im Netzwerk nutzen.

3.2.2 Erste Phase - AS Austausch

Die initiale Authentisierung eines Prinzipals erfolgt mit Hilfe von zwei Nachrichten. Zuerst wird mit einer KRB_AS_REQ Nachricht ein Ticket-Granting-Ticket vom Authentisierungsserver (AS) angefordert. Die Nachricht wird im Klartext übertragen und enthält den Prinzipalnamen und eine Kennung für den Ticket-Granting-Server, von dem weitere Tickets (Server-Tickets) bezogen werden können.

Nach Empfang der KRB_AS_REQ Nachricht prüft der Authentisierungsserver, ob es für den Prinzipal einen Eintrag in der Kerberos Datenbank gibt. Ist das der Fall, so generiert er ein Ticket-Granting-Ticket für den gewünschten Ticket-Granting-Server, das mit seinem geheimen Schlüssel[18] verschlüsselt ist.

Das verschlüsselte Ticket-Granting-Ticket besteht aus einem Prinzipal-Identifikator, einem Zeitstempel, der IP Adresse des Prinzipals, der Lebensdauer des Tickets und einem Sitzungsschlüssel zur Verschlüsselung der Verbindung zwischen Client und Ticket-Granting-Server.

Es wird in einer KRB_AS_REP Nachricht an den Prinzipal gesendet, die neben dem verschlüsselten Ticket-Granting-Ticket weitere Informationen, wie eine Kopie des Sitzungsschlüssels und Zusatzinformationen zur Validierung wie Prinzipalidentifikator, Zeitstempel und Lebensdauer, enthält. Darüber hinaus ist die KRB_AS_REP Nachricht mit dem geheimen Prinzipalschlüssel[19] verschlüsselt, der nur dem Prinzipal und dem Authentisierungsserver bekannt ist.

---

18 Der geheime Schlüssel des Ticket-Granting-Server ist nur dem Authentisierungsserver und dem Ticket-Granting-Server bekannt.

19 Handelt es sich bei dem Prinzipal um eine Person, so wird der geheime Prinzipalschlüssel vom Kennwort abgeleitet.

Nach Empfang entschlüsselt der Prinzipal die Nachricht mit seinem geheimen Prinzipalschlüssel, vergleicht den Zeitstempel und entnimmt den Sitzungsschlüssel und das verschlüsselte Ticket-Granting-Ticket.

3.2.3 Zweite Phase - TGS Austausch

Während der zweiten Phase fordert der Prinzipal mit Hilfe des Ticket-Granting-Ticket beim Ticket-Granting-Server (TGS) ein Server-Ticket für einen bestimmten Dienst im Netzwerk an, wobei die zweite Phase wieder durch den Austausch von zwei Nachrichten gekennzeichnet ist.

Der Prinzipal generiert eine KRB_TGS_REQ Nachricht, die neben dem verschlüsselten Ticket-Granting-Ticket und dem Prinzipal-Identifikator des gewünschten Netzwerkdienstes einen Authentikator enthält, der mit dem vorher erhaltenen Sitzungsschlüssel des Ticket-Granting-Server verschlüsselt ist. Der Authentikator wird immer vom Client-Prinzipal erzeugt und enthält die IP-Adresse des Client-Prinzipals und einen Zeitstempel.

Nach Erhalt der KRB_TGS_REQ Nachricht entschlüsselt der Ticket-Granting-Server mit seinem geheimen Schlüssel das Ticket-Granting-Ticket und entnimmt den darin enthaltenen Sitzungsschlüssel, mit dem er danach den Authentikator entschlüsselt. Stimmen die Identitätsinformationen des Client-Prinzipals im Ticket mit denen im Authentikator überein, so gilt die Identität des Clients als bewiesen. Der Ticket-Granting-Server erzeugt nun ein Server-Ticket für den gewünschten Server, das analog dem Ticket-Granting-Ticket aufgebaut ist. Das Server-Ticket wird dann mit dem geheimen Schlüssel des gewünschten Server verschlüsselt, der wieder aus der Kerberos Datenbank entnommen wird. Anschließend wird eine KRB_TGS_REP Nachricht erzeugt, die das verschlüsselte Server-Ticket, eine Kopie des Sessionkey und weitere Informationen wie Client-Prinzipal-Identifier, Zeitstempel und Lebensdauer enthält. Die so generierte Antwort wird nun mit dem Sessionkey des Ticket-Granting-Ticket verschlüsselt und an den Client-Prinzipal zurückgesendet.

Der Client-Prinzipal entschlüsselt die Nachricht mit seinem gespeicherten Sitzungsschlüssel, bewahrt das verschlüsselte Server-Ticket für die weitere Nutzung auf und initiiert eine Anfrage an den Netzwerkdienst, den er benutzen möchte.

3.2.4 Dritte Phase - CS Austausch

Zur Inanspruchnahme eines Dienstes sendet der Client-Prinzipal eine entsprechende Nachricht (KRB_AP_REQ) an den Server. Diese Nachricht ist der Nachricht KRB_TGS_REQ an den TGS vergleichbar. Es fehlt jedoch die Angabe des gewünschten Servers. Die Nachricht KRB_AP_REQ dient der Initiierung der Kommunikation und der Übertragung eines Sessionkeys innerhalb des Server-Tickets. Falls der Client die Identität des Servers prüfen möchte, so kann er veranlassen, dass der Server mit einer Nachricht KRB_AP_REP antwortet, die der Bestätigung seiner Identität dient. Bestandteil dieser Nachricht ist der um eins erhöhte Wert des vom Client gesendeten Zeitstempels. Diese Antwortnachricht ist ebenfalls mit dem Sessionkey verschlüsselt. Durch Prüfung dieser Antwort kann der Client feststellen, ob die Identität des Servers stimmt. Nur der Server kann den Sessionkey kennen, da nur er in der Lage ist, das vom Client gelieferte Ticket zu entschlüsseln.

Kapitel 4

Windows 2000

Die Implementierung eines Iris-Biometrik-Verfahrens in das Windows 2000 Authentisierungssystem erfordert Kenntnisse über die internen Anmelde- und Authentisierungsmechanismen des Betriebssystems. Hierbei ist wichtig, welche Komponenten an Anmelde- und Authentisierungsvorgängen beteiligt sind, wie sie miteinander zusammenarbeiten und ob eine Ergänzung oder ein Austausch einzelner Komponenten möglich ist.

Dazu wird im Abschnitt 4.1 „Betriebssystemarchitekturüberblick“ der Aufbau des Windows 2000 Betriebssystems dargestellt, wobei auf die Komponenten im Benutzer- und Kernel- Modus eingegangen wird.

Abschnitt 4.2 „Das Integrale Sicherheitssubsystem“ gibt eine Übersicht über die Architektur des integralen Sicherheitssubsystems, das Sicherheitsmechanismen des Betriebssystems, wie Zugangskontrolle, Zugriffskontrolle und Überwachung und damit auch Anmelde- und Authentisierungsmechanismen bereitstellt.

Abschnitt 4.3 „Anmeldearchitektur“ stellt die Aufgaben und Funktionsweise von Komponenten des integralen Sicherheitssubsystems dar, die Anmelde- und Authentisierungsmechanismen zur Verfügung stellen.

Abschließend wird in Abschnitt 4.4 „Authentisierung im Netzwerk“ der Ablauf einer Anmeldung am Betriebssystem mit Hilfe von NTLM und Kerberosauthentisierung dargestellt, um die Authentisierungsmechanismen und das Zusammenwirken der Komponenten der Anmeldearchitektur des integralen Sicherheitssubsystems aufzuzeigen.

4.1 Betriebssystemarchitekturüberblick

Windows 2000 ist ein Betriebssystem der Firma Microsoft und gehört zur Familie der Windows NT Betriebssysteme, deren Entwicklung Ende 1988 begann. Es wird seit Februar 2000 verkauft und ist der Nachfolger von Windows NT[20] 4.0. Das Windows 2000 Betriebssystem ist modular aufgebaut und besteht aus mehreren Komponenten, deren Zusammenarbeit die Betriebssystemfunktionalität bereitstellt.

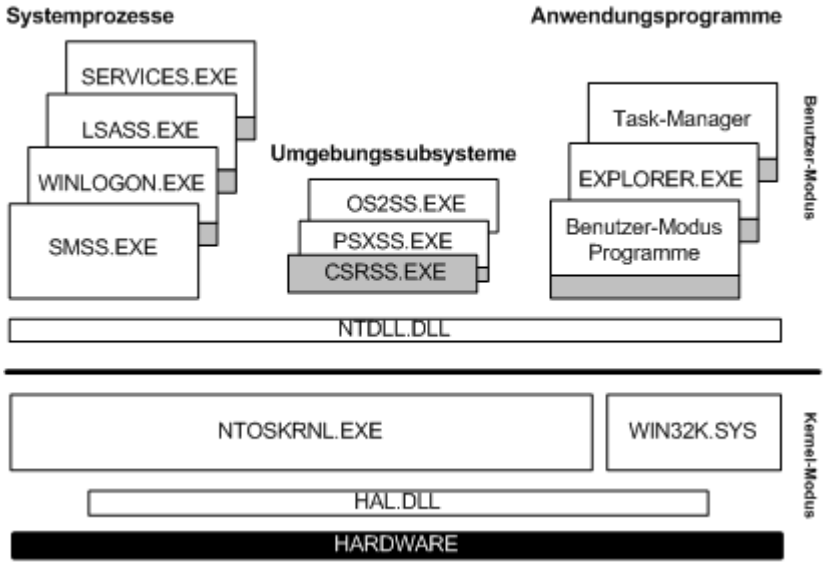

Abbildung 14: Architektur des Windows 2000 Betriebssystems

Die Komponenten werden in bestimmten von der Hardware bereitgestellten Betriebsarten (Benutzer- und Kernel-Modus) ausgeführt, in denen Programme unterschiedliche Rechte auf den Hardware- und Speicherzugriff besitzen.

Der Kernel-Modus ist eine priorisierte Betriebsart, bei der Programme direkt auf die Hardware und den gesamten Adressraum des Speichers zugreifen können. Der Benutzer-Modus ist im Gegensatz dazu eine Betriebsart mit weniger Rechten, bei der Programme nur auf ihren eigenen Adressraum und nicht direkt auf die Hardware zugreifen können.

20 Die Abkürzung NT steht dabei für Neue Technologie (New Technology) und symbolisiert einen Neuanfang bei der Betriebssystementwicklung der Firma Microsoft, die bis zum Beginn der Windows NT Betriebssystementwicklung über ein am Markt sehr erfolgreiches, aber technologisch veraltetes Betriebssystem (Microsoft Disk Operating System, MSDOS) für IBM kompatible Microcomputer verfügte.

4.1.1 Komponenten im Kernel-Modus

Im Kernel-Modus wird ein Betriebssystemkern NTOSKRNL.EXE und das Fensterverwaltungs- und Grafiksystem WIN32K.SYS ausgeführt. Beide Komponenten benutzen zur Hardwareansteuerung die Funktionen einer dynamischen Bibliothek (HAL.DLL), die auch Hardwareabstraktionsschicht genannt wird.

Die Hardwareabstraktionsschicht ist in Assemblercode implementiert und verbirgt die Eigenschaften konkreter Hardwareplattformen, wie E/A Schnittstellen, Unterbrechungskontroller und Multiprozessorkommunikationssystemen hinter einer Standardschnittstelle (Application Programming Interface, API).
Dadurch sehen die unterschiedlichen Hardwareaufrufe oberhalb der Hardwareabstraktionsschicht gleich aus und es wird möglich, Programmcode, der zum Ansprechen der Hardware die Hardwareabstraktionsschicht benutzt, wiederzuverwenden.

Der Betriebssystemkern und das Fensterverwaltungs- und Grafiksystem stellen dem Benutzer-Modus Systemdienste durch exportierte Funktionen bereit, deren Schnittstellen in einer weiteren dynamischen Bibliothek (NTDLL.DLL) enthalten sind.

4.1.2 Komponenten im Benutzer-Modus

Im Benutzer-Modus werden mehrere Systemprozesse ausgeführt, die zum Betriebssystem gehören und die Systemdienste der Kernel-Modus-Komponenten benutzen. Diese können überwiegend bestimmten Subsystemen zugeordnet werden, die in Umgebungssubsysteme und das integrale Sicherheitssubsystem unterschieden werden.

Umgebungssubsysteme stellen Programmen im Benutzer-Modus die Systemdienste des Betriebssystemkerns in Form einer bestimmten Betriebssystemumgebung mit Hilfe der dazugehörigen, betriebssystemspezifischen Schnittstellenfunktionen zur Verfügung. Anwendungsprogramme nutzen die Systemdienste des Betriebssystemkerns nicht direkt, sondern indirekt über ein Umgebungssubsystem, das die Systemdienste des Betriebssystemkerns in der jeweiligen betriebssystemspezifischen Form offen legt.

Das Windows 2000 Betriebssystem verfügt über Umgebungssubsysteme für Win32 (CSRSS.EXE), OS/2 (OS2SS.EXE) und POSIX (PSXSS.EXE) Umgebungen. Dadurch wird es möglich, Programme, die für ältere Windowsversionen,

OS/2 oder POSIX geschrieben wurden, ebenfalls auf Windows 2000 auszuführen. Das Win32 Umgebungssubsystem ist aber das primäre Umgebungssubsystem des Betriebssystems, auf dem andere Umgebungssubsysteme sowie der überwiegende Teil der Betriebssystemkomponenten im Benutzer-Modus basiert.

Das integrale Sicherheitssubsystem ist ein weiteres Subsystem im Benutzer-Modus und stellt Sicherheitsmechanismen wie Identifikation, Authentifikation, Zugriffskontrolle, Überwachung und kryptographische Dienste bereit. Zu den Systemprozessen des integralen Sicherheitssubsystems gehören der interaktive Anmeldeprozess WINLOGON.EXE und die lokale Sicherheitsautorität LSASS.EXE.

Weiterhin gibt es im Benutzer-Modus Systemprozesse, die weder zu den Umgebungssubsystemen, noch zu dem integralen Sicherheitssubsystem gehören, wie der Sitzungs-Manager SMSS.EXE [Russinovich 1998, Russinovich 1999] oder der Dienststeuerungs-Manager SERVICES.EXE. Der Sitzungs-Manager ist für die Initialisierung des Betriebssystems verantwortlich und überwacht den Zustand des Betriebssystems. Der Dienststeuerungs-Manager stellt Prozessen Funktionen zur Steuerung von Windows 2000 Diensten bereit und ist die zentrale Komponente zur Diensteverwaltung im Betriebssystem.

4.1.3 Registrierungsdatenbank

Eine Besonderheit in den Betriebssystemen der Windows NT-Familie ist die Registrierungsdatenbank, in der Konfigurationseinstellungen der Betriebssystemkomponenten und der installierten Programme gespeichert sind.

Das Betriebssystem wird durch die Einträge in der Registrierungsdatenbank gesteuert, indem die entsprechenden Initialisierungseinstellungen zum Laden von Komponenten zur Verfügung gestellt werden. Fehlerhafte Einträge können zur Folge haben, dass das System nicht mehr korrekt ausgeführt wird oder überhaupt nicht mehr hochfährt.

Der Aufbau der Registrierungsdatenbank ist hierarchisch und setzt sich aus mehreren Teilbäumen (Subtrees) zusammen, die Schlüssel (Keys) mit Unterschlüsseln (Subkeys) und Werteinträgen (Entrys) enthalten. Jeder Unterschlüssel ist dabei wieder ein Schlüssel und kann aus mehreren Unterschlüsseln und Werteinträgen bestehen. Die Werteinträge dienen zur Speicherung von Daten und setzen sich aus einem Namen (Entryname), Datentyp (Datatype) und einem Wert (Value) zusammen. Die Registrierungsdatenbank kann mit dem Programm REGEDIT.EXE bearbeitet werden.

4.2 Das integrale Sicherheitssubsystem im Überblick

Das integrale Sicherheitssubsystem stellt die Windows 2000 Sicherheitsfunktionen zur Verfügung und wurde nach den C2 Anforderungen der Trusted Computer System Evaluation Criteria (TCSEC) des US-Verteidigungsministeriums [US DoD 1983] entworfen. Es verfügt über folgende Funktionen:

Anmeldefunktion, die den Benutzer zur Eingabe seiner Benutzerkennung auffordert und ihm Zugang zum Betriebssystem gewährt, nachdem eine Authentisierung erfolgreich stattgefunden hat.

Diskrete Zugriffskontrolle, die es dem Besitzer einer Ressource ermöglicht festzulegen, wer Zugriff auf eine Ressource erhält und welche Operationen Benutzer der Ressource ausführen dürfen.

Überwachungsfunktion, die es ermöglicht, versuchte Zugriffe auf die Systemressourcen zu erkennen und aufzuzeichnen.

Funktionen zum Schutz vor Objektwiederverwendung, die verhindern, auf die gelöschten Daten anderer Benutzer zuzugreifen.

Darüber hinaus stellt das integrale Sicherheitssubssystem kryptografische Dienste zur Verfügung, die sowohl eine Verschlüsselung und Signierung von Daten als auch einen Schutz von Kommunikationsverbindungen ermöglichen.

Die Schutzfunktionen des integralen Sicherheitssubsystems werden von Komponenten bereitgestellt, die überwiegend im Benutzer-Modus ausgeführt werden. Hauptkomponenten des integralen Sicherheitssubsystems im Benutzer-Modus sind der interaktive Anmeldeprozess WINLOGON.EXE und die lokale Sicherheitsautorität LSASS.EXE. Diese nutzen die Dienste verschiedener anderer Komponenten, die als dynamische Bibliotheken implementiert wurden und ebenfalls zum integralen Sicherheitssubsystem gehören wie die grafische Identifizierungs- und Authentisierungsschnittstelle (Graphical Identification and Authentication Interface, GINA, MSGINA.DLL), Authentisierungspakete (MSV1_0.DLL, KERBEROS.DLL), Sicherheitspakete, den Anmeldedienst (NETLOGON.DLL), den Dienst der lokalen Sicherheitsautorität (LSASRV.DLL), den Sicherheitskonten-Manager (Security Account Manager, SAM, SAMSRV.DLL), den Active Directory Verzeichnisdienst (NTDSA.DLL) sowie Datenbanken wie die Sicherheitskontendatenbank, in der sich Informationen über registrierte Benutzer befinden, und die Richtliniendatenbank, die

Informationen über Rechte im Betriebssystem enthält. Im Kernel-Modus befinden sich als Bestandteil des Betriebssystemkerns der Sicherheitsreferenzmonitor und Kernelsicherheitsgerätetreiber (Kernel Security Device Driver, KsecDD, KSECDD.SYS). Nachfolgend wird ein einführender Überblick über die Funktionen der einzelnen Komponenten des integralen Sicherheitssubsystems gegeben.

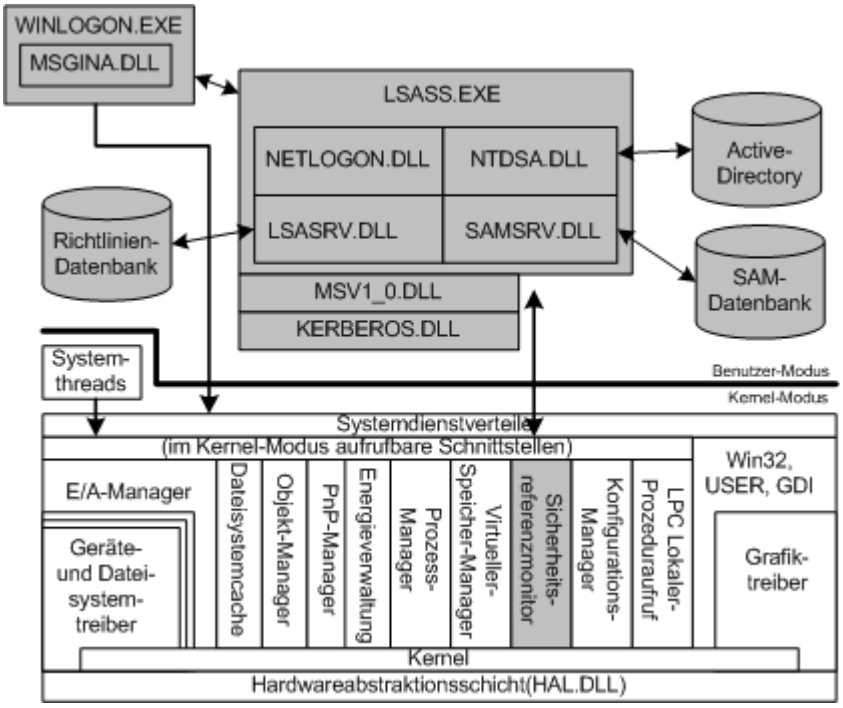

Abbildung 15: Architektur des integralen Sicherheitssubsystems

Sicherheitsreferenzmonitor: Der Sicherheitsreferenzmonitor ist Bestandteil des Betriebssystemkerns NTOSKRNL.EXE und wird im Kernel-Modus ausgeführt. Er prüft die Berechtigungen von Objekten, überwacht die Objektzugriffe und Operationen im Betriebssystem und erzeugt entsprechende Überwachungsnachrichten, die an die lokale Sicherheitsautorität gesendet werden.

Lokale Sicherheitsautorität LSASS.EXE: Die lokale Sicherheitsautorität LSASS.EXE ist die zentrale Komponente des integralen Sicherheitssubsystems und wird als Systemprozess im Benutzer-Modus ausgeführt. Sie ist für mehrere unterschiedliche Aufgaben wie die Authentisierung, die Verwaltung der lokalen Sicherheits- und Überwachungsrichtlinie und das

Schreiben von Überwachungsnachrichten des Sicherheitsreferenzmonitors in die Ereignisprotokolle zuständig. Die lokale Sicherheitsautorität nutzt dazu mehrere dynamische Bibliotheken, die die einzelnen Sicherheitsmechanismen zur Verfügung stellen.

Interaktiver Anmeldeprozess WINLOGON.EXE: Der interaktive Anmeldeprozess WINLOGON.EXE ist wie die lokale Sicherheitsautorität eine Komponente, die als Systemprozess im Benutzer-Modus ausgeführt wird. Zu den Aufgaben von WINLOGON.EXE zählen die Verwaltung von interaktiven Sitzungen, das Bereitstellen der An- und Abmeldemöglichkeit am Betriebssystem, die Initialisierung der Schutzmechanismen für die grafische Benutzerschnittstelle, das Laden von Benutzerprofilen, das Erkennen von Nachrichten zur Steuerung der Sitzungen und die Steuerung des Bildschirmschoners. Dazu benutzt WINLOGON.EXE ebenfalls dynamische Bibliotheken (z.B. MSGINA.DLL), die einzelne Sicherheitsmechanismen zur Verfügung stellen.

MSGINA.DLL: MSGINA ist eine dynamische Bibliothek, die dem interaktiven Anmeldeprozess WINLOGON.EXE die grafische Benutzerschnittstelle mit Anmeldedialogen zur Verfügung stellt[21]. Der Name GINA ist dabei eine Abkürzung für grafische Identifizierungs- und Authentisierungsschnittstelle (Graphical Identification and Authentication Interface) und drückt den Verwendungszweck von GINA als Komponente des integralen Sicherheitssystems aus. Dabei ist zu beachten, dass GINA nicht die Authentisierung selbst durchführt, sondern sie nur vorbereitet, indem sie die Anmeldeinformationen aufnimmt und an den Systemprozess der lokalen Sicherheitsautorität LSASS.EXE zur Validierung weiterleitet. Darüber hinaus wird mit Hilfe von GINA die interaktive Sitzung eines Benutzers gesteuert, indem Sicherheitssequenzen[22] des Benutzers abgefangen werden und Behandlungsfunktionen ausgeführt werden, die den interaktiven Anmeldeprozess steuern.

Authentisierungspakete: Authentisierungspakete sind dynamische Bibliotheken, die von der lokalen Sicherheitsautorität LSASS.EXE zur Authentisierung genutzt werden. Sie verarbeiten die im interaktiven Anmeldeprozess

21 Durch diese Architektur ist es möglich Trojanische Pferde in das Betriebssystem zu bringen, die die Benutzereingaben während der Anmeldung in Listen speichern. Ein „Trojanisches Pferd“ ist ein (ausführbares) Programm, dem unerkannt zusätzliche Funktionen aufgeprägt sind [Brunnstein 1991, S.22]

22 Eine Sicherheitssequenz ist ein Signal, das durch den Benutzer und das Sicherheitssystem ausgelöst werden kann, um bestimmte Benachrichtigungen an einzelne Komponenten zur Steuerung zu übermitteln. Eine Sicherheitssequenz ist z.B. die Tastenkombination „Str+Alt+Entf“.

WINLOGON.EXE erfassten und zur lokalen Sicherheitsautorität übermittelten Anmeldedaten und überprüfen mit bestimmten Verfahren die Identität eines Benutzers.

Sicherheitspakete: Sicherheitspakete sind dynamische Bibliotheken, die Sicherheitsunterstützungsanbieter (Security Support Provider, SSP) implementieren. Sicherheitsunterstützungsanbieter stellen über die Sicherheitsunterstützungsanbieter-Schnittstelle (Security Support Provider Interface, SSPI) Sicherheitsdienste zur Verfügung, die zur Authentisierung im Netzwerk genutzt werden können, und regeln wie der Nachrichtenaustausch erfolgt.

Netzwerkanmeldedienst NETLOGON.DLL: Der Netzwerkanmeldedienst ist ein Windows 2000 Dienst, der im Kontext der lokalen Sicherheitsautorität LSASS.EXE ausgeführt wird. Er dient zur Ausführung der NTLM[23] Authentisierung über das Netzwerk und verfügt über einen integrierten Suchdienst für Domänenkontroller. [Solomon Russinovich 2000, S.417]

Sicherheitskonten-Manager: Der Sicherheitskonten-Manager (Security Account Manager, SAM) ist für die Verwaltung der lokalen Benutzer- und Gruppenkonten in der Sicherheitskontendatenbank zuständig. Er ist in der Datei SAMSRV.DLL implementiert und wird im Kontext der lokalen Sicherheitsautorität LSASS.EXE ausgeführt. Während der Authentisierung wird der Sicherheitskonten-Manager genutzt, um die Informationen aus der Sicherheitskontendatenbank (Benutzerkontendatenbank) zu lesen.

Sicherheitskontendatenbank: In der Sicherheitskontendatenbank sind Informationen über die Benutzer- und Gruppenkonten gespeichert, wobei Windows 2000 über eine Sicherheitskontendatenbank in der Registrierung und im Active Directory verfügt. Die Sicherheitskontendatenbank der Registrierung befindet sich unter dem geschützten Teilbaum HKEY_LOCAL_MACHINE\SAM und wird immer dann verwendet, wenn sich ein Benutzer lokal am Betriebssystem anmeldet. Das „Active Directory“ ist der Windows 2000 Verzeichnisdienst und wird auf den Domänenkontrollern im Windows 2000 Netzwerk ausgeführt. Die Informationen aus dem Active Directory werden immer dann genutzt, wenn sich ein Benutzer in einer Windows 2000 Domäne anmeldet.

Richtliniendatenbank: Die Richtliniendatenbank dient zur Speicherung der Einstellungen der lokalen Systemrichtlinie und befindet sich in der Registrierungsdatenbank unter dem Schlüssel HKEY_LOCAL_MACHINE\SECURITY. Die lokale Systemrichtline enthält Richtlinien für

23 Die Abkürzung NTLM bedeutet New Technology Lan Manager.

Benutzer, in denen festgelegt ist, welche Benutzer sich am lokalen System anmelden dürfen, wie lange Kennwörter verwendet werden können und welche Benutzerrechte (Privilegien) Benutzer und Gruppen im System besitzen. Die Benutzerrechte legen dabei nicht die Rechte eines bestimmten Objekts fest, sondern welche Aktionen ein Benutzer im System ausführen darf, z.B. das Herunterfahren des Computers oder das Ändern der Systemzeit.

Kernelsicherheitsgerätetreiber: Die Kernelsicherheitsgerätetreiber sind in der Datei KSECDD.SYS enthalten, implementieren die Dateisystemverschlüsselung und werden im Kernel-Modus ausgeführt.

4.3 Anmeldearchitektur

Die Windows 2000 Anmeldearchitektur besteht aus mehreren Komponenten, deren Zusammenarbeit eine Anmeldung am Betriebssystem ermöglicht. Hauptkomponenten sind dabei der interaktive Anmeldeprozess WINLOGON.EXE und die lokale Sicherheitsautorität LSASS.EXE (Abbildung 16). Der interaktive Anmeldeprozess stellt mit Hilfe einer austauschbaren dynamischen Bibliothek (MSGINA.DLL) die grafische Benutzerschnittstelle mit Anmelde- und Sitzungsdialogen zur Datenerfassung und Steuerung von Sitzungen bereit. Nach der Erfassung der Anmeldedaten übermittelt er diese an die lokale Sicherheitsautorität zur Überprüfung. Die lokale Sicherheitsautorität verwendet dazu Authentisierungspakete (AP/SSP), in denen die Algorithmen zur Überprüfung implementiert sind.

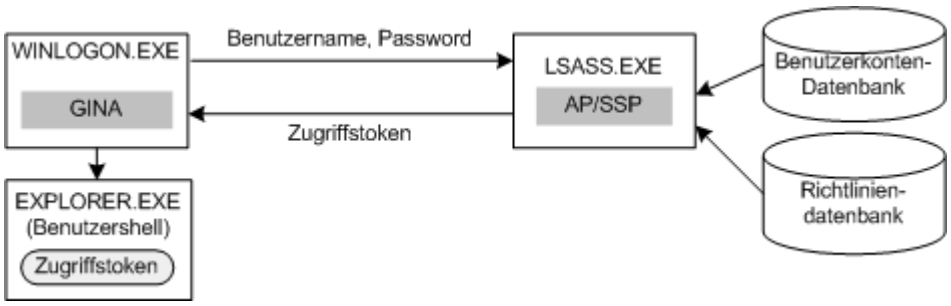

Abbildung 16: Anmeldung am Betriebssystem

Nach erfolgreicher Überprüfung wird ein Zugriffstoken erzeugt, das an den interaktiven Anmeldeprozess WINLOGON.EXE übermittelt wird. Das Zugriffs-

token enthält Informationen über die Identität, die Gruppenzugehörigkeiten und die Privilegien des Benutzers, und wird bei sämtlichen Zugriffen auf Objekte für die Zugriffskontrolle und Überwachung genutzt. Das integrale Sicherheitssubsystem erkennt die Identität des Benutzers anhand der Informationen im Zugriffstoken. Nach Erhalt des Zugriffstokens, startet WINLOGON.EXE die grafische Benutzershell (EXPLORER.EXE) als Benutzerprozess und bindet dabei das Zugriffstoken an den Benutzerprozess. Die weiteren vom Benutzer gestarteten Prozesse erben das Zugriffstoken des initialen Benutzerprozesses EXPLORER.EXE. In den folgenden Abschnitten werden die an der Anmeldung beteiligten Komponenten näher dargestellt.

4.3.1 Interaktiver Anmeldeprozess WINLOGON.EXE

Der interaktive Anmeldeprozess ist ein Systemprozess im Benutzermodus und wird vom Sitzungs-Manager SMSS.EXE während der Betriebssysteminitialisierung gestartet. Er startet die Systemprozesse der lokalen Sicherheitsautorität LSASS.EXE und des Dienststeuerungs-Managers SERVICES.EXE [Solomon Russinovich 2000, S.165] und ist darüber hinaus für das Erzeugen eines Arbeitsstationsobjekts und mehrerer Desktopobjekte zuständig.

Dazu erstellt WINLOGON.EXE eine interaktive Arbeitsstation namens „\Windows\WinSta0“, um Tastatur, Maus und Bildschirm für den Benutzer zugänglich zu machen [Solomon Russinovich 2000, S.443]. Das Arbeitsstationsobjekt wird dabei so initialisiert, dass nur der interaktive Anmeldeprozess darauf zugreifen kann. Dadurch kann kein anderer Prozess auf das interaktive Arbeitsstationsobjekt zugreifen, sofern nicht WINLOGON.EXE ausdrücklich den Zugriff gewährt.

Nach der Erstellung des Arbeitsstationsobjekts öffnet WINLOGON.EXE drei Desktops: Winlogon-Desktop „\Windows\WinSta0\Winlogon“, Anwendungs-Desktop „\Windows\WinSta0\ Default“ und Bildschirmschoner-Desktop „\Windows\WinSta0\Screen-Saver“.

Der Winlogon-Desktop dient zur Anzeige von Anmeldedialogen und wird wie die interaktive Arbeitsstation geschützt. Der Zugriff von anderen Prozessen auf den Inhalt der Anmeldedialoge wird so verhindert und ist nur von WINLOGON.EXE aus möglich. Der Winlogon-Desktop wird sofort nach seiner Erstellung aktiv und bleibt solange im Vordergrund, bis eine erfolgreiche interaktive Anmeldung stattgefunden hat.

Ist das der Fall, so bringt der interaktive Anmeldeprozess den Anwendungs-Desktop in den Vordergrund. Der Anwendungs-Desktop stellt dem Benutzer die grafische Arbeitsumgebung bereit und ist immer aktiv, wenn der Benutzer mit Programmen arbeitet. Er ist nicht so streng geschützt wie der Winlogon-Desktop. Sämtliche Programme können auf den Inhalt des Anwendungsdesktops zugreifen. Der Bildschirmschoner-Desktop dient zur Darstellung des Bildschirmschoners und wird immer dann aktiv, wenn eine bestimmte Zeit nicht mehr mit Programmen gearbeitet wurde.

Nach der Erstellung der Desktops stellt WINLOGON.EXE durch Aufruf der Funktion „LsaRegisterLogonProcess“ eine LPC-Verbindung mit dem LsaAuthenticationPort Anschluss der lokalen Sicherheitsautorität LSASS.EXE her, um während der An- und Abmeldevorgänge die Anmeldedaten an die lokale Sicherheitsautorität zu übertragen. Anschließend richtet der interaktive Anmeldeprozess die Fensterumgebung ein und registriert und initialisiert eine Datenstruktur einer Fensterklasse, die eine Winlogon-Prozedur mit dem Fenster verknüpft, das nachfolgend erstellt wird. Darüber hinaus registriert WINLOGON.EXE eine Sicherheitssequenz (Secure Attention Sequence, SAS) und ordnet diese dem erstellten Fenster zu, sodass die Winlogon-Fensterprozedur aufgerufen wird, wenn ein Benutzer eine Tastenkombination drückt, die als Sicherheitssequenz definiert wurde [Solomon Russinovich 2000, S.443].

Nach der Initialisierung befindet sich Winlogon ständig in einem von drei Zuständen, wobei diese in den Zustand „Angemeldet“, „Abgemeldet“ und „Gesperrt“ differenziert werden.

Im Zustand „Abgemeldet“ befindet sich der interaktive Anmeldeprozess nach erfolgreicher Initialisierung und aktiviert den Winlogon-Desktop. Der Benutzer wird aufgefordert die Tastenkombination „Strg+Alt+Entf“ zu drücken und anschließend im Anmeldedialog seine Anmeldedaten einzugeben. Nach erfolgreicher Anmeldung wechselt Winlogon in den Zustand „Angemeldet“.

Im Zustand „Angemeldet“ wechselt der interaktive Anmeldeprozess zum Anwendungs-Desktop und startet die grafische Benutzershell (EXPLORER.EXE), um den Benutzer eine grafische Arbeitsumgebung zur Verfügung zu stellen. Dort kann der Benutzer Anwendungsprogramme starten und seine Arbeit verrichten. Wird im Zustand „Angemeldet“ eine Sicherheitssequenz ausgelöst, so wechselt der interaktive Anmeldeprozess unter Aufrecherhaltung des Zustands „Angemeldet“ vom Anwendungs-Desktop zum Winlogon-Desktop und zeigt einen Dialog (Windows Sicherheit) an, mit dem der Benutzer das Kennwort ändern, den Task-Manager ausführen, die Arbeitsstation herunterfahren, sperren oder sich abmelden kann. Meldet sich der Benutzer ab, so schließt der interaktive Anmeldeprozess alle Programme, beendet die Benutzersitzung und

wechselt in den Zustand „Abgemeldet“. Sperrt der Benutzer die Arbeitsstation, so wechselt der interaktive Anmeldeprozess in den Zustand „Gesperrt“.

Im Zustand „Gesperrt“ wird solange der Winlogon-Desktop angezeigt, bis der angemeldete Benutzer die Arbeitsstation entsperrt oder ein Administrator die Abmeldung erzwingt. Entsperrt der angemeldete Benutzer die Arbeitsstation, so wechselt der interaktive Anmeldeprozess vom Zustand „Gesperrt“ zum Zustand „Angemeldet“. Erzwingt ein Administrator die Abmeldung, so wechselt der interaktive Anmeldeprozess vom Zustand „Gesperrt“ in den Zustand „Abgemeldet“.

4.3.2 Grafische Identifikations- und Authentisierungsschnittstelle GINA

Die grafische Identifizierungs- und Authentisierungsschnittstelle (GINA) ist in einer dynamischen Bibliothek implementiert und stellt dem interaktiven Anmeldeprozess WINLOGON.EXE grafische Anmeldedialoge sowie deren Steuerungsfunktionen und Funktionen zur Erzeugung und Steuerung der interaktiven Benutzersitzung bereit.

GINA wird während der Initialisierung des interaktiven Anmeldeprozesses in den Adressraum von WINLOGON.EXE geladen und fängt danach ständig „SAS Nachrichten“ ab. SAS-Nachrichten sind bestimmte im Betriebssystem registrierte Nachrichten, die durch Tastenkombinationen oder die Nutzung von bestimmten Geräten wie Chipkartenlesegeräten ausgelöst werden können.

Nachdem eine SAS-Nachricht von GINA abgefangen wurde, erfolgt eine Verarbeitung und eine Erzeugung einer neuen SAS-Nachricht, die an den interaktiven Anmeldeprozess weitergeleitet wird. Der inaktive Anmeldeprozess WINLOGON.EXE reagiert dann auf die SAS-Nachricht mit der Ausführung einer bestimmten Aktion, wie z.B. den Wechsel in einen neuen Zustand und das damit eventuell verbundene Umschalten der Desktops.

Registrierung im Betriebssystem - Initialisierung

Die Initialisierung von GINA erfolgt während der Initialisierung von Winlogon. Es ist möglich, eine alternative GINA in den interaktiven Anmeldeprozess zu laden. Dazu muss sie im Betriebssystem registriert werden. Die Registrierung erfolgt in der Registrierungsdatenbank unter dem „HKEY_LOCAL_MACHINE-\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon“. Dort muss ein

neuer Zeichenfolgeneintrag mit dem Namen GINADLL eingefügt werden, in dem der Name und das Verzeichnis der neuen GINA enthalten sind.

Schnittstellenfunktionen

GINA verfügt über mehrere Schnittstellenfunktionen, die von Winlogon aufgerufen werden können. Diese sind in [Microsoft3, Platform SDK Documentation\Security\Logon Authentication\Winlogon and Gina\Winlogon and GINA Reference\GINA Export Functions“] detailliert beschrieben. Bei einer Neuimplementation der GINA müssen diese Funktionen größtenteils ersetzt werden.

4.3.3 Authentisierungspakete

Authentisierungspakete dienen der Authentisierung von Benutzern und sind in dynamischen Bibliotheken implementiert. Sie werden im Kontext der lokalen Sicherheitsautorität LSASS.EXE ausgeführt und während der Initialisierung dieser in ihren Adressraum geladen. Windows 2000 verfügt über Authentisierungspakete für die NTLM (MSV1_0.DLL) und Kerberos (KERBEROS.DLL) Authentisierung.

Diese verifizieren die Anmeldedaten eines Benutzers, stellen fest, ob er im System registriert ist und überprüfen die Identität mit einem bestimmten Authentisierungsverfahren. Darüber hinaus erstellen Authentisierungspakete eine neue Sitzung, indem sie eine Sitzungs-ID für den erfolgreich verifizierten Benutzer erstellen und ein Zugriffstoken erzeugen, das den Sicherheitskontext des Benutzers darstellt.

Registrierung im Betriebssystem - Initialisierung

Die Initialisierung von Authentisierungspaketen erfolgt beim Betriebssystemstart während der Initialisierung der lokalen Sicherheitsautorität LSASS.EXE. Diese lädt alle Authentisierungspakete, die unter dem Registrierungsschlüssel „HKEY_LOCAL_MACHINE\System\CurentControlSet\Control\Lsa\“ im Eintrag „Authentication Packages“ aufgeführt sind, in ihren Adressraum. Der Eintrag „Authentication Packages“ ist ein Binärwerteintrag, innerhalb dessen die Namen der einzelnen Authentisierungspakete ohne die Dateinamenserweiterung „.dll“ abgelegt sind. Zwischen den einzelnen Namen der Authentisierungspakete befindet sich die Zeichenkette „\0“, die diese voneinander trennt. Befindet sich das Authentisierungspaket nicht im Verzeichnis „%Systemroot%\System32“, so ist im Eintrag der komplette Pfadname anzugeben, in dem sich das Authentisierungs-

paket befindet. Die Installation neuer Authentisierungspakete erfordert die Registrierung im Eintrag „Authentication Packages“, damit das neue Authentisierungspaket beim nächsten Start des Betriebssystems von der lokalen Sicherheitsautorität geladen wird. Die Verwendung von Authentisierungspaketen erfolgt mit Hilfe der Funktion LsaLogonUser [Microsoft 2000-3, Platform SDK Documentation\Security\Logon Authentication\LSA Authentication\LSA Authentication Reference\LSA Functions\Logon Functions\LsaLogonUser], bei der ein Authentisierungspaket zur Authentisierung ausgewählt werden kann. Obwohl die Registrierung und die Schnittstellenfunktionen von Authentisierungspaketen in [Microsoft 3] relativ ausführlich beschrieben sind, stellt Microsoft im Platform SDK kein Beispiel für ein Authentisierungspaket bereit.

Schnittstellenfunktionen

Ein Authentisierungspaket stellt mit Hilfe von Funktionen bestimmte Dienste bereit, die zur Authentisierung eines Benutzers benötigt werden. Die Schnittstellen der Funktionen sind dokumentiert, sodass es möglich ist, neue Authentisierungspakete zu erstellen. Eine detaillierte Beschreibung der Schnittstellen befindet sich in [Microsoft3, Platform SDK Documentation\Security\Logon Authentication\LSA Authentication\LSA Authentication Reference\LSA Functions\Logon Functions“]. Nachfolgend wird nur auf die Funktion „LsaApLogonUser“ eingegangen, die zur Anmeldung an ein Windows 2000 System dient und die Identität eines Benutzers verifiziert. Sie ist die Implementation der „LsaLogonUser“ Funktion, und wird während der Anmeldung eines Benutzers aufgerufen. In den Parametern der Funktion „LsaLogonUser“ kann ein Authentisierungspaket angegeben werden, mit dem dann eine Verifikation der Identität durchgeführt wird. Dabei kommt dann die Funktion „LsaApLogonUser“ zur Anwendung.

4.3.4 Sicherheitspakete

In Sicherheitspaketen befinden sich Protokolle, die zur Authentisierung im Netzwerk eingesetzt werden. Diese werden mit Hilfe von Sicherheitsunterstützungsanbietern zur Verfügung gestellt, und können von Programmen über die Funktionen der Sicherheitsunterstützungsanbieterschnittstelle (Security Support Provider, SSPI[24]) aufgerufen werden, um die Identität eines Benutzers über das Netzwerk zu verifizieren. In Windows 2000 können Authentisierungspakete und Sicherheitspakete zusammen in eine dynamische Bibliothek integriert werden

24 Eine gute Einführung zum SSPI befindet sich in [Schmidt 2000, S.426]

(AP/SSP). Das hat den Vorteil, das es nur eine Bibliothek gibt, die sowohl von der LSA als auch von Anwendungsprogrammen zur Authentisierung von Benutzern genutzt werden kann.

Registrierung im Betriebssystem

Die Registrierung eines Sicherheitspakets erfolgt in der Registrierungsdatenbank unter dem Registrierungsschlüssel: „HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\“. Dort sind im Werteintrag „Security Packages„. Der Eintrag „Security Packages“ ist ein Binärwerteintrag, innerhalb dessen die Namen der einzelnen Sicherheitspakete ohne die Dateinamenserweiterung „.dll“ abgelegt sind. Zwischen den einzelnen Namen der Sicherheitspakete befindet sich die Zeichenkette „\0“, die diese voneinander trennt. Befindet sich das Authentisierungs- und Sicherheitspaket nicht im Verzeichnis „%Systemroot%\System32“, so ist im Eintrag der komplette Pfadname anzugeben, in dem sich das Sicherheitsspaket befindet. Die Installation neuer Sicherheitsspakete erfordert die Registrierung im Eintrag „Security Packages“, damit das neue Sicherheitspaket beim nächsten Start des Betriebssystems von der lokalen Sicherheitsautorität geladen werden kann.

Schnittstellenfunktionen

Sicherheitspakete müssen bestimmte Schnittstellenfunktionen implementieren. Die Funktionen, die erstellt werden müssen, um ein neues Sicherheitspaket zu erstellen, sind in [Microsoft3, Platform SDK Documentation\Security\Security Packages\Custom Security Package Reference\Custom Security Package Functions“] beschrieben.

4.3.5 Subauthentisierungspakete

Subauthentisierungspakete sind dynamische Bibliotheken, die von Authentisierungspaketen genutzt werden können, um das Authentisierungsverfahren zu erweitern oder zu ersetzen.

Beispielsweise könnte in einem Subauthentisierungspaket eine Funktion eingebaut sein, die prüft, von welcher Arbeitsstation sich ein Benutzer anmeldet. Falls der Benutzer sich von einer falschen Station aus anmeldet, kann der Zugang verweigert werden, obwohl das Passwort richtig gewählt wurde. Weiterhin könnte ebenfalls der Passwortprüfmechanismus durch ein anderes Verfahren ersetzt werden [Microsoft 2000-3, Platform SDK Documentation\Security\Logon

Authentication\LSA Authentication\About LSA Authentication\ Authentication Packages\Subauthentication Packages]. Der Mechanismus zur Benutzung eines Subauthentisierungspakets muss vom Authentisierungspaket zur Verfügung gestellt werden. Die Microsoft Authentisierungspakete für NTLM und Kerberosauthentisierung unterstützen die Einbindung von Subauthentisierungspaketen [Microsoft 2000-3, Platform SDK Documentation\Security\Logon Authentication\LSA Authentication\About LSA Authentication\Useing LSA Authentication\Creating Subauthentication Packages].

Registrierung im Betriebssystem – Initialisierung

Die Nutzung von Subauthentisierungspaketen erfordert die Registrierung im Betriebssystem durch einen Eintrag in der Registrierungsdatenbank. Die Art der Eintragung hängt davon ab, für welches Authentisierungspaket ein Subauthentisierungspaket genutzt werden soll.

NTLM Subauthentisierungspakete (MSV1_0 Subauthentisierungspakete) werden unter dem Schlüssel „HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Control\Lsa\MSV1_0“ registriert.

Kerberos Subauthentisierungspakete werden unter dem Schlüssel „HKEY _LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos“ registriert. Unter diesen Schlüsseln muss ein Werteintrag „AuthN“ für das entsprechende Subauthentisierungspaket erstellt werden, wobei „N“ eine mit dem Subauthentisierungspaket verbundene Nummer ist (z.B. Auth255).

Die Nummer wird während des „LsaLogonUser“ Funktionsaufrufs genutzt, um das entsprechende Subauthentisierungspaket anzusprechen. Subauthentisierungspakete können Nummern im Intervall von 0 bis 255 besitzen. Die Nummer „Null“ ist für den Subauthentisierungsfilter des Domainkontrollers reserviert. Weitere reservierte Nummern sind die Nummern von 1 bis 128, die für Microsoft Authentisierungspakete reserviert sind. Die Nummern von 128 bis 254 können von Softwareanbietern genutzt werden. Diese können bei Microsoft unter der E-Mailadresse subauth@microsoft.com registriert werden[25]. Die Nummer 255 steht

25 Die Registrierung von Subauthentisierungspaketen gewährleistet zwar eine einheitliche Informationsbasis über die vorhanden Subauthentisierungspakete und deren Nummern, kann jedoch nicht verhindern, dass in einem Netzwerk versehentlich zwei unterschiedliche Subauthentisierungspakete mit der gleichen Identifikationsnummer auf unterschiedlichen Servern installiert sind. Dies ist besonders kritisch, wenn die dargestellte Situation auf zwei Domänenkontrollern stattfindet. Domänenkontroller sind Server, auf denen eine Authentisierung im Netzwerk erfolgt. Ein Netzwerk kann über mehrere Domänenkontroller verfügen, um die Ausfallsicherheit der Authentisierungsdienste zu erhöhen und eine Lastverteilung zu ermöglichen. Fällt ein Domänenkontroller aus, so übernimmt ein anderer Domänenkontroller mit einem anderen Subauthentisierungspaket, aber der gleichen Identifikati-

frei zur Verfügung und kann innerhalb eines Netzwerks, in dem keine weiteren Subauthentisierungspakete mit der Nummer 255 vorhanden sind, verwendet werden [Microsoft 2000-3, Platform SDK Documentation \Security\Logon Authentication\LSA Authentication\About LSA Authentication\Useing LSA Authentication\Registering Custom Authentication Packages\Registering Subauthentication Packages].

Schnittstellenfunktionen

Subauthentisierungspakete verfügen über zwei Schnittstellenfunktionen, MSV1_0SubAuthenticationRoutine und MSV1_0SubAuthenticationFilter.

MSV1_0SubAuthenticationRoutine: Die Funktion MSV1_0SubAuthenticationRoutine wird nur bei nicht-interaktiven Authentisierungen auf dem Server aufgerufen, auf dem sich das Benutzerkonto befindet. Darüber hinaus wird die Funktion nur vom MSV1_0 Authentisierungpaket benutzt. Kerberos Authentisierungspakete rufen diese Funktion nicht auf. [Microsoft 2000-3, Platform SDK Documentation\Security \Logon Authentication\LSA Authentication\LSA Authentication Reference\LSA Functions\Subauthentication Functions\MSV1_0SubAuthenticationRoutine]. Innerhalb dieser Funktion kann der MSV1_0 Authentisierungsmechanismus verändert werden, da das Authentisierungspaket das Subauthentisierungspaket für die Authentisierung benutzt [Microsoft 2000-3, Platform SDK Samples – Readme zu Subauth Sample].

MSV1_0SubAuthenticationFilter: Die Funktion MSV1_0SubAuthenticationFilter wird nach einer Authentisierung vom Authentisierungspaket aufgerufen und kann sowohl vom MSV1_0 Authentisierungspaket als auch vom Kerberos Authentisierungspaket genutzt werden [Microsoft 2000-3, Platform SDK Documentation\Security \Logon Authentication\LSA Authentication\LSA Authentication Reference\LSA Functions\Subauthentication Functions\MSV1_0SubAuthenticationFilter]. Die Funktion wird nur für Subauthentisierungspakete der Nummer Null (Auth0) aufgerufen [Microsoft 2000-3, Platform SDK Samples – Readme zu Subauth Sample]. Hier können erweiterte Prüfungen wie die Anmeldezeit geprüft werden.

---

onsnummer, die Authentisierung. Dadurch kann der Authentisierungsvorgang abgebrochen werden, obwohl die richtigen Merkmale zur Authentisierung übermittelt wurden.

4.4 Authentisierung im Netzwerk

Die Authentisierung im Netzwerk findet in Windows 2000 Umgebungen mit Hilfe von Domänenkontrollern statt. Diese stellen in der allgemeinen Architektur eines Client-Server-Authentisierungssystems die Authentisierungsserver dar und führen Dienste aus, mit denen eine Authentisierung im Netzwerk ermöglicht wird. Darüber hinaus befindet sich auf Domänenkontrollern auch die Benutzerkontendatenbank, mit Hilfe der die Verifikation der Identität eines Benutzers erfolgt.

4.4.1 NTLM Authentisierung

Die NTLM[26] Authentisierung ist ein Authentisierungsmechanismus, der in früheren Versionen von Windows 2000 (Windows NT 4.0, Windows NT 3.51) als primäre Authentisierung eingesetzt wurde. Windows 2000 unterstützt die NTLM Authentisierung, um in Netzwerken aus Windows 2000 und Windows NT Computersystemen eine Authentisierung zu ermöglichen.

Dabei kommt eine NTLM Authentisierung immer dann zum Einsatz, wenn die die Authentisierung vornehmende Instanz, die Authentisierung anfordernde Instanz oder beide auf Windows NT 4.0 (oder einer früheren Version) basieren. Darüber hinaus wird die NTLM Authentisierung von Windows 2000 Clienten verwendet, die keiner Domäne angehören.

Bei der Anmeldung am Betriebssystem mit Hilfe von NTLM werden die Anmeldeinformationen mittels Winlogon und GINA erfasst und an die lokale Sicherheitsautorität übermittelt (Abbildung 17 - 1).

Diese leitet aus dem Kennwort einen Schlüssel ab und übermittelt eine Authentisierungsanfrage an einen Domänenkontroller[27]. Der Domänenkontroller erzeugt eine Zufallszahl und sendet diese an die lokale Sicherheitsautorität des Clients zurück.

Diese modifiziert sie mit einem bekannten Verfahren (response), verschlüsselt sie und sendet sie zurück an den Domänenkontroller. Der Domänenkontroller prüft die Modifikation und sendet bei erfolgreicher Prüfung einen Identifikator und die

26 Die Abkürzung NTLM steht für NT LAN Manager.

27 Die Anfragen an den Domänenkontroller werden über einen sicheren Kanal übermittelt, den Netlogon zur Verfügung stellt.

Gruppenzugehörigkeiten zurück zur lokalen Sicherheitsautorität des Client (Abbildung 17 4-7).

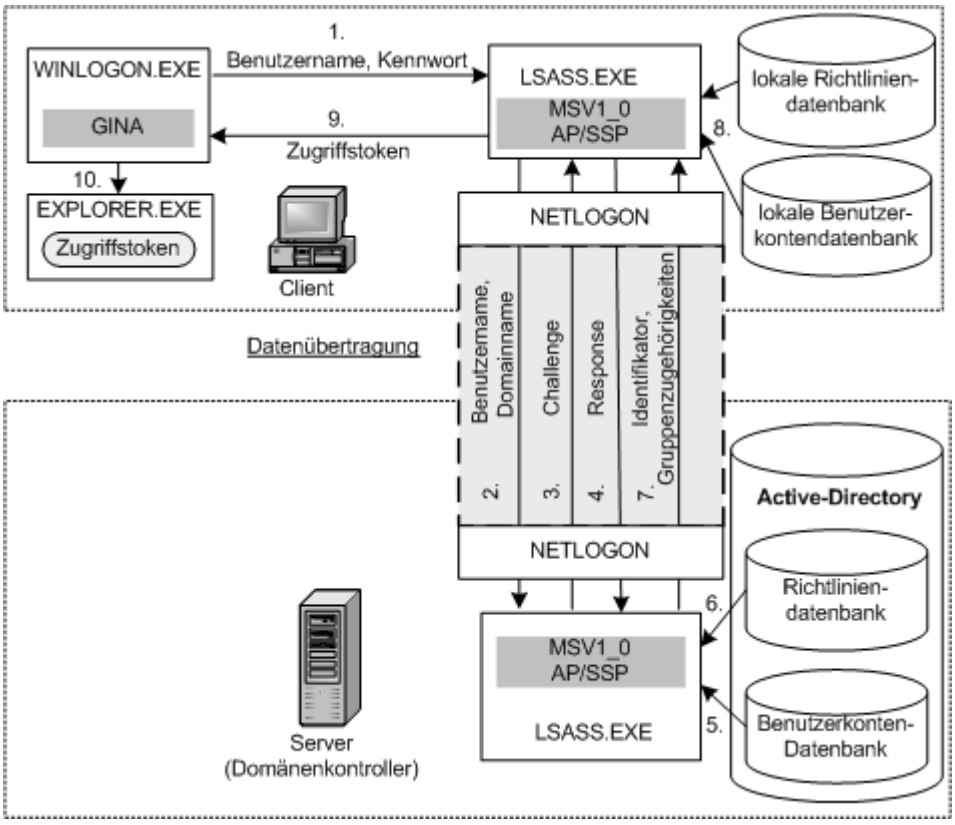

Abbildung 17: Windwos 2000 NTLM Authentisierung

Diese erzeugt aus den Informationen und den lokalen Richtlinien und Gruppenzugehörigkeiten ein Zugriffstoken, das an den interaktiven Anmeldeprozess WINLOGON.EXE übergeben wird (Abbildung 17 -9). Nach Erhalt des Zugriffstokens, startet WINLOGON.EXE die grafische Benutzershell (EXPLORER.EXE) und bindet dabei das Zugriffstoken an den Benutzerprozess (Abbildung 17 -10).

4.4.2 Kerberos Authentisierung

Windows 2000 unterstützt eine Authentisierung nach dem Kerberos V5 Protokoll. Das Kerberos Authentisierungssystem wurde dabei vollständig in die Windows 2000 Betriebssystemarchitektur integriert, wobei die Dienstkompo-

nenten wie der Authentisierungsdienst und der Ticket Granting Dienst nur in den Server Versionen des Windows 2000 Betriebssystems zur Verfügung gestellt werden. Authentisierungs- und Ticket Granting Dienst werden im Kontext der lokalen Sicherheitsautorität ausgeführt und nutzen den Windows 2000 Verzeichnisdienst (Active Directory) als Kerberos Datenbank. Das Kerberos-Protokoll ist in einem Sicherheitspaket (SSP) implementiert und kann von Systemdiensten und Anwendungen über die SSPI Schnittstelle zur Authentisierung genutzt werden.

Anmeldung am Betriebssystem mit Kerberosauthentisierung

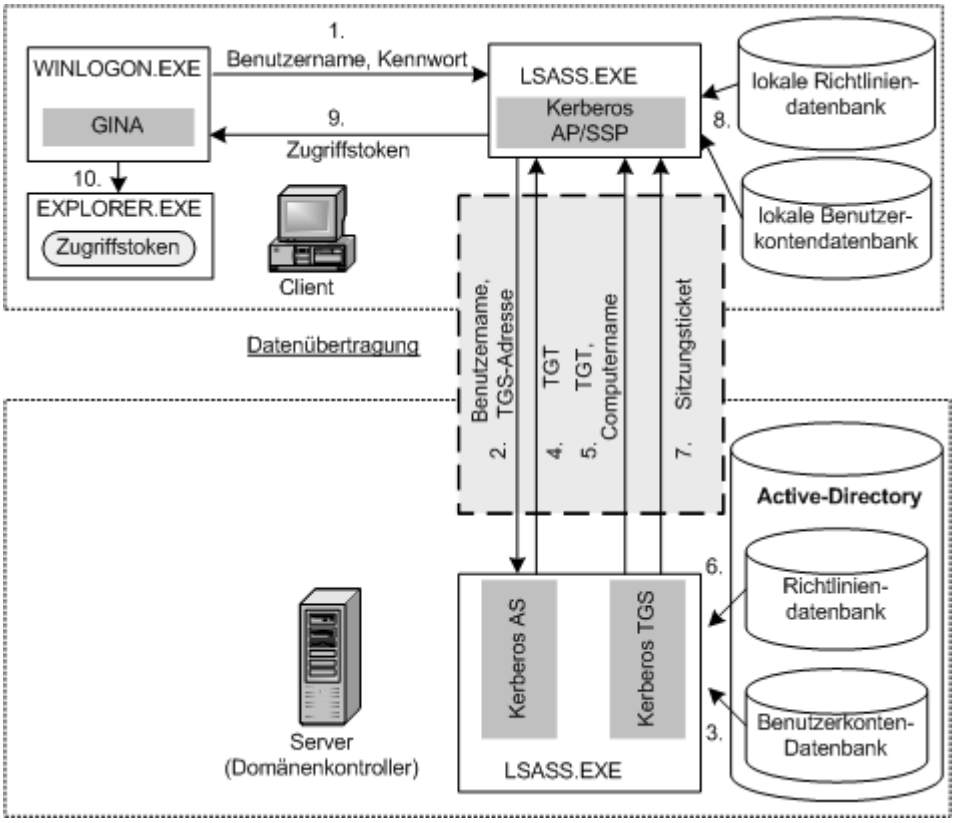

Abbildung 18: Kerberos-Authentisierung in Windows 2000

Bei der Anmeldung am Betriebssystem mit Hilfe von Kerberos werden die Anmeldeinformationen mittels Winlogon und GINA erfasst und an die lokale Sicherheitsautorität übermittelt (Abbildung 18 - 1) .
Diese benutzt das Kerberos AP/SSP, um in einem AS-Austausch ein TGT-Ticket vom Kerberos-Authentisierungsserver zu beziehen (Abbildung 18 - 2 bis 4). Das TGT-Ticket ermöglicht den Bezug weiterer Tickets von einem TGT-Server und ist mit einem vom Kennwort abgeleiteten Schlüssel verschlüsselt. Die Fähigkeit

das TGT-Ticket erfolgreich zu entschlüsseln, authentisiert den Benutzer und versetzt ihn in die Lage das TGT-Ticket zum Bezug weiterer Tickets zu nutzen.

Nach Empfang und erfolgreicher Entschlüsselung des TGT-Tickets fordert die LSA beim Ticket-Granting-Server ein Sitzungsticket für den Computer an, an dem sich der Benutzer anmeldet (Abbildung 18 5- 7). Danach erzeugt sie mit Hilfe des Sitzungstickets ein Zugriffstoken und übergibt es an WINLOGON.EXE (Abbildung 18 - 9). Die Erzeugung des Zugriffstokens bezieht lokale Computerrichtlinien und Gruppenzugehörigkeiten ein, die aus der lokalen Richtlinien und Benutzerkontendatenbank bezogen werden (Abbildung 18 – 8).

Nach Erhalt des Zugriffstokens, startet WINLOGON.EXE die grafische Benutzershell (EXPLORER.EXE) und bindet dabei das Zugriffstoken an den Benutzerprozess (Abbildung 18 -10).

Kapitel 5

Konzeption

Dieses Kapitel beschreibt die Erarbeitung von allgemeinen Konzepten zur Implementierung eines Iris-Biometrik-Verfahrens in Client-Server-Authentisierungssysteme.

Dazu werden im Abschnitt 5.1 Anforderungen definiert, die als Grundlage zur Erstellung von Konzepten in den nachfolgenden Abschnitten dienen.

Im Abschnitt 5.2 „Grobkonzept“ werden die Phasen der biometrischen Identitätsbestimmung, den Phasen der Identitätsbestimmung in Computersystemen zugeordnet und danach als Funktionseinheiten zur biometrischen Authentisierung betrachtet. Darauf aufbauend wird ein Grobkonzept erstellt, das beschreibt, welche Funktionseinheiten zur biometrischen Identitätsbestimmung in einem Client-Server-Authentisierungssystem auf dem Client und auf dem Server ausgeführt werden sollen.

Anhand des im Abschnitt 5.2 erstellten Grobkonzepts wird im Abschnitt 5.3 ein Modularisierungskonzept erarbeitet, das die einzelnen Funktionseinheiten Modulen zuordnet, deren Zusammenwirken und Schnittstellenfunktionen beschreibt.

Auf Grundlage des im Abschnitt 5.3 erstellten Modularisierungskonzepts werden im Abschnitt 5.4 Konzepte zur Implementierung in bestehende Systeme erarbeitet, wobei hier auf die Implementierung durch Vorschaltung und Verkettung und auf die Implementierung durch Einbettung und Integration eingegangen wird.

5.1 Anforderungen

5.1.1 Kennwortersatz durch Iris?

Bei der Euphorie über die Iris als biometrisches Merkmal zur Identitätsbestimmung stellt sich die Frage, inwieweit die Iris das Kennwort als Authentisierungsmerkmal ersetzen kann. Eine Ersetzung wäre nicht nur für den Benutzer attraktiv, der sich nun sein Kennwort nicht mehr merken müsste, sondern würde darüber hinaus die Administrationskosten einsparen, die bei der Vergabe neuer Kennwörter entstehen.

Die Iris ist ein physiologisches biometrisches Merkmal und die Muster, die verwendet werden, sind nach ihrer entgültigen Entwicklung konstant. Tönnesen kritisiert in [Tönnesen 1999], dass sich bei physiologischen biometrischen Verfahren durch die Konstanz und das begrenzte Vorhandensein des biometrischen Merkmals nur eine begrenzte Anzahl biometrischer Templates erzeugen läßt.

In einem technischen System, in dem zur Erzeugung des biometrischen Templates nur die biometrischen Samples als Eingangsdaten vorkommen, kann es somit nur zwei verschiedene biometrische Templates als Referenzmuster geben.

Sollte es einem Angreifer durch die Imitation des biometrischen Merkmals gelingen in ein System einzudringen, so müsste nach Bekanntwerden des Angriffs immer wieder ein neues biometrisches Template erstellt werden. Der Mensch besitzt allerdings nur zwei Iriden. Nach dem zweiten erfolgreichen Angriff kann kein unbekanntes Template mehr erstellt werden, falls als Eingangsdaten nur das biometrische Sample verwendet wird.

Darum sollten zur Authentisierung mittels Iris-Biometrik immer zusätzliche Mechanismen eingesetzt werden. Hier kann das Iris-Biometrik-Verfahren durch Authentisierungsverfahren ergänzt werden, die personenbezogene Merkmale wie Wissen oder Besitz benutzen.

Denkbar wäre es, die Templaterzeugung direkt abhängig von Kennwörtern oder kryptografischen Schlüsseln zu gestalten, sodass als Eingangsdaten zur Erzeugung eines Templates das biometrische Sample und ein Kennwort oder Schlüssel notwendig ist. Nach einem Angriff könnte durch einen Wechsel des Kennworts oder Schlüssels ein neues benutzerspezifisches biometrisches Template der gleichen Iris erzeugt werden.

Damit verbunden würde allerdings ein Grundsatzproblem auftreten. Bei einer fehlgeschlagenen Authentisierung könnte nicht mehr entschieden werden, ob die Authentisierung wegen eines falschen Kennworts, Schlüssels oder der fehlgeschlagenen Verifikation einer Iris entstanden ist, da das Kennwort nicht mehr extra verifiziert wird. Dieser Umstand kann nur durch ein zusätzliches Abspeichern des Kennworts und dessen Verifikation im Authentisierungsvorgang und einer Rückmeldung bei einer fehlgeschlagenen Kennwortverifikation behoben werden. Dann besteht allerdings wieder die Gefahr, dass ein Angreifer die eventuellen technischen Schwächen des Systems ausnutzt und durch „Brute Force Angriffe“ das Kennwort knackt.

Als alternativer Mechanismus zur Verwendung von Kennwörtern und Schlüsseln könnte die Integration von Verfahren zur Lebenderkennung während der Authentisierung sein. Durch das zufällige Aussenden von Lichtreizen könnte die Größenänderung der Pupille beobachtet werden. Darüber hinaus wäre es möglich durch bestimmte Optiken zufällig in Regionen der Iris Reflektionen zu erzeugen und so zu prüfen, ob eine Person tatsächlich an der Mensch-Sensorschnittstelle anwesend ist.

Die Entwicklung von Mechanismen zur Lebenderkennung ist noch Forschungsgegenstand und nicht abgeschlossen. Die Implementation der Iris-Biometrik wird daher durch ein Kennwort ergänzt.

5.1.2 Datenschutz und Datensicherheit

Bei den biometrischen Eingabe- und Referenzdaten der Iris-Biometrik handelt es sich um Daten eines physiologischen biometrischen Verfahrens, bei denen eine dauerhafte Bindung zwischen den biometrischen Merkmalsdaten und den dazu gehörigen Personen besteht.

Dadurch ist eine Missbrauchsgefahr über einen langen Zeitraum gegeben, die sich nicht nur auf die Sicherheit des biometrischen Authentisierungssystems, sondern auch auf die informationelle Selbstbestimmung von Personen auswirken kann, deren biometrische Daten verarbeitet werden.

Die unbemerkte Erfassung eines biometrischen Merkmals eignet sich zur Überwachung und Verfolgung von Personen [Köhntopp 1999, S.182], da bei Kenntnis von Personen und biometrischen Daten jederzeit ein Rückschluss auf die Identität einer Person möglich ist.

Darum sollte systemtechnisch verhindert werden, dass ohne Kenntnis der betroffenen Personen eine Erfassung und Vermessung der Iris stattfinden kann.

Die biometrischen Eingabe- und Referenzdaten sollten nach dem Grundsatz der Datenvermeidung und Datensparsamkeit gespeichert werden und keinen überflüssigen Informationsgehalt enthalten.

Aus den gespeicherten biometrischen Daten dürfen keine Rückschlüsse auf die hinter den biometrischen Daten stehende, natürliche Person gezogen werden [Laßmann 1999, S.136]. Sonst wäre es durch die Zusammenführung von Datenbeständen möglich Benutzerprofile zu erstellen, die ohne Einwilligung der Personen für kommerzielle Zwecke genutzt werden. Die biometrischen Templates stellten dann eine Art Beweis dar, dass eine Person das biometrische Authentisierungssystem eines bestimmten Dienstes nutzt.

Nach Möglichkeit sollten die biometrischen Daten im Verfügungsbereich des Nutzers gespeichert werden. Dieses kann zum Beispiel durch Speicherung auf einer Chipkarte geschehen, die der Benutzer bei sich führt.

Bei einer zentralen Speicherung sollte eine Anonymisierung oder Pseudonymisierung erfolgen bzw. die Daten nach Gebrauch gelöscht werden.

Alle zur Bestimmung der Identität notwendigen Daten sind nach den Grundsätzen der Vertraulichkeit, Integrität und Authentizität zu verarbeiten und zu speichern. Vertraulichkeit bedeutet hier die Eigenschaft, dass die Daten nicht von unbefugten Personen gelesen und interpretiert werden können. Die Eigenschaft der Integrität bezeichnet die Fähigkeit Manipulationen an den Daten zu erkennen. Mit der Authentizitätseigenschaft wird gewährleistet, dass überprüft werden kann, ob die Daten einer bestimmten Person zuzuordnen sind oder von einem bestimmten Dienst im Netzwerk versendet wurden.

5.1.3 Technische Anforderungen

Flexibilität, Erweiterbarkeit und Modularisierung

Die Implementierung sollte mit Hilfe mehrerer Komponenten erfolgen, deren Schnittstellen so definiert sind, dass es zu einem späteren Zeitpunkt möglich ist diese durch neue Versionen auszutauschen, um einen hohen Grad an Flexibilität und Erweiterbarkeit zu gewährleisten.

Modularisierung

Flexibilität und Erweiterbarkeit sollten durch eine starke Modularisierung mit Hilfe von Bibliotheken erreicht werden. Durch öffentliche Schnittstellen der einzelnen Module wird die interne Arbeitsweise der Komponenten nach außen verborgen. Darüber hinaus sollten die Schnittstellen nicht abhängig von einem bestimmten System sein, das zur Erfüllung einer Aufgabe innerhalb eines Moduls genutzt wird.

Integration

Die Implementierung des Iris-Biometrik-Verfahrens sollte so erfolgen, dass möglichst wenig Änderungen im nichtbiometrischen Authentisierungssystem durchzuführen sind. Weiterhin sollte die Implementierung die vorhandene Zuverlässigkeit, Robustheit und Leistungsfähigkeit des Systems nicht einschränken.

Implementierung von Standards

Wenn möglich, sollte immer eine Implementierung und Integration nach den Vorgaben vorhandener Standards erfolgen, um ein Zusammenarbeiten von Komponenten verschiedener Hersteller zu ermöglichen. Im Bereich der Biometrik hat sich allerdings noch kein einheitlicher Standard herausgebildet. Es gibt mehrere miteinander konkurrierende Standards, wie SVAPI (Speaker Verification API)[28], BAPI (Biometric API)[29], BioAPI (Biometric API)[30], HA-API (Human Authentication API), CDSA/UAS (Common Data Security Architekture / User Authentication System)[31]. Da sich noch kein endgültiger Standard für biometrische Systeme herausgebildet hat, erfolgt die Implementierung unabhängig von exisitierenden Biometrik Standards.

5.2 Grobkonzept

Im Grobkonzept erfolgt die Festlegung von allgemeinen Richtlinien, anhand derer ein Architekturentwurf zur Implementierung des Iris-Biometrik-Verfahrens erfolgt. Dabei wird von der allgemeinen Architektur eines Client-Server-Authen-

28 www.srapi.com/svapi

29 www.iosoftware.com

30 www.bioapi.org

31 www.biometrics.org

tisierungssystems ausgegangen und anhand von gedanklichen Implementierungsbeispielen festgestellt, welche Variante am Sinnvollsten ist.

Der allgemeine Ablauf zur Bestimmung einer Identität in Computersystemen besteht aus vier Phasen: Datenaufnahme, Vorverarbeitung, Identifikation und Authentisierung. Der allgemeine Ablauf eines biometrischen Verfahrens, basierend auf [Zhang 2000], besteht aus nur drei Phasen: Datenaufnahme, Merkmalsextraktion und Vergleich.

In der Vergleichsphase findet allerdings auch eine Identifikation und Verifikation (Authentisierung) statt, sodass der allgemeine Ablauf eines biometrischen Verfahrens, basierend auf [Zhang 2000], ebenfalls als vierphasig betrachtet werden kann (Datenaufnahme, Merkmalsextraktion, Identifikation, Authentisierung). Weiterhin können die Identifikationsphase und die Authentisierungsphase im Ablauf zur Bestimmung der Identität in Computersystemen mit nichtbiometrischen Verfahren zu einer Vergleichsphase zusammengefasst werden, da dort ebenfalls Vergleiche stattfinden. Diese Phasen können als äquivalent betrachtet werden, auch wenn die Algorithmen zum Vergleich sich unterscheiden. Das biometrische Verfahren zum Vergleich zweier biometrischer Templates kann in die Vergleichsphase des nichtbiometrischen Verfahrens integriert werden. Die Vorverarbeitungsphase im allgemeinen Ablauf zur Bestimmung der Identität in Computersystemen mit nichbiometrischen Verfahren dient der Aufbereitung der Daten zur Identifikation und Authentisierung (Vergleichsphase).

Die Phase der Merkmalsextraktion, in der ein biometrisches Template erzeugt wird, kann ebenfalls als Phase zur Datenaufbereitung für die Vergleichsphase betrachtet werden. Das biometrische Verfahren zur Erzeugung des biometrischen Template kann somit in die Vorverarbeitungsphase integriert werden. Die Phase der Merkmalsextraktion wird im Weiteren als Phase der Templateerzeugung bezeichnet. Die Phasen der Datenaufnahme erfüllen in beiden Abläufen den gleichen Zweck und dienen der Erfassung der Identifikations- und Authentisierungsmerkmale. In den einzelnen Phasen erfolgt keine Zustandsspeicherung.

Die Daten werden von einer Phase zur anderen weitergereicht und können mit Hilfe von Funktionseinheiten realisiert werden. In einem Client-Server-Authentisierungssystem können diese entweder auf dem Client oder auf dem Server ausgeführt werden. Die nachfolgenden Überlegungen sollen helfen, zu entscheiden, welche Funktionseinheiten auf dem Client und dem Server ausgeführt werden sollten. Darüber hinaus sollen sie Überlegungen und Anregungen zur Implementation der Datenaufnahme, Erzeugung des biometrischen Template, Datenübertragung, Speicherung der Referenztemplates und dem Vergleich zweier biometrischer Templates darstellen.

5.2.1 Datenaufnahme - Erfassung von biometrischen Merkmalen

Während der Erfassung von biometrischen Merkmalen wird mit Hilfe eines Sensors ein digitales Abbild (Biometrisches Sample) des biometrischen Merkmals erzeugt, das als Grundlage zur Erstellung einer biometrischen Kenngröße in Form eines biometrischen Template dient.

Das biometrische Sample kann dabei auch Merkmale enthalten, die nicht zur weiteren Verarbeitung genutzt werden. So wird bei der Iris-Biometrik ein digitales Abbild der Augenoberfläche erzeugt, das neben der Iris ebenfalls die Pupille und die Sklera enthält.

Der Erfassungsvorgang findet vor der Referenzbildung (Einlernphase) und vor jeder biometrischen Identitätsbestimmung (Identifikation, Authentisierung) statt. Vorraussetzung ist hierbei die richtige Positionierung des Auges vor dem Sensor (Digitalkamera), um die Iris unter idealen Bedingungen digital zu erfassen, die Anzahl weiterer Erfassungsversuche zu verringern und damit den Anforderungen an Benutzerfreundlichkeit und Effizienz gerecht zu werden.

Die grafischen Dialoge zur Erfassung der Iris enthalten deshalb ein Positionierungsfenster, in dem ein Benutzer die Positionierung des Auges vor der Kamera selbst beobachten kann. Darüber hinaus wird der Benutzer mit Hilfe von grafischen oder akkustischen Signalen über eine zur Erfassung geeignete Positionierung und den Erfassungsvorgang informiert, damit er die Erfassungsposition während der Erfassungsphase aufrechterhält und eine qualitativ hochwertige Erfassung des Auges und somit der Iris gewährleistet wird. Durch die Eingabe des Benutzernamens und das Einleiten des Erfassungsvorgangs erfolgt die Erfassung der Iris als biometrisches Merkmal mit der Kenntnis und Zustimmung des betroffenen Benutzers. Ferner ist bei der Erstellung der Dialoge zur Erfassung biometrischer Merkmale darauf zu achten, vorhandene Zugriffsschutzmechanismen auf Dialogfenstern zu nutzen, damit der Zugriff anderer Prozesse auf den Inhalt der Anmeldedialoge, wie den Eingabefeldern für Benutzername und Kennwort und dem Positionierungsfenster, nicht möglich ist. Der Erfassungsvorgang findet auf dem Client statt.

5.2.2 Erzeugung eines biometrischen Template

Bei der Erzeugung eines biometrischen Template wird aus einem biometrischen Sample ein Template erstellt, das als Kenngröße zur Identifikation und Authentisierung einer Person dient. Der Erzeugungsvorgang ist dabei unabhängig von externen Geräten oder gemeinsamen Datenbasen und bedingt nur das Vorhandensein eines digital biometrischen Sample des biometrischen Merkmals, sodass ein biometrisches Template zentral auf einem Server oder dezentral auf einem Clienten erzeugt werden kann.

Erzeugung des biometrischen Template auf einem Server

Eine zentrale Erzeugung eines biometrischen Template auf einem Server erfordert die Übertragung des zuvor erstellten biometrischen Sample. Das biometrische Sample weist bei der Iris-Biometrik ein hohes Datenvolumen auf, da es sich hier um ein digitales Bild der Augenoberfläche mit hoher Auflösung handelt. Dadurch kann bei der Übertragung im Netzwerk ein hoher Datenverkehr auftreten. Bei der gleichzeitigen Übertragung mehrerer biometrischer Samples zum Server kann es zu Engpässen im Netzwerk kommen, die sich negativ auf die Effizienz- anforderungen auswirken.

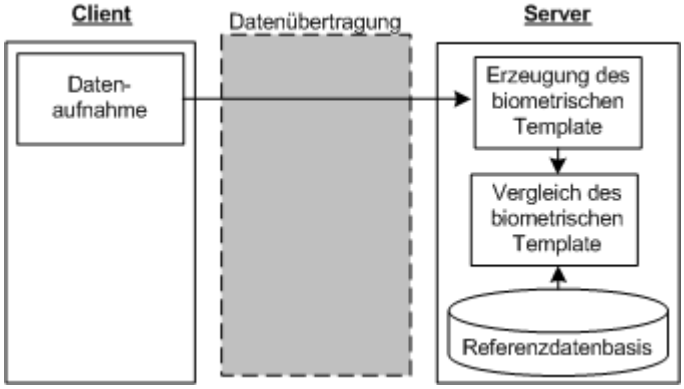

Abbildung 19: Zentrale Erzeugung des biometrischen Template

Darüber hinaus kann die Erzeugung eines biometrischen Iristemplate viel Rechenleistung in Anspruch nehmen. Eine gleichzeitige Erzeugung mehrerer Iristemplates kann teure Hochleistungscomputer erfordern, um die Verfügbarkeit des Servers aufrechtzuerhalten.

Dezentrale Erzeugung des biometrischen Template

Bei der dezentralen Erzeugung eines biometrischen Template wird dieses direkt nach der Erfassung des biometrischen Merkmals auf dem Client erstellt. Eine Datenübertragung des zuvor erzeugten Iris-Samples über ein Netzwerk ist nicht notwendig. Somit tritt bei der dezentralen Erzeugung des biometrischen Template kein Datenverkehr auf.

Allerdings kann eine nachfolgende Datenübertragung des biometrischen Template erfolgen, falls ein zentraler Vergleich des biometrischen Template mit einem Referenztemplate stattfindet. Ein Iristemplate besitzt jedoch gegenüber einem digitalen Irisabbild mit hoher Auflösung ein massiv reduziertes Datenvolumen. Daugman schreibt in [Daugman 1998], dass zur Erzeugung eines Iristemplates Bilder mit einer Auflösung von 640 x 480 Bildpunkten geeignet sind. Ein BMP-Bild dieser Größe mit einer Farbtiefe von 256 Farben hätte eine Größe von 302 Kbyte. Der Phasenvektor eines Iristemplate ist im Gegensatz dazu nur 256 Bytes groß. Iristemplates sind mit ihrem Datenvolumen besser für eine Übertragung geeignet. Die Gefahr von Engpässen im Netzwerk, verursacht durch die gleichzeitige Übertragung mehrerer Iristemplates, sinkt.

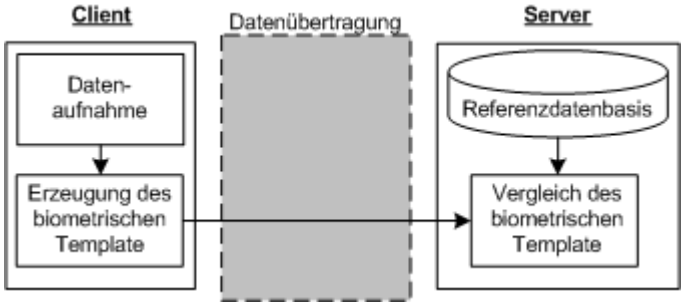

Abbildung 20: Dezentrale Erzeugung des biometrischen Template

Darüber hinaus entsteht der Rechenaufwand zur Erstellung eines biometrischen Template auf den einzelnen Clienten. Die Wahrscheinlichkeit, dass ein Clientsystem durch eine nichtausreichende Rechenleistung ausfällt, ist somit sehr gering, da sich am Client in der Regel nur eine Person zur Zeit anmelden kann. Die Erzeugung des Iristemplate auf dem Client ist somit eine bessere Lösung und wird bevorzugt.

5.2.3 Übertragung von Daten zur Authentisierung

In klassischen Verfahren zur Authentisierung, wie Challenge-Response oder Kerberos, erfolgt keine Übertragung von Kennwörten über das Netzwerk[32]. Stattdessen werden von Kennwörtern Schlüssel abgeleitet, mit denen Zufallszahlen und Zeitstempel verschlüsselt werden. Die Fähigkeit zu deren Entschlüsselung setzt die Kenntnis eines gemeinsamen Geheimnisses voraus und dient als Nachweis der Identität.

Die mit biometrischen Verfahren verbundenen biometrischen Templates sind aber nicht als kryptografische Schlüssel geeignet, da ein biometrisches Template durch unterschiedliche Positionierung des Merkmals vor dem Sensor oder unterschiedlicher optischer Verhältnisse geringfügig variieren kann.

Ein nur geringfügig anderer Schlüssel hat aber einen anderen verschlüsselten Nachrichteninhalt zur Folge, aus dem nicht mehr der ursprüngliche Nachrichteninhalt gefolgert werden kann. Da das bei der Identitätsbestimmung erzeugte biometrische Template geringfügig vom Referenztemplate abweichen kann, besteht kein identisches gemeinsames Geheimnis mehr, aus dem ein identischer Schlüssel zur Ver- und Entschlüsselung abgeleitet werden kann.

Folglich eignet sich das biometrische Template in der dargestellten erzeugten Form nicht zur Ableitung eines kryptografischen Schlüssels. Die Protokolle zur Authentisierung im Netzwerk, wie Challenge-Response oder Kerberos, können in der bestehenden Form nicht verwendet und müssen ergänzt oder ersetzt werden.

Dazu müsste eine Nachbearbeitung erfolgen, bei der ähnliche Templates auf eine gemeinsame Kenngröße abgebildet werden, die als Basis zur Ableitung eines kryptografischen Schlüssels dient. Hier stellt sich jedoch die Frage, inwieweit der kryptografische Schlüssel verwendet werden darf, denn die Sicherheit eines kryptografischen Verfahrens basiert auf der Geheimhaltung des kryptografischen Schlüssels. Das biometrische Merkmal, aus dem der Schlüssel abgeleitet wird, ist jedoch öffentlich und kann jederzeit an jedem Ort aufgenommen werden. Mit der Kenntnis der Algorithmen zur Erstellung eines biometrischen Template, der Nachbearbeitung und der Ableitung eines kryptografischen Schlüssels hätte ein Angreifer die Möglichkeit einen verschlüsselten Datenverkehr zur Authentisierung mit dem Authentisierungsserver aufzubauen und eine andere Identität anzunehmen. Aus Sicht der Kryptografie wäre diese Vorgehensweise so nicht akzeptierbar.

32 Die Funktionsweise der Kerberosauthentisierung wird m Kapitel 3 erläutert.

Die Übertragung der Daten sollte in einem geschützten Kanal stattfinden, der nach den Regeln eines extra dafür entwickelten Protokolls abgesichert wird. Hierfür gibt es kryptografische Protokolle, bei denen während der Initialisierung ein Sitzungsschlüssel ausgehandelt wird, mit dem eine Verschlüsselung des nachfolgenden Datenverkehrs stattfindet [Kronberg 1999, S.89]. Im Zusammenhang mit der Authentisierung im Netzwerk durch ein Iris-Biometrik-Verfahren wird am Arbeitsbereich AGN des Fachbereichs Informatik an einem neuen Protokoll gearbeitet, das auch die Lebenderkennung mit einbezieht.

5.2.4 Speicherung biometrischer Referenztemplates

In den dargestellten Client-Server-Authentisierungssystemen erfolgt die Speicherung der Referenzdaten zur Authentisierung (Kennwörter) in einer zentralen Datenbank. Diese enthält Benutzerkonten, in denen sich die zu den einzelnen Benutzern dazugehörigen Referenzdaten befinden und auf deren Inhalt mit Hilfe des dazugehörigen Benutzerkontennamens zugegriffen werden kann. Die Datenbanken von verschiedenen Client-Server-Authentisierungssystemen unterstützen unterschiedliche Standards, um die gespeicherten Daten zu schützen und die Integrität dieser sicherzustellen.

Darum sollten die biometrischen Referenztemplates generell verschlüsselt und signiert gespeichert werden. Darüber hinaus sollte, wenn möglich, eine Anonymisierung der Identitätsinformation erfolgen, damit nicht aus dem Datensatz auf die hinter den Referenzdaten stehende Person geschlossen werden kann.

Die Operationen zur Verwaltung der gespeicherten Daten und das Format, indem die Daten abgelegt werden, sollte mit Hilfe eines extra Moduls (Referenzdatenbankmanagementmodul REFDBM) in Form einer dynamischen Bibliothek erfolgen. Dadurch kann, wenn notwendig, ein Wechsel der Datenbank als Speichermedium mit wenig Aufwand realisiert werden, da nur die Bibliothek und Export- und Importprogramme neu implementiert werden müssen. Durch das zur Verfügung stellen mehrerer, verschiedener Implementationen des Referenzdatenbankmanagementmoduls ist es dann auch möglich, die Daten in verschiedenen Referenzdatenbanken abzulegen.

Darüber hinaus könnte das REFDBM-Modul mit weiteren Datenbanken während der Authentisierungsphase kommunizieren. Denkbar wäre hier eine zentrale Datenbank mit biometrischen Referenzdaten, die zum Auffinden von Terroristen dient, wie in [Brömme 2002] beschrieben.

5.2.5 Vergleich eines biometrischen Template

Der Vergleich eines biometrischen Template kann auf einem Client oder Server stattfinden. Im Gegensatz zur Erzeugung eines biometrischen Template ist der Vergleich immer abhängig von einer Datenbasis, die Referenzsignaturen enthält.

Vergleich biometrischer Templates auf dem Client

Findet ein Vergleich des biometrischen Template mit einem Referenztemplate auf dem Client statt, so muss eine Übertragung des Referenztemplate von der zentralen Datenbasis zum Clienten erfolgen. Der Server wird in dieser Architektur nur benutzt, um auf die Referenzdatenbasis zuzugreifen. Der Vergleich biometrischer Templates mit Referenztemplates auf dem Client ist nicht zu empfehlen, da hier mit gezielten Falschanfragen die Referenztemplates von Nutzern ausgespäht werden können, da auf jeden Fall zum Vergleich ein Referenztemplate übertragen werden muss. So könnte ein Angreifer bequem durch Anfragen die gesamte Referenzdatenbank auslesen.

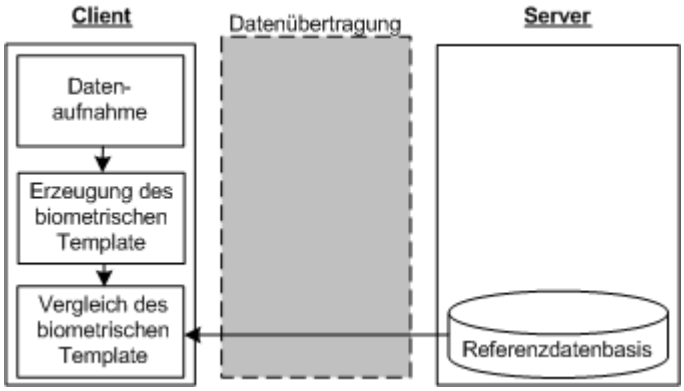

Abbildung 21: Dezentraler Vergleich des biometrischen Template

Die Anforderungen an den Datenschutz und die Datensicherheit sind nur ungenügend erfüllt. Der Rückschluss auf die hinter dem biometrischen Referenztemplate stehende natürliche Person ist durchaus möglich und somit der Grundsatz der Vertraulichkeit aufgehoben.

Darüber hinaus kann eine Manipulation des Vergleichvorgangs auf einem Clienten leichter erfolgen als auf einem speziell gesicherten dedizierten Server, da ein Clientrechner unter Umständen nicht über Sicherheitsmechanismen verfügt, die

zur Gewährleistung der Integrität des Gesamtsystems notwendig wären, sodass eine unbemerkte Manipulation des Clientsystems möglich ist.

Vergleich biometrischer Templates auf einem Server

Der Vergleich der biometrischen Templates sollte somit nur zentral auf einem speziell abgesicherten Server stattfinden, der ebenfalls die Datenbasis mit dem biometrischen Referenztemplate enthalten sollte, damit eine Übertragung der biometrischen Referenztemplate über das Netzwerk vermieden wird.

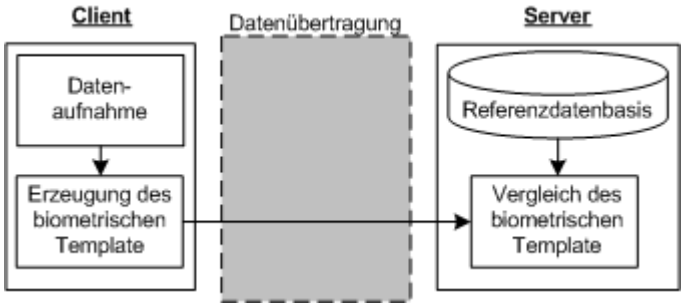

Abbildung 22: Dezentrale Erzeugung des biometrischen Template

Im Weiteren wird daher der Vergleich eines biometrischen Template mit einem Referenztemplate auf einem Server und nicht auf den Clienten stattfinden.

5.3 Modularisierungskonzept

Im Modularisierungskonzept erfolgt die Aufteilung der an der Authentisierung beteiligten Funktionseinheiten in Module und der Entwurf einer Gesamtarchitektur, in der das Zusammenwirken der Module beschrieben wird.

Der Entwurf orientiert sich dabei am Grobkonzept, das die Erzeugung des biometrischen Templates auf dem Client und den Vergleich mit einem Referenztemplate auf dem Server vorschreibt. Weiterhin wurde bereits ein Modul zur Verwaltung der Referenztemplates und der Datenübertragung vorgeschlagen, um bei der Speicherung der biometrischen Templates flexibel zu sein.

5.3.1 Architekturüberblick

Die an der Authentisierung beteiligten Funktionseinheiten sind Datenaufnahme, Templateerzeugung, Datenübertragung, Referenzdatenverwaltung und Templatevergleich. Templateerzeugung und der Vergleich zweier biometrischer Templates (Templatevergleich) kann in einem Modul (biometrisches Funktionsmodul, BFM) zusammengefasst werden. Die Implementierung der Datenübertragungsfunktionseinheit und die Referenzdatenverwaltung erfolgt getrennt in zwei dynamischen Bibliotheken (Datenübertragungsmodul, DTM und Referenzdatenbankmanagementmodul, REFDBM).

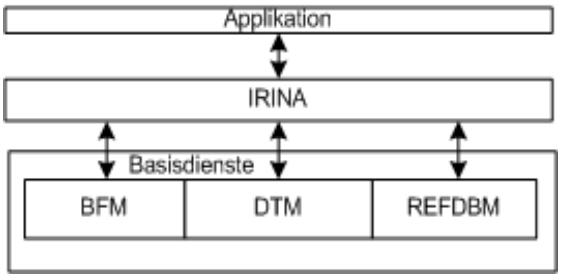

Abbildung 23: Schichtenarchitektur eines Client-Server-Authentisierungssystems

BFM-Modul, DTM-Modul und REFDBM-Modul können sich gegenseitig aufrufen und von anderen Modulen verwendet werden. Sie bilden mit ihren Funktionen eine Basisdienstschicht, die einer darüberliegenden Schicht Basisdienste zur Verfügung stellt. Die darüberliegende Schicht stellt Funktionen zum Erfassen der Iris und zur Authentisierung mittels eines Iris-Biometrik-Verfahrens bereit und wird im Folgenden Iris-Identifikations- und Authentisierungsschnittstelle (IRINA) genannt.

5.3.2 Die Datentypen BIOMETRIC_SAMPLE & BIOMETRIC_TEMPLATE

Die Nutzung eines biometrischen Verfahrens zur Authentisierung setzt die digitale Erfassung des biometrischen Merkmals in einem biometrischen Sample und die Erzeugung eines biometrischen Template voraus. Nachfolgend werden Datentypen definiert werden, die das biometrische Sample und das biometrische Template darstellen.

BIOMETRIC_SAMPLE

Ein biometrisches Sample stellt das digitale Abbild eines biometrischen Merkmals dar. Das Format des biometrischen Sample ist abhängig von der Sensorsoftware mit Hilfe der es erstellt wurde und kann je nach Sensortyp unterschiedlich sein. Bei der Iris-Biometrik handelt es sich um hochauflösende Bilder oder Bildsequenzen, die in unterschiedlichen Formaten, wie z.B. BMP, GIF, JPG oder TIFF, auftreten können. Das in Zukunft für die Iris-Biometrik genutzte Format für die Datenaufnahme und Erstellung des biometrischen Sample kann nicht vorausgesagt werden. Darüber hinaus kann die zukünftige Software zur Ansteuerung von Sensoren ganz neue Formate anbieten. Ferner stellt sich die Frage, ob das biometrische Sample ungeschützt in Funktionen übergeben werden sollte, oder ob direkt nach der Datenaufnahme eine Signierung und Verschlüsselung stattfinden muss. Weiterhin steht projektintern noch nicht endgültig fest, welches Format für das biometrische Sample verwendet werden soll.

Darum wird die Datenstruktur für ein biometrisches Sample (BIOMETRIC_-SAMPLE) so definiert, dass sie beliebige Formate aufnehmen kann. Nachfolgende Definition zeigt den Aufbau eines BIOMETRIC_SAMPLE.

Datentyp: BIOMETRIC_SAMPLE

```
typedef struct{
        DWORD    structSize;
        DWORD    samplesize;
        TYPE_INFO typeInfo;
        BIOMETRIC_SAMPLE_FORMAT sampleFormat;
        PVOID      pReserved1;
        PVOID      pReserved2;
}BIOMETRIC_SAMPLE, *PBIOMETRIC_SAMPLE;
```

Der Parameter *„structSize“* gibt die Größe der BIOMETRIC_SAMPLE Struktur an und kann dazu dienen, die Struktur im Speicher zu identifizieren. Der Parameter *„samplesize“* gibt die Größe des biometrischen Samples in Byte an. Dieser Parameter ist wichtig, um für das BIOMETRIC_SAMPLE Speicher zu reservieren. Mit Hilfe von *„typeInfo“* kann die Struktur zur Laufzeit im Speicher erkannt werden. Darüber hinaus gibt „typeInfo“ Auskunft über die Version des BIOMETRIC_SAMPLE. Das *„sampleFormat“* legt fest, um welches Format es sich bei den Informationen handelt, die im BIOMETRIC_SAMPLE enthalten sind. Momentan kann der Typ BIOMETRIC_SAMPLE_FORMAT drei mögliche Werte für Bildformate annehmen. Diese können zukünftig durch das Hinzufügen neuer Formate beliebig erweitert werden.

Weiterhin sind in der Struktur zwei Void-Pointer enthalten (*„preserved1“*, *„preserved2“*), die für die zukünftige Nutzung reserviert sind. Hier werden in Zukunft Pointer auf Strukturen platziert, die zusätzliche Informationen angeben. Beispielsweise könnte dort ein Pointer auf eine PROTECTION-Struktur enthalten sein, die angibt, ob und wie Informationen im BIOMETRIC_SAMPLE geschützt sind.

BIOMETRIC_TEMPLATE

Das Format des biometrischen Template ist abhängig vom Algorithmus der es erzeugt und kann ebenfalls noch nicht vorausgesagt werden. Darum wird für das biometrische Template ebenfalls ein Datentyp (BIOMETRIC_TEMPLATE) definiert, der biometrische Templates beliebiger Formate aufnehmen kann.

BIOMETRIC_TEMPLATE

```
typedef struct{
        DWORD    structSize;
        DWORD    templateSize;
        TYPE_INFO typeInfo;
        BIOMETRIC_TEMPLATE_FORMAT templateFormat;
        PVOID    pReserved1;
        PVOID    pReserved2;
}BIOMETRIC_TEMPLATE, *PBIOMETRIC_TEMPLATE;
```

Der Parameter *„structSize“* gibt die Größe der BIOMETRIC_TEMPLATE Struktur an. Die *„templateSize“* enthält die Größe des Template in Byte. Mit Hilfe von *„typeInfo“* kann das biometrische Template identifiziert werden. Durch *„templateFormat“* wird angegeben, in welchem Format die im BIOMETRIC_TEMPLATE gespeicherten Informationen vorliegen. Momentan ist für das Format nur der Wert IRISTEMPLATE_V1 festgelegt. Das BIOMETRIC_TEMPLATE_FORMAT kann um beliebige Einträge erweitert werden. Weiterhin sind in der Struktur zwei Void-Pointer enthalten, die für die zukünftige Nutzung reserviert sind (*„preserved1“*, *„preserved2“*).

Hier werden in Zukunft Pointer auf Strukturen platziert, die zusätzliche Informationen angeben. Beispielsweise könnte dort, wie in der BIOMETRIC_SAMPLE-Struktur ein Pointer auf eine PROTECTION-Struktur enthalten sein, die angibt, ob und wie die Informationen im BIOMETRIC_TEMPLATE geschützt sind.

5.3.3 Iris-Identifikations- und Authentisierungsmodul (IRINA)

Das Iris-Identifikations- und Authentisierungsmodul wird von Programmen zur Identitätsbestimmung mittels Iris-Biometrik-Verfahren genutzt. Es verfügt über die Funktionen „GetBiometrikSample“ und „BiometricLogonUser“.

Funktion „getIrisBiometricSample“

Die Funktion „getIrisBiometrikSample“ dient der Datenaufnahme und zeigt bei ihrem Aufruf einen Dialog zur Erfassung des Auges und der Benutzerdaten an.

Schnittstellendefinition

```
int getIrisBiometrikSample(
    char * pUsername,   // [out]
    char *pNamespace,  // [out]
    char * pKey,  // [out]
    PBIOMETRIC_SAMPLE  pIrisSample, // [out]
    PVOID pOptional // [in, out]
);
```

Diese werden nach der Datenaufnahme zusammen mit dem biometrischen Sample zurückgegeben. Zur Erstellung des biometrischen Sample wird das biometrische Funktionsmodul genutzt, das weiter unten dargestellt wird. Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Funktionsparameter

pUsername: Pointer auf den Namen eines Benutzers, der im Dialog zur Erfassung des biometrischen Merkmals angezeigt wird.

pNamespace: Pointer auf den Namen der Domäne, an der sich der Benutzer anmelden möchte.

pKey: Pointer auf ein eingegebenes Passwort oder einen Schlüssel, der von einer Chipkarte stammt.

pIrisSample: Pointer auf einen Puffer, der ein biometrisches Sample enthält. Dort ist das biometrische Sample der Iris gespeichert.

pOptional: Pointer auf einen Puffer, der optionale Daten zur Authentisierung enthält. Dort können beliebige zusätzliche Daten abgelegt werden.

Funktion „irisBiometricLogonUser“

Die Funktion „irisBiometrikLogonUser“ dient zur Authentisierung und Anmeldung eines Benutzers, mit den vorher erfassten biometrischen Sampledaten der Augenoberfläche, und kapselt den gesamten Identifikations- und Authentisierungsvorgang. Die Funktion gibt einen Zeiger auf einen Puffer (pSecdata) zurück, der Informationen mit dem benutzerspezifischen Identifikator enthält.

Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Schnittstellendefinition

```
int irisBiometricLogonUser(
   char    *pUsername, // [in]
   char    *pNamespce, // [in]
   char     *pKey, // [in]
   PBIOMETRIC_SAMPLE pIrisSample, // [in]
   char    * pBfmModule, // [in]
   PVOID pSecSystemData, // [out]
   PVOID pReserved  // [in, out]
   PVOID pOptional  // [in, out]
);
```

Funktionsparameter

Die Funktionsparameter pUsername, pNamespace, pKey, pIrisSample und pOptional wurden bereits in der Beschreibung der Funktion „getIrisBiometricSample“ erklärt.

pBfmModule: Pointer auf Namen des BFM-Moduls, das während der Anmeldung zur Erzeugung und zum Vergleich des biometrischen Templates verwendet werden soll.

pSecSystemData: Pointer auf Daten, die vom Authentisierungssystem nach erfolgreicher Anmeldung zurückgegeben werden, wie Identifikatoren oder Zugriffstoken.

5.3.4 Biometrisches Funktionsmodul (BFM)

Das Biometrische Funktionsmodul (BFM) ermöglicht das Erzeugen eines biometrischen Iris-Templates aus einem zuvor während der Datenaufnahme erstellten biometrischen Sample der Augenoberfläche und den Vergleich zweier biometrischer Templates.

Es ist unabhängig von der Ansteuerung von Gerätetreibern zur Datenaufnahme oder Datenbanken, die Referenztemplates enthalten, um einen möglichst flexiblen Einsatz zu ermöglichen und stellt damit lediglich Basisfunktionen bereit, die zur biometrischen Identifikation und Authentisierung benutzt werden können.

Funktion CreateIrisTemplate

Schnittstellendefinition
int createIrisTemplate(PBIOMETRIC_SAMPLE *pIrisSample*, // [in] PVOID *pReserved* , // [in, out] PVOID *pOptional*, // [in] PBIOMETRIC_TEMPLATE **pIristemplate* // [out]);

Mit der Funktion „createIrisTemplate“ kann ein biometrisches Template der Iris erstellt werden. Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Funktionsparameter

Die Funktionsparameter pIrisSample und pOptional wurden bereits in der Beschreibung der Funktion „getIrisBiometricSample“ erklärt.

pIrisTemplate: Pointer auf einen Puffer, der ein biometrisches Template enthält. Dort ist das biometrische Template der Iris gespeichert.

Funktion „compareBiometricTemplates“

Die Funktion „compareBiometricTemplates“ vergleicht zwei Templates auf Übereinstimmung. Als Eingangsdaten werden zwei Iristemplates übergeben, die miteinander verglichen werden sollen.

Schnittstellendefinition

```
int compareBiometricTemplates(
   PBIOMETRIC_TEMPLATE pIristemplate,  // [in]
   PBIOMETRIC_TEMPLATE pIrisreferencetemplate, // [in]
   PVOID pReserved, // [in,out]
   PVOID pOptional, // [in]
   BOOL  result // [out]

);
```

Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Funktionsparameter

Die Funktionsparameter pIrisTemplate und pOptional wurden bereits in der Beschreibung der Funktion „getIrisBiometricSample“ erklärt.

pIrisreferencetemplate: enthält Pointer auf einen Puffer, der ein biometrisches Template enthält. Dort wird das biometrische Referenztemplate gespeichert.

pResult: gibt an, ob die Iristemplates übereinstimmen oder nicht.

pReserved: enthält Pointer auf Struktur mit zusätzlichen Informationen und ist für die zukünftige Nutzung vorgesehen.

5.3.5 Datentransfermodul (DTM)

Das Datentransfermodul kapselt das Protokoll zur Authentisierung und verfügt nur über eine öffentliche Funktion „dtmBiometrikLogonUser“, die die gleichen Ein- und Ausgangsdaten wie die Funktion „irisBiometrikLogonUser“ besitzt.

5.3.6 Referenzdatenbankmanagementmodul (REFDBM)

Das Referenzdatenbankmanagementmodul (REFDBM) dient zur Ansteuerung der Datenbank, in der sich die biometrischen Referentemplates befinden und stellt dazu Verwaltungsfunktionen bereit. Es ist stark abhängig vom zugrundeliegenden Datenbanksystem und muss für die Ansteuerung verschiedener Datenbanksysteme jeweils neu implementiert werden.

Funktion „createBiometricAccount“

Schnittstellendefinition
int createBiometricAccount(PVOID pDatabaseInfoBuffer, // [in] char *pAccountName, // [in] PBIOMETRIC_TEMPLATE pIrisTemplate, // [in] PVOID pOptionalAccountData // [in] **);**

Die Funktion „createBiometricAccount“ dient der Erstellung eines Kontos in der biometrischen Referenzdatenbank.

Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Funktionsparameter

pDatabaseInfoBuffer: Pointer auf einen Puffer, der Daten zur Verbindung mit der Referenzdatenbank enthält.

pAccountName: Pointer auf Namen eines Benutzerkontos in der Referenzdatenbank.

pOptionalAccountData: Pointer auf Daten, die zusätzlich zum Iristemplate in der Referenzdatenbank gespeichert werden sollen.

Funktion „deleteBiometricAccount“

Schnittstellendefinition
int deleteBiometricAccount(PVOID pDatabaseInfoBuffer, // [in] char *pAccountName // [in] **);**

Die Funktion „deleteBiometricAccount“ dient zum Löschen eines Benutzerkontos in der biometrischen Referenzdatenbank. Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Funktionsparameter

Die Funktionsparameter pDabaseInfoBuffer und pAccountName wurden bereits in der Beschreibung der Funktion „createBiometricAccount“ erklärt.

Funktion „getBiometricAccountData“

Die Funktion „getBiometricAccountData“ dient zum Abruf der in der biometrischen Referenzdatenbank gespeicherten Daten eines Benutzerkontos.

Schnittstellendefinition
int GetBiometricAccountData(PVOID pDatabaseInfoBuffer, // [in] char *pAccountName, // [in] PBIOMETRIC_TEMPLATE *ppIrisTemplate, // [out] PVOID *ppOptionalAccountData // [out] **);**

Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Funktion „updateBiometricAccountData“

Die Funktion „updateBiometricAccountData“ dient zum Aktualisieren der in der biometrischen Referenzdatenbank gespeicherten Daten.

Bei erfolgreicher Ausführung liefert die Funktion einen Rückgabewert von Null, ansonsten einen Fehlercode größer Null.

Schnittstellendefinition
int updateBiometricAccount(PVOID pDatabaseInfoBuffer, //[in] char *pAccountName, //[in] PBIOMETRIC_TEMPLATE pnewIrisTemplate, //[in] PVOID pnewOptionalAccountData //[in] **);**

Funktionsparameter

Die Funktionsparameter pDabaseInfoBuffer, pAccountName und PBIOMETRIC_TEMPLATE wurden bereits erklart.

pNewIrisTemplate: Pointer auf ein neues Iristemplate.

pNewOptionalAccountData: Pointer auf aktualisierte Daten, die zusätzlich zum Iristemplate abgespeichert werden.

5.4 Konzepte zur Implementierung in bestehende Systeme

Nachdem nun ein Modularisierungskonzept erarbeitet wurde, wird gezeigt, wie eine Implementierung in bestehende Client-Server-Authentisierungssysteme erfolgen kann. Dabei werden zwei Möglichkeiten betrachtet: Implementierung durch Vorschaltung und Verkettung und Implementierung durch Einbettung und Integration.

5.4.1 Vorschaltung und Verkettung

Bei der Implementierung durch Vorschaltung und Verkettung wird ein biometrisches Authentisierungssystem dem nichtbiometrischen Authentisierungssystem vorgeschaltet. Es findet zuerst eine biometrische Authentisierung statt. Bei erfolgreicher Authentisierung besitzt das biometrische Client-Server-Authentisierungssystem Informationen, mit dem es einen Benutzer nun auf dem bisherigen

Weg authentisiert und im Computersystem mit Hilfe des nichtbiometrischen Authentisierungssystems anmelden kann. Es erfolgt eine doppelte Authentisierung. Um diesen Mechanismus zur Verfügung zu stellen, müssen die Authentisierungssysteme miteinander verkettet werden. Das biometrische Authentisierungssystem muss über Informationen der Identifikations und Authentisierungsmerkmale der Benutzer des nichtbiometrischen Authentisierungssystems verfügen.

Authentisierungsmerkmale des nichtbiometrischen Authentisierungssystems wie Kennwörter können durch zufällig erzeugte Schlüssel ausgetauscht werden, da der Benutzer diese nicht mehr zu Gesicht bekommt und eingeben muss. Diese können systemtechnisch periodisch gewechselt werden, damit eine größere Sicherheit gewährleistet ist.

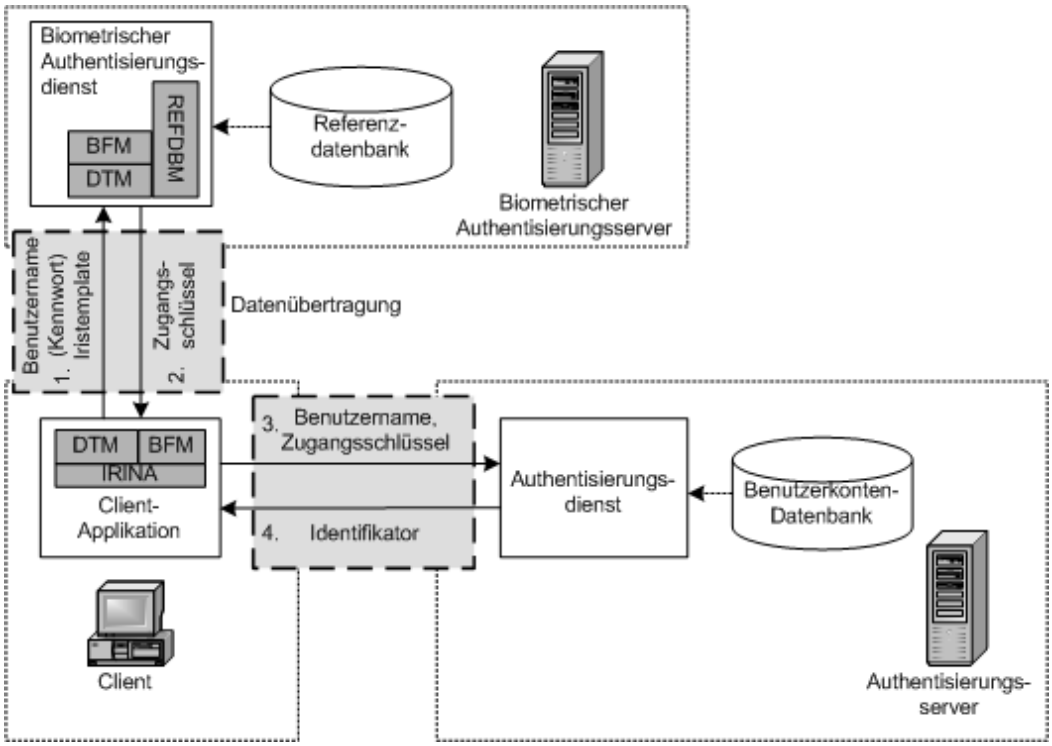

Abbildung 24: Implementierung durch Vorschaltung und Verkettung

Dabei muss dann ebenfalls eine Aktualisierung in den Benutzerkonten der biometrischen Referenzdatenbank erfolgen. Im nichtbiometrischen Authentisierungssystem wird die dialogbasierte Datenerfassungsfunktionseinheit abgeschaltet und durch eine neue aus dem IRINA-Modul ersetzt.

Das biometrische Funktionsmodul, der Referenzdatenbankmanager und das Datenübertragungsmodul müssen neu implementiert werden, um das biometrische Authentisierungssystem zur Verfügung zu stellen.

Darüber hinaus müssen Administrationsprogramme zur Verwaltung von Kennwörtern im nichtbiometrischen Authentisierungssystem angepasst werden. Werden im biometrischen Authentisierungssystem neben der Iris Kennwörter zur Authentisierung benutzt, und erfolgte im nichtbiometrischen Authentisierungssystem eine Ersetzung durch zufällige Schlüssel, so darf sich ein Wechsel des Kennworts nur auf das biometrische Authentisierungssystem auswirken, da im nichtbiometrischen Authentisierungssystem die Zugangsschlüssel von selbst erzeugt und gewechselt werden.

5.4.2 Einbettung und Integration

Die zweite Möglichkeit ist die Einbettung und Integration in ein vorhandenes Authentisierungssystem. Diese kann mit einem erheblichen Aufwand verbunden sein, da in die Systemmechanismen des nichtbiometrischen Authentisierungssystems selbst eingegriffen werden muss.

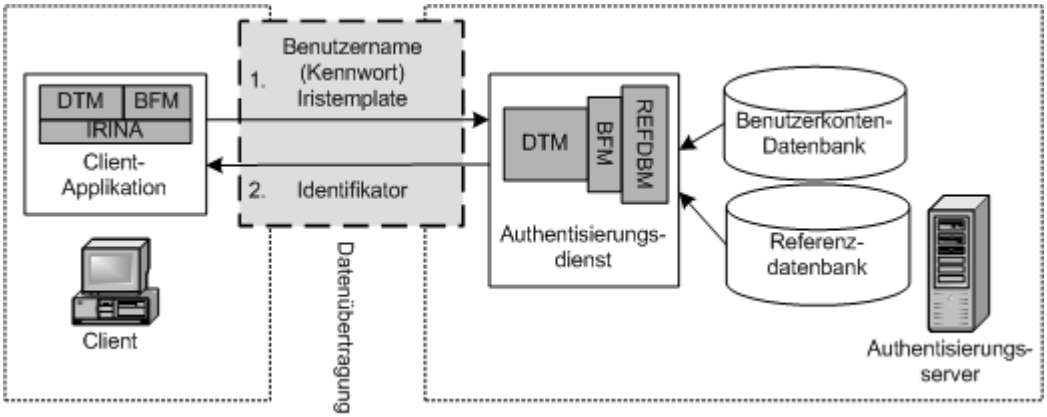

Abbildung 25: Implementierung durch Einbettung und Integration

Hier muss wieder das biometrische Funktionsmodul BFM erstellt werden, um das Iris-Biometrik-Verfahren in das System zu integrieren. Auf die Implementierung des Referenzdatenbankmanagementmoduls kann eventuell verzichtet werden, da

das nichtbiometrische Authentisierungssystem selbst über Funktionen verfügen kann, mit denen die Daten in eine systemeigene Referenzdatenbank gespeichert werden können. Die Implementierung der Funktionalität des DTM-Moduls muss gegebenenfalls in einem Modul des nichtbiometrischen Authentisierungssystems stattfinden, da das nichtbiometrische Authentisierungssystem intern andere Schnittstellen verwendet. Gleiches gilt für die Funktionalität des IRINA Moduls. Nichtbiometrische Authentisierungssysteme können intern über Module verfügen, in denen die Funktionen zur Erfassung der Anmeldeinformationen und zur Authentisierung implementiert sind. Eine ordnungsgemäße Einbettung sollte immer vollständig in die vorhandene Modularchitektur erfolgen.

Kapitel 6

Implementierung in Windows 2000

Das integrale Sicherheitssubsystem von Windows 2000 lässt durch seinen modularen Aufbau die Implementierung und Integration neuer Verfahren zur Authentisierung zu.

In diesem Kapitel wird dargestellt, wie eine Implementierung durch *Vorschaltung und Verkettung* und *Einbettung und Integration* in Anlehnung an das im Kapitel 5.3 erstellte Modularisierungskonzept im Windows 2000 Betriebssystem erfolgen kann.

Dabei wird erläutert, innerhalb welcher Komponenten der Anmeldearchitektur, die im Modularisierungskonzept definierten Module integriert werden können, um eine Authentisierung mit Hilfe eines biometrischen Authentisierungsverfahrens, z.B. der Iris-Biometrik, zu implementieren.

Dazu wird im Abschnitt 6.1 „Entwicklungs- und Testumgebung“ über das Entwickeln und Testen neuer Betriebssystemkomponenten berichtet. Hier wird auf die verwendeten Arbeitsmittel zur Implementation und die Vorgehensweise zum Testen neuer Betriebssystemkomponenten eingegangen.

Abschnitt 6.2 „Erstellung neuer Betriebssystemkomponenten“ zeigt, mit welchen Methoden neue Betriebssystemkomponenten erzeugt werden können.

Abschließend wird im Abschnitt 6.3 die Implementierung durch Vorschaltung und Verkettung und im Abschnitt 6.4 die Implementierung durch Einbettung und Integration erläutert.

6.1 Entwicklungs- und Testumgebung

Als Entwicklungsplattform diente ein Pentium III Computer mit einem 500 Mhz Prozessor, einer ATI Xpert98 Grafikkarte, einer Haupauge Videokarte, einer selbstentwickelten Digitalkamera und einer 20 GB Festplatte. Auf der Entwicklungsplattform waren mehrere Betriebssysteme installiert.

Dabei handelte es sich um Windows NT 4.0 Workstation, 2 Windows 2000 Professional auf unterschiedlichen Partitionen und einer 120 Tage Testversion von Windows 2000 Server.

Im Zeitraum von März 2001 bis Juni 2001 wurde das Windows NT 4.0 Betriebssystem (Servicepack 6) als Entwicklungssystem mit Visual C++ 6.0 (Servicepack 5) und dem Plattform SDK vom Oktober 2000 benutzt. Das Entwicklungssystem befand sich dabei auf einer FAT Partition, sodass auf die Partition mit MSDOS zugegriffen werden konnte.

Daneben gab es ein Windows 2000 (Servicepack 1) Betriebssystem, das ebenfalls auf einer FAT Partition für den Zugriff mit MSDOS installiert war und hauptsächlich zum Testen der erstellten Software diente. Wenn das Testsystem im Testvorgang unbrauchbar wurde, konnte mit Hilfe einer Kopie der Registrierungsdatenbank der ursprüngliche Zustand des Testsystems wiederhergestellt werden, indem der Computer mit MSDOS gestartet wurde und eine Batchdatei die Registrierungsdatenbank mit den registrierten neuen Betriebssystemkomponenten durch die Ursprungsversion Ersetzen. Diese Vorgehensweise hat sich bewährt, da sie eine sehr effiziente und kostengünstige Variante war, das Betriebssystem wieder in den Ursprungszustand zu bringen.

Eine Alternative wäre der Start von Windows 2000 im abgesicherten Modus gewesen. Dort hätten die entsprechenden Registrierungseinträge gelöscht oder geändert werden können, sodass das Testsystem wieder startbar ist. Dieser Vorgang benötigt aber sehr viel mehr Zeit als der Start mit MSDOS. Allein der Bootvorgang im abgesicherten Modus benötigte ca. 3 Minuten (MSDOS benötigte 1,5 Minuten), sodass der Einsatz des abgesicherten Modus eine unattraktive Alternative darstellte.

Eine weitere Möglichkeit wäre der Einsatz von NTFS-DOS. Dann hätte der Computer ebenfalls mit Hilfe von MSDOS gestartet werden können, und es wäre mit dem NTFS-DOS Dateisystemtreiber möglich gewesen, auf eine NTFS Partition zuzugreifen und dort die Registrierungsdatenbank zu ersetzen. Das Testsystem hätte so auf eine NTFS Partition installiert werden können.

Es wäre auch möglich gewesen, das Testsystem auf einer NTFS Partition zu installieren und dann die beschädigte Registrierungsdatenbank über ein parallel installiertes Windows NT oder Windows 2000 Betriebssystem zu erneuern.

Da der Bootvorgang von Windows NT oder Windows 2000 länger als der von MSDOS dauert, wurde diese Alternative als schnelle Wiederherstellungsmöglichkeit verworfen.

Ab Juni 2001 wurde ein weiteres Windows 2000 Professional als Entwicklungssystem eingerichtet. Dieses System war aber im Unterschied zum Windows NT 4.0 Entwicklungssystem auf einer NTFS Partition installiert, jedoch mit Ausnahme der Betriebssystemversion völlig identisch zum anderen Windows NT 4.0 Entwicklungssystem (Visual C++ 6.0 (Servicepack 5) und MSDN Library Oktober 2000).

Zeitgleich wurde eine 120 Tage Testversion von Windows 2000 Advanced Server auf einer weiteren Partition installiert, um die Komponenten dort ebenfalls testen zu können. Da diese Installation den Active Directory Verzeichnisdienst enthalten sollte, musste sie auf eine NTFS Partition installiert werden, was den Reparaturvorgang und die Wiederanlaufzeit des Betriebssystems erheblich verlängerte. Zwar konnte auf die beschädigte Installation mit einem anderen parallel installierten Windows Betriebssystem zugegriffen werden, jedoch dauerte dieser Vorgang länger als der Start von MSDOS.

6.2 Erstellung neuer Betriebssystemkomponenten

Bei der Erstellung neuer Betriebssystemkomponenten, die als dynamische Bibliothek implementiert werden und andere ersetzen, bieten sich zwei Vorgehensweisen an.

Entweder kann die Komponente komplett neu erstellt werden oder es werden nur bestimmte Funktionen neu implementiert und der Rest der Funktionalität wird von der ursprünglichen Betriebssystemkomponente bereitgestellt.

Die komplette Neuerstellung ist oft zeitaufwendig und kann sehr komplex werden. Die Schnittstellenfunktionen einiger Komponenten sind zwar in der MSDN Library sehr gut dokumentiert, es gibt aber selten Beispiele, die den internen Aufbau der Betriebssystemkomponenten darstellen. Beispiele dafür sind die Schnittstellen eines Authentisierungspakets oder eines Sicherheitsunterstützungsanbieters. Dort sind die Schnittstellenfunktionen dokumentiert, die genaue

interne Funktionsweise bleibt dem Entwickler jedoch verborgen, da nur Beispiele existieren, die zeigen, wie die einzelnen Komponenten benutzt werden, aber nicht, wie sie implementiert sind, sodass es nicht einfach ist, eine eigene Entwicklung zu betreiben. Darum ist es oft zweckmäßig nur einzelne Teilfunktionen einer Komponente zu ersetzen.

Bei dieser Vorgehensweise werden in einer neuen Komponente einige Funktionsaufrufe selbst bearbeitet und der Rest an die alte Betriebssystemkomponente weitergeleitet. Die neue Komponente lädt bei ihrer Initialisierung die alte in ihren Adressraum und leitet während der Laufzeit mit Hilfe von Funktionszeigern, die Aufrufe, die nicht ersetzt werden, an die ursprüngliche Komponente weiter.

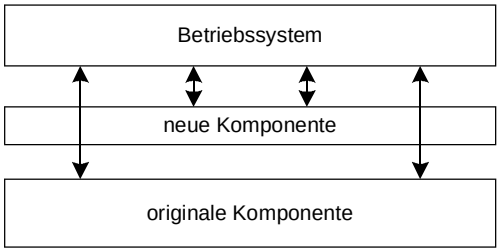

Abbildung 26: Ersetzen einer Betriebssystemkomponente

Allerdings ist diese Vorgehensweise nicht immer möglich. Im Zusammenhang mit Authentisierungsprotokollen sind innerhalb der damit verbundenen dynamischen Bibliotheken Protokolle gekapselt. Eine Modifikation des Protokolls ist darum mit einer teilweisen Ersetzung nicht möglich, da dem Entwickler die Abfolge und Steuerung des Nachrichtenaustauschs zwischen den Kommunikationspartnern verborgen bleibt. Dann müssen alle Funktionen der Komponente ersetzt werden.

6.3 Implementierung durch Vorschaltung und Verkettung

6.3.1 Allgemeines

Die Implementierung des Iris-Biometrik-Verfahrens durch Vorschaltung und Verkettung hat den Vorteil, dass nicht mehrere Komponenten des integralen Sicherheitssubsystems ergänzt oder erneuert werden müssen. Es wird ein biometrisches Authentisierungssystem aufgebaut, das parallel zum betriebssystemeigenen existiert und eine Authentisierung mittels Iris-Biometrik durchführt.

Lediglich die Clientkomponente des betriebssystemeigenen Authentisierungssystems muss ausgetauscht oder ergänzt werden, um eine Erfassung von biometrischen Merkmalen und eine Überprüfung mit Hilfe des biometrischen Authentisierungssystems zu ermöglichen.

Innerhalb des Windows 2000 Betriebssystems ist das die GINA Bibliothek, in der Anmeldedialoge zur Erfassung von Anmeldeinformationen und Sitzungsdialoge zur Steuerung der Sitzung enthalten sind.

Sie wird durch eine neue GINA mit dem Namen IRIS-GINA ersetzt, die nun um Anmeldedialoge zur Erfassung der Iris erweitert wurde und die im Konzept vorgeschlagenen Module BFM, DTM und REFDBM nutzt, um eine biometrische Authentisierung mit Hilfe des vorgeschalteten biometrischen Authentisierungssystems zu ermöglichen.

Das IRINA-Modul zur Erfassung der Iris und Einleitung des Authentisierungsvorgangs muss hierbei nicht extra implementiert werden. Die Funktion zur Erfassung der Authentisierungsmerkmale sind in der GINA enthalten und können dort implementiert werden. Die Funktion zur Authentisierung über das Netzwerk am biometrischen Authentisierungsdienst wird vom DTM Modul bereitgestellt, sodass hier eine Stubfunktion innerhalb eines zusätzlichen IRINA Moduls nicht notwendig ist.

6.3.2 Ablauf der Authentisierung

Der Authentisierungsvorgang gestaltet sich nun folgendermaßen: Nach dem Drücken der Tastenkombination „Strg+Alt+Entf“ sendet Winlogon eine SAS Nachricht an eine neu implementierte GINA (IRISGINA).

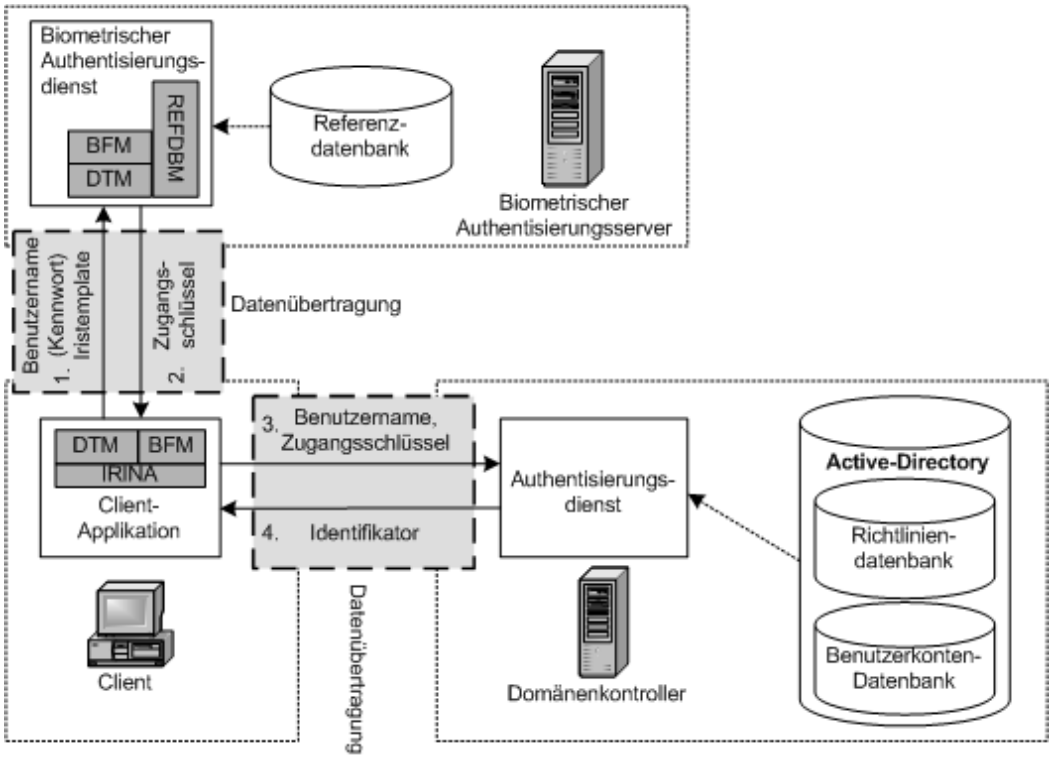

Abbildung 27: Implementierung durch Vorschaltung und Verkettung in Windows 2000

Der Dialog zur Anmeldung wird aufgerufen und stellt dem Benutzer die Möglichkeit zur Erfassung der Iris bereit. Dazu wurde das „Video for Windows API“ genutzt, mit dem es möglich ist, CCD Kameras anzusteuern und Bilder oder Videoströme aufzunehmen.

Nach der erfolgreichen Erfassung der Authentisierungsmerkmale wird das erfasste Irisabbild mit Hilfe des BFM-Moduls in ein biometrisches Template umgewandelt (createIrisTemplate) und zusammen mit dem Passwort an das DTM Modul (dtmBiometricLogonUser) zur Authentisierung am biometrischen Authentisierungsserver weitergeleitet.

Das DTM Modul auf der Serverseite prüft mit Hilfe des REFDBM Moduls, ob ein Konto für den Benutzer existiert und gibt gegebenenfalls die in der Referenzdatenbank enthaltenen Informationen des Benutzerkontos zurück.

Nun prüft das DTM Modul mit Hilfe des BFM Moduls (compareBiometricTemplates), ob das Iristemplate mit dem Referenztemplate aus der Referenzdatenbank übereinstimmt. Weiterhin können zusätzliche Prüfungen erfolgen, bei denen das Passwort aus der Referenzdatenbank genutzt wird. Waren die Prüfvorgänge erfolgreich, so sendet das DTM Modul des biometrischen Authentisierungsservers den benutzerspezifischen Zugangsschlüssel zum DTM-Modul des Clients, welches innerhalb der IRISGINA ausgeführt wird.

Dort wird nun durch die Funktion LogonUser der herkömmliche Authentisierungsvorgang eingeleitet, wobei der Zugangsschlüssel als Passwort benutzt wird.

6.3.3 Vor- & Nachteile

Die Implementierung des Iris-Biometrik-Verfahrens in Windows 2000 mittels Vorschaltung und Verkettung bedarf des Austauschs von nur einer Komponente des integralen Sicherheitssubsystems. Hier muss die GINA Bibliothek erneuert werden, um im Anmeldedialog eine Erfassung biometrischer Merkmale zu ermöglichen und den Anmeldevorgang einzuleiten.

Die GINA Schnittstellenfunktionen sind gut dokumentiert und eine Implementierung daher gut durchführbar. Darüber hinaus werden hier die internen Authentisierungsmechanismen nicht verändert, sodass hier eine relativ unabhängige Implementierung vom Betriebssystem stattfinden kann.

Durch die Integration vom System generierter, zufälliger Zugangsschlüssel, die als Passwortersatz im betriebssystemeigenen Authentisierungssystem dienen, können die Schwächen von Passwortsystemen im betriebssystemeigenen Authentisierungssystem gezielt bekämpft werden.

Hier muss jedoch auf Wechselwirkungen mit der bestehenden Passwortpolicy geachtet werden. Die Richtlinie zum Wechseln von Kennwörtern muss abgeschaltet und durch einen automatischen Dienst ersetzt werden.

Weiterhin muss darauf geachtet werden, dass durch den automatischen Wechsel der Zugangsschlüssel keine Anomalien zwischen der biometrischen Referenzdatenbank des biometrischen Authentisierungssystems und der Benutzerkontenda-

tenbank des betriebssystemeigenen Authentisierungssystems auftreten, da die Zugangsschlüssel in beiden Systemen gespeichert werden müssen.

Sind die biometrischen Referenzdaten nicht im Verzeichnisdienst des Betriebssystems enthalten, so muss eine Strategie zur Ausfallsicherheit erarbeitet und implementiert werden. Der betriebssystemeigene Verzeichnisdienst wird auf mehreren Domänenkontrollern repliziert und syncronisiert sich ständig. Bei Ausfall eines Domänenkontrollers kann immer noch eine Authentisierung erfolgen, wenn weitere Domänenkontroller in der Domäne enthalten sind.

Im betriebssystemeigenen Authentisierungssystem bleiben die Schwächen der zur Authentisierung verwendeten Protokolle erhalten, da dort keine Änderungen an den Protokollen erfolgen.

Darüber hinaus ist es systemintern immer noch möglich, sich nur mit einem Passwort anzumelden. Da die Passwörter jedoch durch Zugangsschlüssel ersetzt wurden, können Brute Force Angriffe erschwert werden.

6.4 Implementierung durch Einbettung und Integration

6.4.1 Allgemeines

Die Implementierung durch Einbettung und Integration erfordert das Ergänzen oder die Ersetzung mehrerer Komponenten des integralen Sicherheitssubsystems und ist dadurch in Windows 2000 aufwendiger als die Implementierung durch Vorschaltung und Verkettung.

Im Betriebssystem muss hier nicht nur die GINA, sondern auch das Authentisierungs- und Sicherheitspaket (AP/SSP) ergänzt oder ausgetauscht werden. Die Implementation der Funktionalität einiger im Modularisierungskonzept festgelegten Module erfolgt direkt in die Komponenten des integralen Sicherheitssubsystems.

Da bei der Implementierung durch Einbettung und Integration die GINA erweitert oder ersetzt werden muss, kann hier ebenfalls eine Implementierung des IRINA-Moduls entfallen.

Die Funktionalität des DTM Moduls für eine Authentisierung im Netzwerk wird in Windows 2000 durch ein Authentisierungs- und Sicherheitspaket (AP/SSP) bereitgestellt.

Das Referenzdatenbankmanagementmodul sollte extra implementiert werden und kann die Funktionen der Benutzerkontendatenbank des Betriebssystems benutzen. Eine Implementierung des biometrischen Funktionsmoduls ist notwendig, um Iristemplates erstellen und vergleichen zu können.

6.4.2 Ablauf

Nach dem Drücken der Tastenkombination „Strg+Alt+Entf“ sendet Winlogon eine SAS Nachricht an eine neu implementierte GINA (IRISGINA). Der Dialog zur Anmeldung wird aufgerufen und stellt dem Benutzer die Möglichkeit zur Erfassung der Iris bereit. Dazu wurde das „Video for Windows API“ genutzt, mit dem es möglich ist, CCD Kameras anzusteuern und Bilder oder Videoströme aufzunehmen. Nach der erfolgreichen Erfassung der Authentisierungsmerkmale wird das erfasste Irisabbild mit Hilfe des BFM-Modul in ein biometrisches Template umgewandelt (createIrisTemplate).

Nun wird die Funktion „LsaLogonUser“ aufgerufen, um Benutzernamen, Iristemplate und Passwort an die LSA zu übertragen. Bei dem Aufruf wird bereits der Name des Authentisierungspakets angegeben, das zur Authentisierung genutzt werden soll (IRIS-AP/SSP).

Die Informationen werden mit Hilfe eines LPC[33] an die lokale Sicherheitsautorität zum passenden Authentisierungs- und Sicherheitspaket übertragen.

Dort wird geprüft, ob eine lokale Anmeldung oder eine Anmeldung an einer Domäne stattfinden soll. Befindet sich der Domänenkontroller im Netzwerk, so werden die Authentisierungsinformationen mit Hilfe der Funktionen des Sicherheitspakets zum Domänenkontroller übertragen und dort mit dem BFM-Modul (compareBiometric Templates) verifiziert. Bei erfolgreicher Verifizierung sammelt das Authentisierungspaket benutzerspezifische Informationen wie den Identifikator und die Gruppenzugehörigkeiten und sendet diese an das Authentisierungs- und Sicherheitspaket des Clients zurück. Diese sucht nun nach lokalen Gruppenzugehörigkeiten und Richtlinien und erzeugt daraus ein Zugriffstoken, das an den interaktiven Anmeldeprozess WINLOGON.EXE zurückgegeben wird.

33 LPC=Local Procedure Call

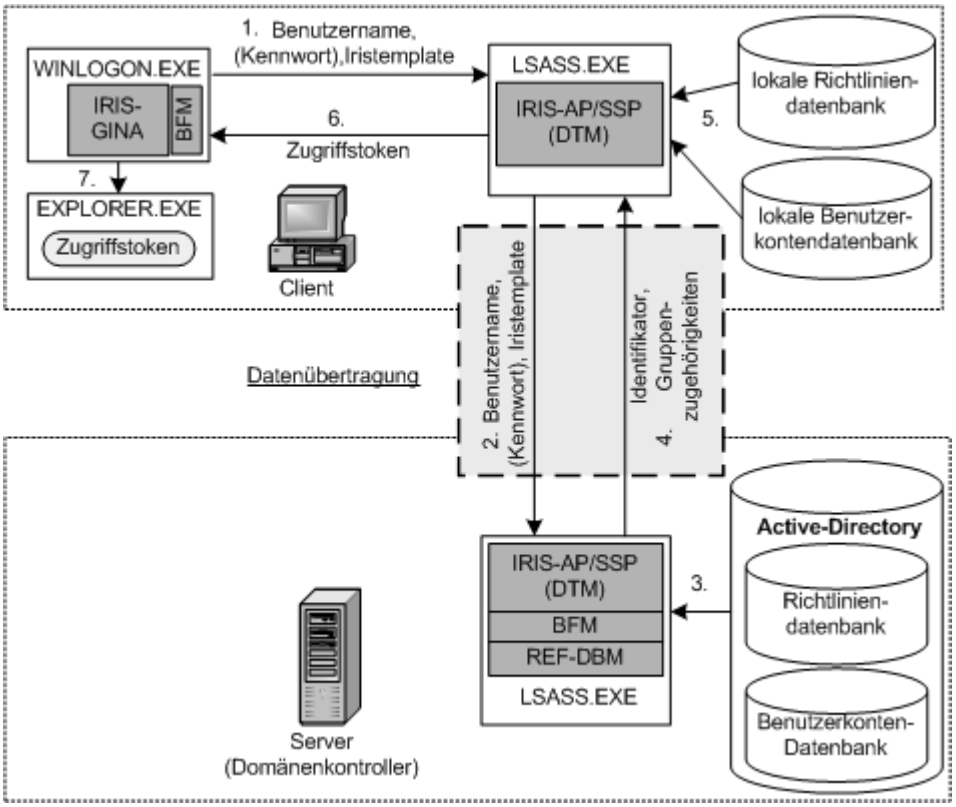

Abbildung 28: Implementierung in Windows 2000 durch Einbettung und Integration

Nach Erhalt des Zugriffstokens, startet WINLOGON.EXE die grafische Benutzershell (EXPLORER.EXE) und bindet dabei das Zugriffstoken an den Benutzerprozess.

6.4.3 Vor- & Nachteile

Gegenüber der Implementierung mittels Vorschaltung und Verkettung hat eine Implementierung durch Einbettung und Integration den Vorteil, dass keine doppelte Authentisierung stattfindet. Die Zeit, die für die Authentisierung benötigt wird ist kürzer, da nur eine Authentisierung stattfindet.

Darüber hinaus kann eine Authentisierung mit Hilfe anderer Verfahren technisch ausgeschlossen werden. Eine Anmeldung mit Benutzername und Passwort ist dann nicht mehr möglich.

Wird ein Kennwort zur biometrischen Authentisierung zusätzlich benutzt, so greift die Kennwortrichtlinie direkt. Keine zusätzlichen Programme zur Synchronisation müssen implementiert werden.

Allerdings erfordert eine Einbettung und Integration die Ersetzung mehrerer Komponenten des integralen Sicherheitssubsystems. Hier müssen die GINA und das Authentisierungs- und Sicherheitspaket ersetzt werden.

Eine Ersetzung des Authentisierungs- und Sicherheitspakets setzt gute Kenntnisse über die internen Sicherheitsmechanismen des Betriebssystems voraus. Diese sind jedoch nur sehr schlecht dokumentiert. Zwar gibt es eine Dokumentation der Schnittstellenfunktionen der einzelnen Komponenten, aber interne Zusammenhänge sind oft nicht klar genug dargestellt. Bücher über die Betriebssystemsicherheit wie [Microsoft 2000-1] stellen die Zusammenhänge oberflächlich, ungenügend und schlecht strukturiert dar.

Ohne die Betrachtung des Aufwands für das Verstehen und Implementieren von betriebssystemspezifischen Funktionen ist der Aufwand zur Implementierung der Funktionen des BFM-Moduls, DTM-Moduls und REFDBM-Moduls ungefähr gleich, da hier identische Funktionen mit Hilfe einer gleichen Sammlung von Funktion (WIN API oder zusätzliche API Funktionen der Bibliotheken von Fremdherstellern) implementiert werden.

Kapitel 7

Zusammenfassung und Ausblick

Diese Arbeit berichtete über die Implementierung einer Iris-Biometrik in ein „Client-Server-Authentisierungssystem“.

Dabei wurde im Kapitel 2 festgestellt, dass die Iris-Biometrik ein physiologisches biometrisches Verfahren ist, das die Muster der Iris benutzt um eine Person wiederzuerkennen. Der Erkennungsvorgang erfolgt durch den Vergleich zweier biometrischer Templates, wobei das eine während der Authentisierung und das andere (Referenztemplate) während einer Enrollementphase erzeugt wurde. Das Template dient dabei als Kenngröße des biometrischen Merkmals (IRIS) einer bestimmten Person, und wird nach einer Datenaufnahme mit Hilfe eines Algorithmus erstellt.

Der Vergleich zweier biometrischer Templates gestaltet sich schwieriger als der Vergleich von Kennwörtern in nichtbiometrischen Authentisierungssystemen. Die biometrischen Templates einer Person stimmen durch unterschiedliche Positionierung des biometrischen Merkmals und verschiedene optische Verhältnisse bei der Datenaufnahme nicht exakt miteinander überein. Es kommen Schwellwertverfahren zum Einsatz, die beim Vergleich eine Toleranz einbeziehen, um zu entscheiden, ob zwei biometrische Templates miteinander übereinstimmen.

Die Implementierung einer Iris-Biometrik kann also nicht nur durch die Integration einer Komponente zur Templaterzeugung erfolgen. Es muss weiterhin die Vergleichskomponente des nichtbiometrischen Authentisierungssystems ersetzt werden.

Kapitel 3 stellte dar, wie die Authentisierung im Netzwerk in nichtbiometrischen Authentisierungssystemen erfolgt. Dort wird mit Hilfe von kryptografischen Verfahren geprüft, ob ein Kommunikationspartner ein gemeinsames Geheimnis kennt. Dabei wird aus den Kennwörtern ein kryptografischer Schlüssel abgeleitet, mit denen Nachrichten zwischen den Partnern verschlüsselt werden. Die Fähigkeit eine Nachricht zu entschlüsseln authentisiert den Kommunikationsparter (Benutzer).

Die biometrischen Templates einer Person stimmen jedoch nicht exakt miteinander überein. Die Ableitung von kryptografischen Schlüsseln aus den biometrischen Templates einer Person hätte unterschiedliche Schlüssel zur Folge. Die Nachrichten, die bei einem Authentisierungsvorgang mit dem kryptografischen Schlüssel, der aus dem neuerstellten biometrischen Template abgeleitet wurde, erzeugt werden, können vom Kommunikationspartner nicht entschlüsselt werden, da die Schlüssel durch die geringfügig voneinander abweichenden biometrischen Templates nicht übereinstimmen werden. Eine Authentisierung über das Netzwerk ist mit diesem Verfahren so nicht möglich.

Die Erzeugung kryptografischer Schlüssel müsste ebenfalls durch Einbeziehung einer Toleranz geschehen, um aus geringfügig abweichenden biometrischen Templates eindeutige Schlüssel zu erzeugen. Genau dann tritt aber ein weiteres Problem auf.

Die Sicherheit eines kryptografischen Verfahrens beruht nicht auf der Geheimhaltung des Verfahrens, sondern auf der Geheimhaltung der Schlüssel. Werden die Muster der Iris benutzt, um einen kryptografischen Schlüssel abzuleiten, so ist dieser nicht geheim, wenn das Ableitungsverfahren bekannt sein sollte, da es sich bei der Iris um ein öffentlich sichtbares biometrisches Merkmal handelt. Sie kann mit Spezialkameras in der Öffentlichkeit aufgenommen und zur Ableitung von kryptografischen Schlüsseln für Angriffe missbraucht werden.

Die Authentisierung im Netzwerk muss also mit anderen Mechanismen erfolgen. Hier wäre die Etablierung eines sicheren Kanals mit Hilfe von kryptografischen Protokollen eine Möglichkeit. Das biometrische Iristemplate wird danach nur noch im abgesicherten Kanal zum Authentisierungsserver übertragen.

Kapitel 5 setzte sich mit der Erstellung von Konzepten zur Implementierung einer Iris-Biometrik in ein Client-Server-Authentisierungssystem auseinander. Bei der Diskussion, ob die Iris als alleiniges Merkmal zur Authentisierung eingesetzt werden sollte, ergab sich, dass diese Vorgehensweise beim heutigen Stand der Technik vermieden werden sollte. Die Iris ist ein physiologisches biometrisches Merkmal, deren Muster während des gesamten Lebens konstant bleiben. Physiologische biometrische Merkmale sind bei Lebewesen aber nur begrenzt vor-

handen. Der Mensch besitzt zwei Iriden, aus denen biometrische Iristemplates zur Authentisierung erzeugt werden können. Gelingt es einem Angreifer die Iris einer Person aufzunehmen oder zu imitieren und während der Authentisierung das Authentisierungssystem damit zu überlisten, so muss ein neues biometrisches Template der anderen Iris der gleichen Person erstellt werden. Nach einem zweiten erfolgreichen Angriff kann kein unbekanntes biometrisches Template der Irismehr erstellt werden, falls als Eingangsdaten nur das biometrische Sample verwendet wird. Darum sollten weitere Eingangsdaten zur Authentisierung wie Kennwörter oder Schlüssel benutzt werden.

Erfolgt die Einbeziehung von Kennwörtern oder kryptografischen Schlüsseln direkt in die Templaterzeugung, so tritt bei einem Fehlschlagen der Authentisierung ein Grundsatzproblem auf, falls das Kennwort nicht extra verifiziert wird. Bei einer fehlgeschlagenen Authentisierung könnte nicht mehr entschieden werden, ob die Authentisierung wegen eines falschen Kennworts, Schlüssels oder der fehlgeschlagenen Verifikation einer Iris entstanden ist. Wird dieser Umstand durch ein zusätzliches Abspeichern des Kennworts und dessen Verifikation im Authentisierungsvorgang und einer Rückmeldung bei einer fehlgeschlagenen Kennwortverifikation behoben, besteht allerdings die Gefahr, dass ein Angreifer die eventuellen technischen Schwächen des Systems ausnutzt und durch „Brute Force Angriffe“ das Kennwort knackt. Allerdings kann nun bei Bekanntwerden des Angriffs durch das Wechseln des Kennworts oder kryptografischen Schlüssels auch nach einem zweiten erfolgreichen Angriff ein neues benutzerspezifisches biometrisches Template der gleichen Iris erzeugt werden.

Um zusätzliche Kennwörter oder kryptografische Schlüssel zu vermeiden, müssten Verfahren integriert werden, die eine zuverlässige Lebenderkennung gewährleisten. Diese Techniken sind aber noch nicht ausgereift und in Erprobung. Darum wurde in die Konzepte zur Implementierung einer Iris-Biometrik ebenfalls ein Kennwort in die Authentisierung einbezogen.

Im Zusammenhang mit der Konzeption zur Implementierung der Iris-Biometrik wurde ein Modularisierungskonzept erarbeitet, das vier Module definiert: Iris-Identifikations- und Authentisierungsmodul (IRINA), Biometrisches Funktionsmodul (BFM), Datentransfermodul (DTM) und Referenzdatenbankmanagementmodul (REFDBM). Das IRINA-Modul stellt Applikationen die Möglichkeit zur Verfügung einen Benutzer mit Hilfe eines Iris-Biometrik-Verfahrens zu authentisieren. Dazu stellt es eine Funktion zur Erfassung der Authentisierungsmerkmale und zur Einleitung des Authentisierungsvorgangs zur Verfügung. Es nutzt dazu die Funktionen des BFM-, DTM-, und REFDBM Moduls. Das BFM-Modul stellt Funktionen zur Erzeugung und zum Vergleich von Iristemplates zur Verfügung. Das DTM-Modul enthält das Protokoll, das eine Authentisierung über das Netzwerk ermöglicht. Das REFDBM-Modul wird für die Verwaltung

biometrischer Referenzdaten verwendet. Die Gesamtheit der Module stellt eine Art eigenes biometrisches Authentisierungssystem dar. Die Integration in nichtbiometrische Authentisierungssysteme kann auf zwei Arten erfolgen: „Vorschaltung und Verkettung“ und „Einbettung und Integration“.

Bei der Implementierung durch „Vorschaltung und Verkettung“ wird das biometrische Authentisierungssystem dem nichtbiometrischen vorgeschaltet. Die Datenerfassungskomponente des nichtbiometrischen Authentisierungssystems wird abgeschaltet und durch eine neue des biometrischen Authentisierungssystems ersetzt. Es erfolgt zuerst eine biometrische Authentisierung am biometrischen Authentisierungssystem und bei Erfolg eine weitere Authentisierung am nichtbiometrischen Authentisierungssystem. Dazu wird ein Zugangsschlüssel verwendet, der bei der erfolgreichen Authentisierung am biometrischen Authentisierungssystem empfangen wird.

Bei der Implementierung durch “Einbettung und Integration“ wird das biometrische Authentisierungssystem direkt in das nichtbiometrische integriert. Die einzelnen Komponenten des nichtbiometrischen Authentisierungssystems werden durch die des biometrischen ersetzt oder ergänzt.

Die Implementation durch „Einbettung und Integration“ kann sich je nach Dokumentationsgrad als sehr viel komplexer als die „Implementation durch Vorschaltung und Verkettung“ gestalten.

Die Implementierung durch „Vorschaltung und Verkettung“ konnte in Windows 2000 gezeigt werden. Die im Modularisierungskonzept definierten Module wurden dazu prototypisch implementiert.

Zukünftige Arbeiten im Bereich der Implementierung von Iris-Biometrik-Verfahren sollten sich mit Mechanismen zur Lebenderkennung und der Biometrik im Zusammenhang mit kryptografischen Verfahren widmen. Weiterhin wäre die Implementierung in Windows XP und die Integration in die .NET Dienste von Microsoft interessant.

Literaturverzeichnis

[Albrecht 1999] Albrecht, A.: *„Biometrie, Digitale Signatur und Elektronische Bankgeschäfte zum Nutzen der Verbraucher“*, Arbeitsgemeinschaft der Verbraucherverbände, 1999

[Bleumer 1999] Bleumer, G.: „Biometrische Ausweise“. In: *Datenschutz und Datensicherheit*, 3(23), 1999

[Behrens Roth 2000] Behrens, M., Roth, R.: „Sind wir zu vermessen, die PIN zu vergessen?“, *Datenschutz und Datensicherheit*, 6(24), 2000

[Bellovin 1989] Bellovin, S., M.: „Security Problems in the TCP/IP Protocol Suite“, *Computer Communication Review*, Vol. 19, No. 2, 1989

[Brause 2001] Brause, R.: *„Betriebssysteme Grundlagen und Konzepte“*, 2. überarbeitete Auflage, Springer Verlag, 2001

[Brömme et al 2001-2] Brömme, A., Kronberg, M., Ellenbeck, O., Kasch, O.: *„A Conceptual Framework for Testing Biometric Algorithms within Operating Systems Authentication“*, Proceedings of the ACM, SAC 2002, Madrid , Spain

[Brömme 2002] Brömme, A.,: *„A Classification of Biometric Applications Wanted by Politics: Passports, Person Tracking and Fight against Terror“*, Faculty of Informatics, University of Hamburg, Germany

[Brunnstein 1991] Brunnstein, Klaus: *„Computer-Viren-Report: Gefahren Wirkung Früherkennung Vorsorge“*, 2., aktualisierte und erweiterte Auflage, Verlag Wirtschaft, Recht und Steuern, 1991

[Büllingen Hillebrandt 2000] Büllingen, F., Hillebrandt A.: „Biometrie als Teil der Sicherungsinfrastruktur“, *Datenschutz und Datensicherheit*, 6(24), 2000

[Chedekel 1995] Chedekel, M.R.: *„Photophysics and photochemistry of melanin“*, In: Melanin: Its Role in Human Photoprotection, Valdenmar: Overland Park, 11-23, 1995

[Colsman 2001] Colsman, K.: *“Implementation und Test eines Sprechererkennungsverfahrens für die biometrische Authentikation“*, Projektbericht im Rahmen des Projektes „Biometrik“ am Arbeitsbereich AGN des FB Informatik der Universität Hamburg, 2001

[Daugman 1994] Daugman, J.: US Patent No. 5,291,560: Biometric Personal Identification System Based on Iris Analysis. Issue Date: 1 March 1994

[Daugman 1998] Daugman, J.: *„How Iris Recognition Works“*, University of Cambridge CB2 3QG, UK, 1998

[Daugman 2001] Daugman, J.: *„Iriserkennung“*, in: Behrens, M., Roth, R. (Hrsg.): Biometrische Identifikation, Datenschutz und Datensicherheit-Fachbeiträge 2001

[Gellert 1999] Gellert, O.: *„Überblick Sicherheitsprobleme“*, Seminar 18.416, Sicherheit in vernetzten Systemen SS99, 1999

[Gollmann 1999] Gollmann, D.: *„Computer Security“*, John Wiley & Sons, Chichester, 1999

[Helden 1995] Helden, J.: „*Verbesserung der Authentifizierung in IT-Systemen durch spezielle Dienste des Betriebssystems*“, Shaker, Aachen, 1995

[Johns 2002] Johns, M.: „*Der Algorithmus zur Iriserkennung von John Daugman*“, Vortrag in Lehrveranstaltung „Aktuelle Probleme der IT und Netzsicherheit“, WS 2001/2002, Universität Hamburg

[Joncheray 1995] Jonchray, L.: „A Simple Active Attack Against TCP“, *Proceeding of the Fifth USENIX UNIX Security* Symposium, USENIX Assoc, 1995

[Köhntopp 1999] Köhntopp, M.: „*Technische Randbedingungen für einen datenschutzgerechten Einsatz biometrischer Verfahren*“, in: Horster, P. (Hrsg.): Sicherheitsinfrastrukturen, Datenschutz und Datensicherheit-Fachbeiträge1999

[Kronberg 1999] Kronberg, M.: „*Keymanagement und Kommunikationsverbindungen*“, Seminar 18.416, Sicherheit in vernetzten Systemen SS99, 1999

[Kronfeld 1962] Kronfeld, P.: „*Gross anatomy and embryology of the eye*“, In: The Eye (Davson, H., Ed.) Academic Press: London, 1962

[Laßmann 1999] Lassmann,G.: “Bewertungskriterien zun Vergleich biometrischer Verfahren“, *Datenschutz und Datensicherheit*, 3(23), 1999

[Microsoft 2000-1] „*Microsoft Windows 2000 Sicherheit*“, Microsoft Press, Unterschleißheim, 2000

[Microsoft 2000-2] „*Microsoft Windows 2000 Server*“, *Die technische Referenz*“, Microsoft Press, Unterschleißheim, 2000

[Microsoft 2000-3] *„Microsoft MSDN Library“*, http://msdn.microsoft.com/

[Morris 1985] Morris, R.: “A Weakness in the 4.2 BSD Unix TCP/IP Software“, *Computing Science Technical Report*, No 117, AT&T Bell Laboratories, 1985

[Morris Thomson 1979] Morris, R., Thomson, K.: „Passwort Security: A Case History“, *Communications of the ACM* 22 (1979), (S.594-597)

[Needham 1978] Needham, R., M., Schroeder, M., D.: „Using Encryption for Authentication in Large Networks of Computers“, Communications of the ACM, 21(12), S. 993-999, 1978

[Pentland et al 2000] Pentland, A., Choudhury, T.: „Face recognition for smart environments“ in: Computer 33 (2), S.50-55, 2000

[Phillips et al 2000] Phillips, P., J., Moon, H., Rizvi, S. A., Rauss, P., J.: „The FERET evaluation methodology for face-recognition algorithms, in: IEEE Trans. Pattern Analysis and Machine Intelligence 22 (10), S.1090-1104, 2000

[Probst 2000] Probst, T.: „Biometrie und Smartcards“, *Datenschutz und Datensicherheit*, 6(24), 2000

[Oppliger 1997] Opplinger, R.: *IT-Sicherheit Grundlagen und Umsetzung*, DUD-Fachbeiträge, Vieweg, 1997

[Rankl 1999] Rankl, W., Effing, W.: *Handbuch der Chipkarten: Aufbau – Funktionsweise Einsatz von Smart Cards*, 3., vollständig überarbeitete und erweiterte Auflage, München, Carl Hanser Verlag, 1999

[RFC 2246] Dierks, T., Allen, C.: „Transport Layer Security Protocol Version 1.0“, Januar 1999, ftp://ftp.ietf.org/rfc/rfc2246.txt

[Russinovich 1998] Russinovich, M.: „Inside Native Applications“, http://www.sysinternals.com/native.htm

[Russinovich 1999] Russinovich, M.: „Inside the Boot Process, Part 2“, http://www.win2000mag.com/Articles/Content/4711_01.html

[Schmidt 2000] Schmidt, J.: „*Windows 2000 Security*“, Markt & Technik Verlag, 2000

[Solomon Russinovich 2000] Solomon, D., Russinovich, M.: „*Inside Microsoft Windows 2000*“, Microsoft Press, 2000

[Schneier 1997] Schneier, B.: „*Applied Cryptography – Protocols, Algorithms ans Source Code in C*“, 2. Auflage, John Wiley & Sons 1997

[Steiner et al 1988] Steiner, J., Neumann, B., C., Schiller J., I.: „*Kerberos: An Authentication Service for Open Network Systems*“, Massachusetts Institute of Technology (MIT), Cambridge, MA, 1988

[Tönnesen 1999] Tönnesen: „Statische und dynamische biometrische Verfahren“, *Datenschutz und Datensicherheit*, 3(23), 1999

[Tanenbaum 1995] Tanenbaum, A., S.: „*Moderne Betriebssysteme*“, 2. Auflage, Carl Hanser Verlag, 1999

[US DoD 1983] US Department of Defense: „Department of Defense trusted computer system evaluation criteria“, CSC-STD-001-83, 1983

[Wirtz 1999] Wirtz, B.: „Biometrische Verfahren“. In: *Datenschutz und Datensicherheit*, 3(23), 1999

[Yong et al 1999] *Yong, Z., Tienju, T., Yunhong,: „Biometric Personal Identification Based on Iris Patterns“, National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing 100080, P.R. China*

[Zhang 2000] Zhang, David, D.: „Automated Biometrics Technologies and Systems“, Kluwer Academic Publishers Group, 2000