INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Erarbeitung eines kryptografischen Modells organisationsweiter Passwortverwaltung am Beispiel des Leibniz-Rechenzentrums

Eva Gembler

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Erarbeitung eines kryptografischen Modells organisationsweiter Passwortverwaltung am Beispiel des Leibniz-Rechenzentrums

Eva Gembler

Aufgabensteller:	Priv. Doz. Dr. Helmut Reiser
Betreuer:	Felix von Eye Priv. Doz. Dr. Wolfgang Hommel
Abgabetermin:	13. September 2012

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 11. September 2012

.....

(Unterschrift des Kandidaten)

Abstract

Die Verwaltung von Passwörtern privilegierter Accounts ist für Unternehmen eine Herausforderung. Oft existieren in größeren Firmen viele privilegierte Accounts, deren Passwörter von den Administratoren geteilt werden müssen. Zudem muss ein Notfall-Zugriff auf diese Accounts möglich sein, in manchen Fällen ist auch eine regelmäßige Änderung der Kennwörter erforderlich. Unter diesen Umständen die Regeln einer guten Passwortsicherheit einzuhalten, ist eine schwierige Aufgabe.

Für das Leibniz-Rechenzentrum (LRZ) ergeben sich diese Probleme ebenfalls. Das Ziel dieser Arbeit ist es, organisationsweite Passwortverwaltung im Hinblick auf das LRZ zu modellieren und kryptografisch abzusichern. Dazu werden die Anforderungen des LRZ an ein gelungenes Passwort-Management zusammengestellt und es wird ein Kriterienkatalog erarbeitet, der diese Anforderungen abdeckt und zur Bewertung von Software für die Passwortverwaltung in Unternehmen verwendet werden kann. Anschließend werden kryptografische Methoden betrachtet, die der Absicherung eines Modells zur Passwortverwaltung dienen können. Mithilfe des kryptografischen Primitivs Attribute-Based Encryption wird außerdem ein theoretisches Modell zur Lösung dieses Problems erstellt.

Abschließend werden zwei Programme zur Passwortverwaltung anhand des erstellten Kriterienkatalogs bewertet.

Inhaltsverzeichnis

1 Einleitung

Vom technischen Aspekt der Informationssicherheit gesehen sind Passwörter eine einfach zu implementierende Maßnahme zum Schutz von sensiblen Systemen und Informationen. Passwörter ermöglichen eine zweifelsfreie Identifizierung und Authentifizierung des Nutzers. Ist das gewählte Passwort stark genug und werden ausreichende Verschlüsselungstechniken genutzt, kann es für einen Angreifer nur schwer zu brechen sein.

Die Problematik entsteht bei der Verwaltung dieser Passwörter durch die Benutzer. Jeder Account erfordert ein eindeutiges Passwort, das zudem komplex sein sollte. Häufig aufgestellte Regeln zur Passwortsicherheit erfordern eine Länge von acht Zeichen oder mehr und eine Mischung aus Buchstaben, Zahlen und Sonderzeichen. Deswegen ist es für Anwender schwer, den Überblick über alle gewählte Passwörter zu behalten. Viele Nutzer wählen deswegen nur leichte Passwörter, die für mehrere Accounts genutzt werden, oder notieren sich diese. Beides ist aus Sicherheitsgründen allerdings nicht zu empfehlen.

Ein besonderes Problem beim Management von Passwörtern ergibt sich für Unternehmen. Eine typische IT-Umgebung besteht aus vielen Komponenten wie beispielsweise Servern und Netzwerkgeräten, deren Zugriff über sogenannte privilegierte Accounts erfolgt. Als privilegierte Accounts bezeichnet man Benutzerkonten mit erweiterten Rechten wie Administrator- oder root-Accounts.

Allein die Anzahl der privilegierten Accounts in einem Unternehmen ist eine Hürde für die Mitarbeiter, zudem müssen diese Passwörter zwischen verschiedenen Administratoren geteilt werden. Oft existieren in einem Unternehmen auch Richtlinien, die einen regelmäßigen Wechsel der Passwörter zur Erhöhung der Sicherheit verlangen, dies wird unter Umständen auch von Sicherheit-Audits wie beispielsweise dem Payment Card Industry Data Security Standard (PCI-DSS) gefordert.

Das erschwert die Verwaltung von Passwörtern für die Administratoren zusätzlich. Es stellt sich die Frage, wie eine solche Anzahl an Zugangskennungen effektiv und ohne Sicherheitslücken verwaltet werden kann.

1.1 Problemstellung

Das Leibniz-Rechenzentrum (LRZ) ist das gemeinsame Rechenzentrum von der Ludwig-Maximilians-Universität München (LMU), der Technischen Universität München (TUM) und der Bayerischen Akademie der Wissenschaften (BAdW). Es stellt IT-Dienstleistungen für die Hochschulen Münchens und der Bayerischen Akademie der Wissenschaften und deren

Studenten bereit, ist für das Münchner Wissenschaftsnetz zuständig und bietet zusätzlich noch Dienste im Bereich Supercomputing an. Das Münchner Wissenschaftsnetz verbindet LMU, TUM, BAdW, die Hochschule München und die Hochschule Weihenstephan-Triesdorf sowie den Großteil der jeweiligen Institutsnetze, anderer wissenschaftlicher Einrichtungen und Studentenwohnheime.[LR11]

Am Leibniz-Rechenzentrum arbeiten rund 150 Mitarbeiter, die für tausende Systeme und Komponenten zuständig sind. Damit steht das Rechenzentrum vor den gleichen Problemen in der Passwortverwaltung, die sich auch für viele größere Unternehmen ergeben.

Im Rahmen dieser Arbeit soll eine mögliche Lösung für das LRZ entwickelt werden. Dabei werden zunächst die allgemeinen Regeln eines guten Passwort-Managements und Eigenschaften typischer Passwort-Tools sowohl für die Nutzung im Privat- als auch im Unternehmensbereich betrachtet. Außerdem wird die spezielle Situation des LRZ genauer untersucht, um darauf aufbauend einen Katalog an Anforderungen für mögliche Lösungen des Problems zu entwickeln. Zudem soll eine theoretische Lösung gefunden werden, die mit Hilfe geeigneter kryptografischer Verfahren den Problemfall abdeckt.

Im Besonderen sollen zumindest folgende Fälle einbezogen werden:

- Im Regelfall müssen mehrere Administratoren mit dem gleichen Passwort auf einen privilegierten Account zugreifen können. Das bedeutet, dass Passwörter zwischen den verschiedenen Mitarbeitern geteilt werden müssen. Es muss also einen einfachen Weg geben, mehreren Personen Zugriff auf das gleiche Passwort zu ermöglichen.
- Eine Lösung muss mit einbeziehen, dass es unter Umständen notwendig ist, Personen, die keine eigentliche Zugriffsberechtigung für einen Account haben, in Notfällen Zugriff auf das Passwort erhalten können. Das macht es auch wichtig, dass zu Auditierungszwecken aufgezeichnet wird, wer wann welches Passwort nutzen konnte.
- Außerdem sollte das Management der Passwörter nicht zu aufwändig für die Administratoren sein.

1.2 Passwortverwaltung

Unter der Verwaltung von Passwörtern kann allgemein der Umgang mit Kennwörtern verstanden werden, also die Erstellung, Speicherung und der Abruf von Passwörtern. Im privaten wie beruflichen Bereich wird das Management von Passwörtern durch die Anzahl an komplexen Passwörtern, die sich ein Nutzer merken muss, notwendig. Der einzelne Nutzer kann die Passwörter mit Hilfe von sogenannten Passwort-Managern oder Passwort-Tools organisieren. Passwort-Manager stellen eine Datenbank bereit, in der die Kennwörter abgespeichert werden können. Der Zugriff auf die Datenbank wird durch die Eingabe eines Master-Passwort möglich. Der Anwender muss sich nur dieses Passwort merken.

In Unternehmen wird die Passwort-Sicherheit durch die erlassenen Richtlinien des Passwort-Managements gesichert. Diese Richtlinien können beispielsweise Vorgaben zur Häufigkeit der Passwortänderung und zur Komplexität des zu wählenden Passworts enthalten.

Auch hier gibt es Softwarelösungen, sowohl Passwort-Tools für den Unternehmensgebrauch als auch Lösungen im Bereich Privileged Password Management. Privileged Password Management konzentriert sich auf die Verwaltung privilegierter Accounts und stellt Mechanismen für das sichere Speichern, Abrufen und Teilen von Passwörtern bereit. Normalerweise gibt es zur Auditierung des Zugriffs auch ein Log, mit dem man nachweisen kann, welcher Benutzer zu welchem Zeitpunkt auf welches Passwort zugegriffen hat, und weitere Funktionen wie etwa die Übernahme der Infrastruktur aus dem Active Directory oder Lightweight Directory Access Protocol.

Speicherung, Abruf und Übertragung von Passwörtern sollten kryptografisch geschehen. Die Kennwörter müssen verschlüsselt abgespeichert werden, dafür gibt es verschiedene Algorithmen wie beispielsweise das häufig gewählte symmetrische Verschlüsselungsverfahren Advanced Encryption Standard (AES). Liegen die Passwörter auf einem Server, muss die Kommunikation mit diesem selbstverständlich ebenfalls verschlüsselt erfolgen.

Im Mittelpunkt dieser Arbeit steht die Passwortverwaltung bei privilegierten Accounts.

1.3 Vorgehensweise und Struktur

In einem ersten Schritt wird in Kapitel 2 eine allgemeine Einführung in das Passwort-Management im Privat- und Unternehmensbereich gestellt. In diesem Kapitel wird erläutert, welche Probleme sich bei der Verwaltung von Passwörtern durch Benutzer und im Bereich von Unternehmen ergeben und was die Grundlagen eines guten Passwort-Managements sind. Es soll ebenfalls dargestellt werden, welche Lösungen für diese Probleme bereits existieren. Dazu wird die Funktionsweise von verschiedenen Software-Lösungen genauer untersucht.

In Kapitel 3 werden Organisationsstruktur und Aufgaben des LRZ genauer dargestellt. Dies dient als Grundlage für die Ermittlung von Anforderungen, die an das Passwort-Management am LRZ gestellt werden. Es wird aufgezeigt, wie ein Modell zur Passwortverwaltung aufgebaut sein und welche Funktionen eine Software-Lösungen bieten muss, um diese Aufgaben zu erfüllen. Ausgehend von den Anforderungen des LRZ soll Schritt für Schritt ein Katalog an Kriterien aufgebaut werden, der diese Erfordernisse erfüllt.

Das Ziel von Kapitel 4 ist es, eine kryptografische Absicherung für das Modell der Passwortverwaltung zu finden. Es wird dargestellt, welche kryptografischen Maßnahmen getroffen werden müssen, um ein solches Modell abzudecken. Ebenfalls wird untersucht, ob es eine theoretische Lösung gibt, die das Passwort-Management in Unternehmen sichern kann.

Abschließend werden in Kapitel 5 verschiedene Programme zum Passwort-Management analysiert. Für eine Bewertung dieser Tools wird der in Kapitel 3 aufgestellte Kriterienkatalog verwendet. Die untersuchten Programme sind die Enterprise Edition mit Enterprise Server von MATESO ([MAT]) und die Password Manager Pro Premium Edition von ManageEngine ([Man]).

2 Passwort-Management

Die Verwaltung von Passwörtern ist sowohl im privaten als auch beruflichen Umfeld von Bedeutung. Im Privatbereich ist es für Personen wichtig, geeignete Kennwörter zu wählen und zu verwalten, um die eigenen Accounts zu schützen. Für Unternehmen ist das Management von Passwörtern erforderlich, um der Gefahr durch externe Angreifer zu begegnen, den Missbrauch von internen Benutzerkonten zu verhindern und sensible Firmendaten zu sichern.

2.1 Passwort-Management aus Anwendersicht

Heutzutage benötigt der Computernutzer viele Passwörter, um Zugang zu Systemen, Internetseiten, ja sogar dem eigenen Bankaccount zu erlangen. Ein Nutzer verwendet Passwörter meist mehrmals am Tag für einfache Aufgaben, vom Einloggen auf dem Computer über die Abfrage der E-Mails bis hin zu Zugriff auf geschützte Programme. Sowohl im beruflichen als auch privaten Umfeld verwaltet eine einzelne Person viele Accounts, der durchschnittliche Nutzer 25 verschiedene Konten, wie eine Studie von Microsoft 2007 zeigte [FH07].

Diese Benutzerkonten sind kontinuierlichen Angriffen ausgesetzt. Das macht es für die Benutzer wichtig, gute Passwörter zu wählen, um ihre Accounts zu schützen. Das Bundesministerium für Sicherheit in der Informationstechnik (BSI) stellt beispielsweise folgende Regeln auf [Buna]:

- Ein Kennwort sollte mindestens über acht Zeichen verfügen, am besten mehr.
- Es sollte nicht nur Buchstaben, sondern auch Zahlen und Sonderzeichen enthalten. Außerdem sollten sowohl Groß- und Kleinbuchstaben genutzt werden.
- Auf gar keinem Fall sollten bekannte Wörter oder Namen verwendet werden, da diese durch einen Wörterbuchangriff leicht zu erraten sind.
- Außerdem sollten auch keine gängigen Tastaturkombinationen wie “qwertz” genutzt werden.

Doch diese Regeln sind Anwendern entweder nicht bekannt oder sie werden nicht angewandt. Wie die Analyse von 32 Millionen gehackten und anschließend im Internet veröffentlichten Passwörtern der Internetseite rockyou.com zeigte, wählen 50% der Nutzer ein leichtes Passwort, das aus einem Namen, einem einfachem Wort oder trivialen Buchstaben- oder Zifferfolgen bestand. Immerhin 1% aller Nutzer griffen auf das gleiche Passwort “123456” zurück [Imp10]. Diese Passwörter sind von einem Angreifer leicht durch einen Brute-Force-Angriff zu brechen und bieten kaum Sicherheit. Nutzer wählen meist einfache Passwörter,

weil diese leicht zu merken sind. Je einfacher jedoch das Passwort, desto schneller ist es aber auch für einen Angreifer zu brechen.

Das BSI hat außerdem Regeln erstellt, wie die Sicherheit eines Passworts am besten zu gewährleisten ist [Buna].

- Passwörter sollten nicht aufgeschrieben oder an Dritte weitergegeben werden. Außerdem sollte ein Passwort niemals per E-Mail versendet werden, da E-Mails in der Regel unverschlüsselt über das Internet versendet werden und ein unbefugter Dritter damit das Passwort einfach herauslesen kann.
- Es sollten auf jeden Fall unterschiedliche Passwörter genutzt werden. Ansonsten erhält ein Angreifer, der sich das Passwort für einen Account beschafft hat, Zugriff auf alle Konten, für die das gleiche Passwort verwendet wurde.
- Wenn eine Anwendung ein bereits eingestelltes Passwort besitzt, ist dieses Passwort auf jedem Fall vom Nutzer zu ändern.
- Oft wird auch empfohlen, Passwörter in regelmäßigen Abständen zu ändern.

Aber auch diese Regeln werden im Normalfall nicht eingehalten. Bei vielen Nutzern ist das Sicherheitsbewusstsein nicht stark ausgeprägt, wie eine Studie von Adams und Sasse über Passwortsicherheit in Unternehmen zeigte [AS99]. Diese Studie zeigte deutlich, dass Regeln zur Passwortsicherheit entweder nicht bekannt sind oder nicht befolgt werden. Viele der befragten Mitarbeiter des Unternehmens hatten keine genaue Vorstellung davon, wie ein sicheres Passwort aussieht und wählten als Folge davon schlechte und für einen Angreifer einfach zu erratende Passwörter. Auch von der Vorgabe, das Kennwort regelmäßig zu ändern, zeigten sich die Mitarbeiter überfordert, sie gingen dazu über, entweder schwache Passwörter zu wählen, diese aufzuschreiben oder diese Kollegen mitzuteilen.

Dass ein Passwort, das für alle lesbar auf einem Post-It am Monitor steckt, keine zufriedenstellende Sicherheit bietet, dürfte offensichtlich sein. Auch ein Zettel mit Passwörtern, der herumgetragen wird, kann schnell verloren gehen und mit ihm alle Zugangsdaten zu den Accounts. Das Passwort mit weiteren Personen zu teilen, stellt ebenfalls ein Sicherheitsrisiko dar.

Ein weiterer oft gewählter Ausweg aus dem Dilemma ist das Abspeichern von Passwörtern auf dem Computer. Dabei werden alle Kennwörter in einer Datei abgelegt, um einfachen Zugriff darauf zu haben Ist diese Datei nicht verschlüsselt, kann jeder mit Zugang zu dem PC darauf zugreifen, was ebenfalls ein Einfallstor für Angreifer sein kann.

Aus dieser Problematik hat sich der Bedarf nach einer professionellen Lösung ergeben. Diese Lösung wird von Passwort-Managern geboten.

2.1.1 Passwort-Manager im privaten Bereich

Für den privaten Bereich gibt es viele verschiedene Passwort-Manager mit unterschiedlichen Funktionalitäten. Alle Passwort-Manager bieten Anwendern die Möglichkeit, ihre gesamten

Kennwörter in einer Datenbank abzuspeichern. Dabei werden im Normalfall nicht nur die Kennwörter, sondern die gesamte Datenbank mit einem passenden Verschlüsselungsalgorithmus verschlüsselt. Zugriff auf die gespeicherten Daten ist nur mit dem Master-Passwort möglich; dieses Passwort ist auch das einzige, dass sich der Anwender merken muss.

Außerdem können Passwort-Manager den Benutzer auf einer Webseite einloggen, dazu tragen sie automatisch den entsprechenden Benutzernamen und das Passwort ein. Das befreit den Anwender davon, jedes einzelne Passwort nachschauen zu müssen.

In vielen Passwort-Managern ist auch ein Passwort-Generator enthalten. Dieser Generator kann automatisch zufällige Passwörter für den Gebrauch erstellen. Der Nutzer kann angeben, wie stark das Kennwort sein soll und hat dafür verschiedene Optionen zur Auswahl, beispielsweise die Länge des zu generierenden Kennworts. Das erstellte Passwort wird dann in der Datenbank gespeichert. Das erlaubt es den Anwendern, ohne Mühe komplexe Passwörter zu erstellen und zu nutzen.

2.1.2 Vorteile von Passwort-Managern

Der größte Vorteil bei der Nutzung eines Passwort-Managers entsteht selbstverständlich durch die Tatsache, dass sich der Nutzer nur noch ein Passwort merken muss, um auf alle Kennwörter zugreifen zu können. Die Möglichkeit des automatischen Einloggens auf Webseiten eliminiert außerdem den Bedarf, alle Passwörter im Kopf haben zu müssen. Da sich ein einzelnes Passwort gut einprägen lässt, steigt auch die Chance, dass der Anwender ein entsprechend komplexes und damit sicheres Passwort wählt.

Die Verwaltung der Passwörter durch den Passwort-Manager ist zudem auch sehr sicher. Die Datenbank mit den Passwörtern ist verschlüsselt und damit vor unberechtigtem Zugriff geschützt, was die Geheimhaltung der Kennwörter sicherstellt. Außer dem Anwender selber sollte im Regelfall niemand auf die Datenbank zugreifen können.

Bessere Sicherheit ist ebenfalls dadurch gewährleistet, dass viele solcher Tools die Erstellung der Passwörter für den Nutzer übernehmen. Damit ist sichergestellt, dass ausreichend komplexe Passwörter für den Gebrauch erstellt und genutzt werden.

Auch einige Angriffe auf die Passwörter durch externe Angreifer werden erschwert. Ein Passwort-Manager kann automatisch die URL einer Webseite überprüfen, was Phishing durch gefälschte Adressen erschwert. Das automatische Einloggen durch das Programm verhindert ebenfalls Keylogging-Angriffe, bei denen die Passwort-Eingabe durch den Nutzer vom Angreifer mitprotokolliert wird.

Auch die Organisation der verschiedenen Nutzeraccounts und zugehörigen Passwörter fällt dem Nutzer leichter, da ein Passwort-Manager eine bessere Übersicht über die verwendeten Kennwörter bietet. Ein solches Tool ist in der Regel auch einfach zu installieren und zu bedienen und damit eine leichte Lösung für privates Passwort-Management.

2.1.3 Nachteile von Passwort-Managern

Der Anwender wird also durch die Nutzung eines Passwort-Managers deutlich entlastet. Diese Lösung hat neben den Vorteilen aber auch einige Risiken. Ein Nachteil ergibt sich aus der Tatsache, dass der Anwender vom Funktionieren der Datenbank abhängig ist. Zum einen benötigt er dauerhaften Zugriff auf den Passwort-Manager, um die gespeicherten Passwörter einsehen und nutzen zu können. Verliert er den Zugang zu der Datenbank, beispielsweise weil diese auf einem verloren gegangenen USB-Stick gesichert war, sind die Passwörter weg. Liegt die Datenbank etwa auf einem Server, muss der Anwender unter allen Umständen darauf zugreifen können.

Zum anderen besteht das Risiko einer Schädigung von Hardware oder Software, was ebenfalls zum Verlust der gespeicherten Daten führen kann. Funktioniert die Datenbank nicht mehr, weil ein Fehler in der Software aufgetreten ist, hat der Nutzer keinen Zugriff mehr auf die Passwörter und die damit verbundenen Accounts.

Das gleiche Problem tritt auf, wenn das Master-Passwort vergessen wird. Da der Zugriff auf den Passwort-Manager äußerst gut gesichert sein muss, gibt es für den Fall des Verlustes des Master-Passworts keine Möglichkeit, dieses zentrale Kennwort wiederherzustellen. Auch in diesem Fall gehen alle Kennwörter verloren.

Außerdem muss der Nutzer darauf achten, dieses Master-Passwort gut zu wählen und die Regeln der Passwort-Sicherheit zu beachten. Gelingt es einem Angreifer, das Passwort zu brechen, erhält dieser Zugriff auf alle Kennwörter und die zugehörigen Accounts. Die Sicherheit der Datenbank hängt also auch von der Bereitschaft und der Fähigkeit des Nutzers ab, ein sicheres Kennwort zu erstellen und zu gebrauchen.

Die Gefahr, die entsteht, wenn ein Angreifer Zugriff auf die Datenbank erhalten hat, ist zudem beachtlich. In dem Fall kommt es zu einer umfassenden Kompromittierung aller Kennwörter und Accounts, da allein das Master-Passwort ausreicht, um Zugang zu allen gesicherten Daten zu erhalten und es keine weiteren Sicherheitsmaßnahmen gibt, die diese Daten schützen würden.

Passwort-Manager für den privaten Gebrauch bieten also für den Anwender einige Vorteile, bergen aber auch Gefahren. Die einfache und sichere Verwaltung der Passwörter wird durch ein solches Tool gewährleistet, aber der Nutzer muss sich bewusst machen, dass die Verwendung auch einige Risiken mit sich trägt und entsprechend handeln, also beispielsweise ein sehr sicheres Master-Passwort benutzen und eine Sicherheitskopie der Datenbank erstellen. Für die private Nutzung ist ein Passwort-Manager aber insgesamt betrachtet eine gute Alternative.

2.2 Passwort-Management im Unternehmen

Viele Unternehmen implementieren Passwörter als Sicherheitsmaßnahme. Zum Beispiel werden Kennwörter benötigt, um sich auf einem Computer einzuloggen oder um Zugriff auf bestimmte Systeme und Software zu erhalten. Die aufgezeigten Probleme, die Nutzer mit der

Verwaltung von Passwörtern haben, bedeuten aber, dass das Management von Kennwörtern für Firmen problematisch sein kann. Im Besonderen verwenden Mitarbeiter oft schwache Passwörter, nutzen gleiche Passwörter für unterschiedliche Accounts, teilen diese gerne mit Kollegen und verfügen allgemein über mangelhaftes Sicherheitsbewusstsein.

Allerdings sind Unternehmen ständigen Bedrohungen durch externe Angreifer ausgesetzt. Große Unternehmen haben jährlich im Durchschnitt 54 ernstzunehmende Angriffe von außerhalb [PW12]. Die Gefahr, die durch einen leichtsinnigen Umgang der Angestellten mit ihren Passwörtern entsteht, ist also nicht zu unterschätzen. Zusätzlich zu externen Angriffen kann Firmen auch ein Problem durch interne Mitarbeiter entstehen. Immerhin 21% aller Unternehmen waren schon Opfer eines Angriffs durch interne Mitarbeiter. 57% der Vorfälle waren durch den Versuch eines internen Nutzers entstanden, Zugriff auf Informationen oder Systeme zu erhalten, für die keine Autorisierung vorlag. 43% der Zwischenfälle waren auf Accounts zurückzuführen, die trotz eines Ausscheiden des Mitarbeiters aus der Firma nicht gelöscht wurden [Fis01].

Interne Mitarbeiter, die unautorisiert auf Ressourcen zugreifen können, stellen damit ein ernstzunehmendes Sicherheitsrisiko dar. Um wichtige Daten und Systeme zu schützen, ist es also erforderlich, den Zugriff von Mitarbeitern auf die Passwörter und Ressourcen zu beschränken, die sie für die Ausführung ihrer Arbeit benötigen.

Zum geeigneten Management der Passwörter erlassen die Unternehmen im Normalfall Richtlinien, die Vorgaben zur Erstellung, Speicherung und Übertragung von Passwörtern enthalten. Die Richtlinien geben beispielsweise vor, wie komplex ein Passwort sein muss, ob und wie oft eine Änderung der Kennwörter erfolgen muss oder dass Passwörter selbst im Intranet nicht unverschlüsselt übertragen werden sollten. Eine Liste an typischen Richtlinien zur Regelung des Passwortgebrauchs findet sich beispielsweise auf der Webseite des BSI [Bunb].

Die Richtlinien und ihre Umsetzung ist auch im Bereich von Sicherheit-Audits von Bedeutung. Manche Regelwerke wie beispielsweise der Payment Card Industry Data Security Standard (PCI-DSS) für Unternehmen, die Kreditkarten-Informationen speichern oder Transaktionen abwickeln, haben Vorgaben an das Passwort-Management innerhalb der Firma.

Eine besondere Problematik bei dem Umgang mit Passwörtern in Firmen entsteht bei privilegierten Accounts. Diese Benutzerkonten mit weitreichenden Rechten erlauben einem Nutzer den vollen Zugriff auf Systeme und die dort gespeicherten Informationen. Deswegen sind privilegierte Accounts begehrtes Ziel externer Angreifer, was bedeutet, dass die Regeln einer guten Passwort-Sicherheit für privilegierte Accounts im besonderen Maße gelten. Außerdem ist die Kennwortverwaltung in diesem Fall erschwert, da diese Konten von mehreren Administratoren genutzt werden, die deswegen die Passwörter untereinander teilen müssen. Zudem gibt es in größeren Unternehmen Tausende an Komponenten, die über diese Accounts verwaltet werden müssen.

Für die Verwaltung von privilegierten Accounts gilt also:

- Das sichere Teilen der Passwörter zwischen den Administratoren muss möglich sein.
- Ein Administrator sollte nur Zugriff auf die Passwörter haben, die er für seine Arbeit benötigt.

- Die Rechenschaftspflichtigkeit der Administratoren muss gegeben sein. Man sollte nachweisen können, wer auf welchem Account welche Aktionen durchgeführt hat.
- Gleichzeitig muss gewährleistet werden, dass für die privilegierten Accounts sichere Passwörter gewählt werden.

2.2.1 Nutzung von privaten Passwort-Managern im Unternehmen

Die Vor- und Nachteile für den einzelnen Anwender wurden schon ausführlich dargestellt. Eine weitere Frage ist, ob ein solches Programm auch in größeren Unternehmen zur Verwaltung der Kennwörter privilegierter Accounts verwendet werden kann.

In diesem Falle sind zwei Möglichkeiten denkbar. Entweder ein einzelner Mitarbeiter erstellt sich selbst eine Datenbank zur Verwaltung der eigenen Passwörter oder es gibt für einen gewissen Personenkreis eine zentrale Datenbank, über die die Kennwörter geteilt werden.

Bei der ersten Alternative tritt neben den genannten Nachteilen von Passwort-Managern für den privaten Gebrauch zusätzlich das Problem auf, dass die Kennwörter für die privilegierten Accounts zwischen den Mitarbeitern geteilt werden müssen. Diese Möglichkeit ist durch Passwort-Manager allerdings nicht gegeben. Ändert ein Mitarbeiter beispielsweise das Passwort für einen Account, müsste er dieses neue Passwort allen zukommen lassen, die ebenfalls Zugriff auf diesen Account haben. Das organisatorische Chaos für einen solchen Fall ist leicht vorstellbar.

Andernfalls könnte eine zentrale Datenbank erstellt werden, deren Master-Passwort zwischen verschiedenen Personen geteilt wird. In diesem Fall stellt der Abgleich der Passwörter kein Problem dar. Eine Lösung könnte folgendermaßen aussehen:

- Alle Mitarbeiter haben Zugriffsberechtigung auf die Datenbank. Damit bleibt aber das Problem bestehen, dass jeder Zugriff auf alle Kennwörter hat. Jeder Nutzer kann alle Passwörter einsehen und hat damit Zugang zu allen Accounts, ebenso kann jeder Passwörter und zugehörige Konten erstellen, ändern und löschen. Die Gefahr des Missbrauchs durch die Mitarbeiter ist also sehr hoch, eine solche Lösung ist damit aufgrund der entstehenden Sicherheitsrisiken nicht empfehlenswert.
- Die Datenbank kann nur durch einen bestimmten Personenkreis, beispielsweise einem bestimmten IT-Team mit gleichen Zugriffsberechtigungen für Accounts, genutzt werden. Die Zusammenstellung dieses Personenkreises dürfte aber Probleme bereiten, da sichergestellt werden muss, dass jeder auf alle Passwörter zugreifen kann, die er für seine Arbeit benötigt. Gesetzt den Fall, dass dies aufgrund der Unternehmensstruktur möglich ist, beseitigt diese Lösung immer noch nicht bestehende Sicherheitsprobleme. Es fehlt die Möglichkeit, den Zugriff der Mitarbeiter auf die Passwörter nachverfolgen zu können. So können Mitarbeiter etwa problemlos Passwörter aus der Datenbank löschen, ohne dass eine solche Aktion auffällt. Auch ein Mitarbeiter, der das Master-Passwort für den Zugriff ändert und damit die Datenbank für die anderen Personen unbenutzbar macht, stellt ein Problem dar.

Das größte Problem bei einer möglichen Verwendung von solchen Passwort-Managern besteht also darin, dass es nur ein einziges Benutzerkonto gibt und der Zugriff eines Benutzers auf die Datenbank nicht beschränkt werden kann. Außerdem lassen Passwort-Manager Funktionen vermissen, die für viele Firmen wichtig sind. Die Problematik der Auditierung des Zugriffes wurde bereits angesprochen, zudem ist für Unternehmen meist noch wichtig, Richtlinien zum Passwortgebrauch erstellen und umsetzen zu können. Auch für Sicherheit-Audits ist eine Lösung über private Passwort-Manager ungenügend.

Im Bereich von Unternehmen lassen sich diese Tools also nicht erfolgreich einsetzen. Die organisatorischen Probleme allein sind beträchtlich, auch vom Sicherheitsstandpunkt her ist das nicht zu empfehlen. Aufgrund dessen wurden spezielle Software-Lösungen für den Einsatz in Unternehmen entwickelt, die auf die besonderen Bedürfnisse von Firmen beim Passwort-Management achten. Die Spanne reicht von Enterprise Password Managern bis hin zu Privileged Password Management Lösungen.

2.2.2 Enterprise Password Manager

Programme in diesem Bereich bieten viele unterschiedliche Funktionen an. Allgemein beschränken Enterprise Password Manager den Zugriff der Nutzer auf die Passwörter, die von ihm benötigt werden. Deswegen bieten diese Programme Mehrbenutzer-Betrieb an, bei dem mehrere Benutzer Zugriff auf eine Datenbank bekommen. Manche Enterprise Password Manager erlauben die Erstellung von Gruppen. Das erleichtert die Verwaltung der Kennwörter, da die Übertragung von Zugriffsberechtigungen auf Basis dieser Gruppen erfolgen kann. Die Vergabe von Zugriffsberechtigungen macht es möglich, nur einzelnen Nutzern oder bestimmten Gruppen Zugang zu den Passwörtern zu erlauben, was die Vertraulichkeit der geschützten Daten sichert.

Die Gruppen- und Nutzerstruktur kann aus bestehenden Verzeichnisdiensten wie beispielsweise Active Directory bei Windows übernommen werden. Manche Enterprise Password Manager erlauben es, die Authentifizierung der Nutzer über die Verzeichnisdienste zu ermöglichen.

Wichtig für viele Unternehmen ist auch die geordnete Freigabe von Passwörtern. Benötigt ein Mitarbeiter Zugriff auf ein Passwort, muss die Vergabe der Berechtigung geeignet autorisiert werden. Im Normalfall sind eine oder mehrere Personen für die Autorisierung des Zugriffs zuständig. Der Benutzer muss eine Anfrage stellen, die verantwortlichen Personen werden darüber informiert und müssen entsprechend über die Anfrage entscheiden. Dies ist besonders wichtig in Notfällen, beispielsweise wenn auf einen bestimmten Account zugegriffen werden muss, der zuständige Administrator aber nicht verfügbar ist.

Auditierungsfunktionen sind ebenfalls von Bedeutung. Oft ermöglichen Enterprise Password Manager es, in einem Log verschiedene Daten aufzuzeichnen, zum Beispiel wer welche Anfragen gestellt oder wer wann ein Passwort geändert hat.

Eine weitere Funktionalität, die viele Enterprise Password Manager anbieten, ist die Erstellung von Richtlinien für die Nutzung von Passwörtern. Beispielsweise können nur Passwörter

angelegt werden, die bestimmten Regeln genügen. Richtlinien können auch regeln, wie oft ein Passwort geändert werden muss.

2.2.3 Privileged Password Management

Privileged Password Management wird oft auch als Privileged Account Management bezeichnet. Allgemein bezieht sich der Begriff auf Software-Lösungen, bei denen die Verwaltung von privilegierten Accounts und den zugehörigen Passwörtern im Mittelpunkt steht. Privileged Password Management geht dabei über die Funktionen, wie sie Enterprise Password Manager bieten, hinaus.

Privileged Password Management ist oft in Privileged Identity Management Suiten integriert. Identity Management befasst sich mit der Verwaltung von Identitäten und deren Berechtigungen in einem bestimmten System. Dabei soll der Benutzer zu jedem Zeitpunkt eindeutig identifizierbar sein. Beim Privileged Identity Management steht entsprechend das Verwalten von Mitarbeitern mit erweiterten Rechten im Mittelpunkt.

Privileged Password Management ist eine komplexe Lösung mit einem entsprechend hohen Preis. Das Ziel ist es, Passwort-Management auch in sehr großen Unternehmen zu ermöglichen. Dazu wird eine Vielzahl an verschiedenen Systemen unterstützt.

Zur Anmeldung am System ist in der Regel auch Multi-Factor-Authentication möglich. Bei Multi-Factor-Authentication wird eine Kombination mehrere Faktoren zur Authentifizierung eines Benutzers verwendet. Zum Beispiel benötigt ein Benutzer neben Name und Passwort auch noch ein Token zur Anmeldung.

Mehr Sicherheit versprechen manche Hersteller auch dadurch, dass die Kennwörter automatisch durch die Software in regelmäßigen kurzen Abständen (beispielsweise einmal pro Tag) in neue Zufallswerte geändert werden. Ein Passwort, das für einen Benutzer offen gelegt wurde, ist diesem so nur maximal für einige Stunden bekannt, bevor es durch ein neues ersetzt wird.

Lösungen des Privileged Password Management sind dabei auch in der Lage, Passwort-Änderungen automatisch am Zielsystem durchzuführen, sofern eine Verbindung zu diesem Zielsystem besteht.

Auch die Überwachung der Benutzeraktivität zu Audit- und Forensikzwecken ist bei manchen Privileged Password Management Lösungen ausgeweitet. Neben der Integration verschiedener Programmen zur Überwachung kann es auch die Möglichkeit geben, die Sitzung eines Benutzers aufzunehmen und anschließend abspielen zu lassen.

Unter Umständen erfordert auch die Kommunikation zweier Anwendungen untereinander Kennwörter. Die dazu benötigten Passwörter liegen meist nur in Textform vor oder sind im Code der Anwendung eingebettet. Dadurch können ebenfalls Sicherheitsprobleme entstehen. Privileged Password Management kann die Übermittlung der Passwörter zwischen diesen Anwendungen übernehmen.

3 Entwicklung eines Kriterienkatalogs für die Passwortverwaltung am LRZ

Das LRZ ist sowohl ein Zentrum für technisch-wissenschaftliches Hoch- und Höchstleistungsrechnen als auch IT-Dienstleister für die Hochschulen in München. Es stellt für die verschiedenen Hochschuleinrichtungen IT-Infrastruktur und zentrale Dienste bereit. Zu den Aufgaben des LRZ gehört auch das Management des Münchner Wissenschaftsnetzes, das als Kommunikationsinfrastruktur die Universitäten und andere wissenschaftlicher Einrichtungen in München miteinander verbindet. Außerdem verfügt das LRZ über ein bibliothekarisches Archivierungs- und Bereitstellungssystem zur Digitalisierung und Archivierung von Büchern und betreibt selbst Forschung im Bereich Technische Informatik und Höchstleistungsrechnen.

Die Dienstleistungen des LRZ reichen von der Bereitstellung eines Hochleistungsrechners über zentrale Systeme zur Speicherung und Archivierung von Daten bis zum Betrieb von E-Mail-, Directory- und Webservern. Außerdem bereitgestellt werden Grafik- und Visualisierungssysteme.

Nach dem Selbstverständnis des LRZ umfassen die primären Aufgaben:

- "die Planung, Bereitstellung und Betrieb einer leistungsfähigen Kommunikationsinfrastruktur als Bindeglied zwischen den zentralen und dezentralen Rechnern und als Zugang zu weltweiten Netzen,
- die Planung, Bereitstellung und Betrieb von Rechnern und Spezialgeräten, die wegen ihrer Funktion zentral betrieben werden müssen (z. B. Mailgateway) oder deren Betrieb dezentral nicht wirtschaftlich oder technisch nicht möglich ist (z. B. Hochleistungsrechensysteme, Datensicherung und Archivierung),
- die Beschaffung günstiger Software-Lizenzen über Hochschul-, Campus- oder Landesverträge,
- die Unterstützung und Beratung bei Fragestellungen der Informationsverarbeitung ("Kompetenzzentrum")."[LR10]

3.1 Interne Organisationsstruktur des LRZ

Die Organisation des LRZ ergibt sich aus den dargelegten Aufgaben. Laut Jahresbericht ist das LRZ in vier Abteilungen aufgeteilt: Benutzernahe Dienste und Systeme, Hochleistungssysteme, Kommunikationsnetze und Zentrale Dienste. [LR11]

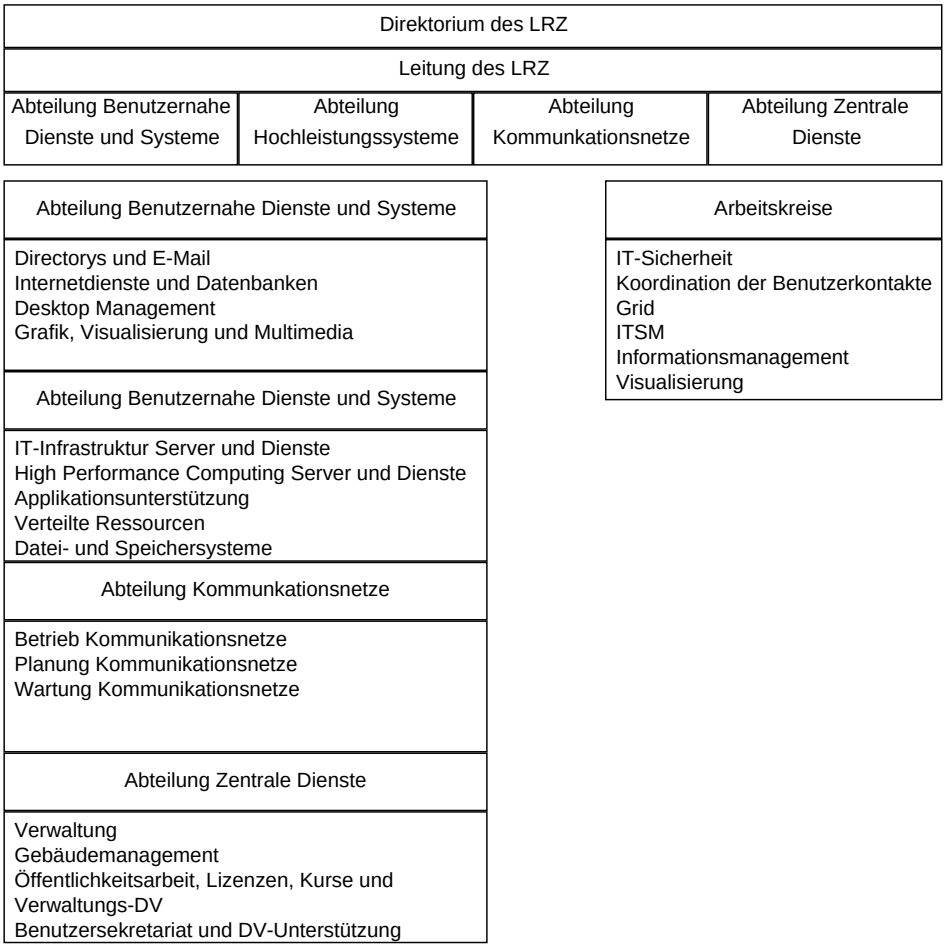

Abbildung 3.1: Die Organisationsstruktur des LRZ

Die Abteilung Benutzernahe Dienste und Systeme ist in die Unterabteilungen Directorys und E-Mail, Internetdienste und Datenbanken, Desktop Management sowie Grafik, Visualisierung und Multimedia gegliedert. Diese Abteilung bietet verschiedene E-Mail-, Web- und Datenbankdienste an und ist zuständig für Benutzerverwaltung und Verzeichnisdienste. Für die Webdienste wie E-Learning-Plattformen müssen Webserver und Hosting-Dienste bereitgestellt werden. Von dieser Abteilung werden auch verschiedene Windows-Server für interne und externe Kunden zur Verfügung gestellt, außerdem Windows-Arbeitsplätze, Terminal-Server und Rechnerpools für Mitarbeiter und Studenten. Ebenfalls in den Zuständigkeitsbereich fällt die Serveradministration und Applikationsunterstützung für die von der Abteilung angebotenen Dienste.

Die Abteilung Hochleistungssysteme besteht aus den fünf Unterabteilungen IT-Infrastruktur Server und Dienste, High Performance Computing Server und Dienste, Applikationsunterstützung, Verteilte Ressourcen sowie Datei- und Speichersysteme. Die Abteilung ist für

den Betrieb des Höchstleistungsrechners und die Hochleistungs-Linux-Systeme zuständig. Es werden Software und User-Support für die Hochleistungssysteme bereitgestellt. Zusätzlich fällt in den Aufgabenbereich dieser Abteilung die Datenhaltung, also Archiv- und Backupsysteme, Möglichkeiten zur Langzeitarchivierung und Online-Speicher wie Network Attached Storage und Storage Area Network, außerdem die Verwaltung verschiedener Linux-Serversysteme.

Die Abteilung Kommunikationsnetze wiederum ist aufgeteilt in die Unterabteilungen Betrieb Kommunikationsnetze, Planung Kommunikationsnetze und Wartung Kommunikationsnetze. Vom LRZ werden das Backbone-Netz für das Münchner Wissenschaftsnetz und die meisten Gebäudenetze der zugehörigen Institute verwaltet. Die physische Infrastruktur wird bereitgestellt und es werden verschiedene Dienste angeboten, um die Nutzung des Netzes zu ermöglichen. Dazu gehört die Bereitstellung von Wireless Local Area Network und Diensten wie beispielsweise Domain Name System und das Dynamic Host Configuration Protocol.

Das Kommunikationsnetz hat besondere Bedürfnisse im Bereich der IT-Sicherheit. Sicherheit muss durch Firewalls, Security Information and Event Management Lösungen und Intrusion Detection and Prevention Systeme und weitere Monitoring-Mechanismen gewährleistet werden.

Die Abteilung Zentrale Dienste schließlich ist zuständig für Verwaltung und Gebäudemanagement. In den Bereich der Abteilung gehört auch Öffentlichkeitsarbeit sowie die Verwaltung von Lizenzen für Kunden an Instituten und Hochschuleinrichtungen. Außerdem werden verschiedene Kurse und Veranstaltungen zu Themen wie bestimmter PC-Software, Hochleistungsrechnen oder Linux organisiert.

Abteilungsübergreifend bestehen mehrere Arbeitskreise. Ein Arbeitskreis ist für die Koordination der Benutzerkontakte zuständig, des Weiteren gibt es noch den Arbeitskreis Visualisierung. Der Arbeitskreis Grid befasst sich mit mehreren Grid-Projekten am LRZ und in München und der Arbeitskreis Informationsmanagement hat das Ziel, die Dokumentations- und Informationsstruktur am LRZ zu vereinheitlichen, um eine bessere Verwaltung von Informationen zu ermöglichen. Der Arbeitskreis IT-Sicherheit ist für die Überwachung und stetige Verbesserung der IT-Sicherheit am LRZ zuständig. Dazu gehört die Entwicklung und Umsetzung von Richtlinien zum Sicherheitsniveau, der Security-Incident-Response-Prozess und das Computer Security Incident Response Team. Das Management von IT-Diensten wird vom Arbeitskreis ITSM überwacht, dessen Aufgabe die Umsetzung von gelungenem IT-Service-Management ist.

Wie man sieht, bietet das LRZ ein breitgefächertes Portfolio an verschiedenen Services an. Insgesamt arbeiten mehr als 150 Mitarbeiter am LRZ, um diese Dienstleistungen zu ermöglichen. Dabei müssen Tausende an Komponenten von den Mitarbeitern administriert werden, allein für das Münchner Wissenschaftsnetz fallen beispielsweise mehrere hundert Router, Switches oder WLAN-Access-Points an.

Dem LRZ stellt sich damit die Herausforderung, das Management der Passwörter für diese IT-Umgebung möglich zu machen. Dabei ergeben sich spezielle Anforderungen für die Kennwortverwaltung.

3.2 Lösungsansätze für die Passwortverwaltung am LRZ

Allgemein lassen sich aus der Verwaltungsstruktur des LRZ einige Anforderungen an das Passwort-Management ableiten. Klar ist, dass die Anzahl an Systemen, privilegierten Accounts und zugehörigen Passwörtern eine automatisierte Lösung nötig macht. Eine automatisierte Lösung ermöglicht es, einen besseren Überblick über die Accounts zu behalten, erlaubt Rechenschaftspflichtigkeit der Administratoren und eine einfache regelmäßige Änderung von Passwörtern.

Es gibt auch Alternativen zu einer solchen Lösung, die aber alle Schwächen besitzen. Möglichkeiten zur Sicherung der Passwörter wären beispielsweise das Hinterlegen eines Passworts in einem Kuvert oder die Aufbewahrung in einem Tresor. Ein Passwort, das auf einen Zettel aufgeschrieben und dann in einem verschlossenen Kuvert verwahrt wird, ist allerdings nicht sicher. Die Kuverts mit den Passwörtern müssten ständig kontrolliert werden, um sicher zu gehen, dass sie noch vorhanden sind und dass niemand das Kuvert geöffnet hat. Diese ständige Kontrolle ist nur schwer umzusetzen.

Die Kuverts in einem Tresor zu verschließen, bedeutet da mehr Sicherheit. Die Schlüssel für diesen Tresor können an mehrere Personen verteilt werden. Dabei ist auch eine mehrstufige Kontrolle des Zugriffs auf die Passwörter denkbar, beispielsweise kann eine Person den Schlüssel für den Raum besitzen, in dem der Tresor ist, eine andere Person hat den Schlüssel für den Tresor selbst. Bei einem solchen Verfahren ist das Unternehmen allerdings darauf angewiesen, dass die Personen mit den Schlüsseln ständig verfügbar sind. Sind sie nicht anwesend, ist kein Zugriff auf die Passwörter möglich.

Das Einsetzen einer Software-Lösung zur Verwaltung der Passwörter ist somit die beste Möglichkeit, da die Alternativen zu viele Nachteile besitzen.

Da das LRZ viele unterschiedliche Dienstleistungen anbietet und viele verschiedene Systeme administriert, von Windows- und Linux-Servern über Datenbanken bis hin zu Routern, muss eine Lösung eine Vielzahl an Systemen verbinden und unterstützen können. Auch muss die Lösung großflächig einsetzbar sein und mit einer großen Anzahl an Passwörtern umgehen können.

Ebenfalls wichtig ist die ständige Verfügbarkeit der Lösung, sodass der Zugriff auf die Passwörter jederzeit möglich ist. Außerdem ist es notwendig, den Zugriff der Benutzer beschränken zu können. Einerseits muss Mehrbenutzerbetrieb möglich sein, andererseits sollte nicht jeder Benutzer alles einsehen können.

Um Zugriffe auf Passwörter nachverfolgen zu können und um internen Missbrauch von privilegierten Accounts möglichst zu verhindern, sollten auch Informationen für verschiedene Auditierungszwecke gesammelt werden. Da das LRZ für viele sensible Systeme zuständig ist, muss eine Lösung zudem verschiedenen Sicherheitsanforderungen genügen. Die Passwörter müssen sowohl gegen externen Zugriff als auch vor internem Missbrauch geschützt werden.

Zudem sollte die Lösung für den Administrator einfach einsetzbar, intuitiv bedienbar und effizient sein.

Ausgehend von diesen sehr allgemeinen Anforderungen wird Schritt für Schritt ein Katalog an Anforderungen und Kriterien ermittelt. Im Besonderen werden verschiedene Funktionen eines Passwort-Management-Systems erläutert. Dabei wird zuerst der Aufbau eines Systems der Passwortverwaltung dargelegt, anschließend werden Anwendungsfälle betrachtet, die auftreten können. Die Verwaltung von Berechtigungen und die Zugriffskontrolle auf die Datenbank werden genauer untersucht (3.2.2), sowohl der Vorgang zur Erstellung von Passwörtern (3.2.3) und als auch zur Autorisierung des temporären Zugriffs im Notfall (3.2.4) sowie der Freigabe der Passwörter (3.2.5). Schlussendlich wird überlegt, wie das Anlegen von Audit-Logs erfolgen kann (3.2.6).

3.2.1 Aufbau des Systems

Aus den grundlegenden Anforderungen ergeben sich weitere Kriterien an ein gewähltes Passwort-Management-System. Das System an sich sollte zumindest folgende Anforderungen erfüllen:

- Passwörter werden zentral gespeichert.
- Das System ist zuverlässig. Das bedeutet einerseits, dass das System stabil läuft und verfügbar ist, die Ausfallzeit also möglichst gering und der Zugriff jederzeit möglich ist. Andererseits sollte das System an sich möglichst fehlerfrei arbeiten.
- Die automatisierte Änderung von Passwörtern ist möglich.
- Verschiedene Sicherheitsanforderungen müssen gegeben sein. Passwörter liegen nicht im Klartext vor oder werden unverschlüsselt übertragen. Der Zugriff auf das System ist nur autorisierten Personen möglich.
- Außerdem ist Mehrbenutzerbetrieb möglich.

Passwörter sollten zentral in einer Datenbank gespeichert sein. Würden sie nur auf einzelnen Rechnern vorliegen, könnte es Probleme geben, wenn dieser Rechner nicht verfügbar ist oder keine Verbindung zum Netzwerk hat. In dem Fall wäre kein Zugriff auf das Passwort möglich. Auch organisatorisch ist es problematisch, wenn ein Passwort nur auf einem bestimmten Rechner gespeichert ist. Deswegen ist eine zentrale Struktur zur Sicherung der Passwörter notwendig.

Der Zugriff auf diese Struktur muss zu jedem Zeitpunkt möglich sein. Das bedeutet, dass es keinen Single Point of Failure in dem Verwaltungssystem geben darf. Die zentrale Datenbank sollte also nicht nur auf einem einzelnen Server vorliegen. Bei diesem Server könnten Probleme auftreten, die den Zugriff nicht möglich machen, außerdem kann es vorkommen, dass die Verbindung zum Server nicht funktioniert. Aber auch in diesen Fällen muss der Zugriff auf die Passwörter möglich sein. Das bedeutet, dass die Datenbank auf mehr als einem Server gesichert sein sollte.

Möglichkeiten zur Sicherung der Daten wären die Bereitstellung eines zweiten Servers, das Anlegen eines Backups und der Betrieb von Virtuellen Maschinen. Wenn Backups angelegt werden, ist es selbstverständlich wichtig, auch diese Daten zu schützen. Allgemein sollte die

Wiederherstellung der Daten nicht zu viel Zeit in Anspruch nehmen. Kurze Zeiträume von wenigen Minuten dürften den Betriebsablauf im Unternehmen wenig einschränken, aber ein längerer Ausfall wäre problematisch.

Werden mehrere Server eingesetzt, ist der zuverlässige Abgleich der Daten zwischen den Servern erforderlich. Zu jedem Zeitpunkt müssen auf den Servern die gleichen Passwörter vorliegen. Das System sollte es also auch ermöglichen, diesen Abgleich vorzunehmen.

Auch bei einer Änderung des Passworts ist der Abgleich der Daten nötig. Das Passwort in der Datenbank und auf dem Zielsystem muss immer übereinstimmen. Das kann bei der Änderung der Passwörter Probleme ergeben, beispielsweise wenn die Lösung das Passwort ändert und dann vergeblich versucht, Verbindung zum Zielsystem aufzubauen, um das neue Passwort zu übermitteln.

In diesem Fall muss das alte Passwort gespeichert bleiben, damit der Zugriff mit dem alten Passwort weiter möglich ist. Das System sollte periodisch versuchen, das neue Passwort zu übermitteln und das Verbindungsproblem an zuständige Personen melden.

Auf jeden Fall ist es erforderlich, alte Passwörter für einen Zeitraum zu speichern. Dies ist nicht nur in dem Fall nötig, wenn der Datenabgleich nicht funktioniert, sondern kann auch wichtig sein, wenn ein System abstürzt und alte Archivdaten zur Wiederherstellung genutzt werden.

Die Speicherung bereits verwendeter Passwörter ist ebenfalls zu empfehlen, um diese bei einem Passwort-Wechsel mit dem neu gewählten Kennwort zu vergleichen. Damit kann ausgeschlossen werden, dass ein altes oder sehr ähnliches Passwort erneut gewählt wird.

Auch eine Regelung der Kommunikation zwischen den verschiedenen Beteiligten im System ist nötig. Die Kommunikation zwischen Benutzer und Lösung, zwischen Zielsystemen und Lösung sowie zwischen den verschiedenen Servern zum Datenabgleich sollte schnell funktionieren, ohne die Sicherheit dabei zu vernachlässigen.

Zudem ist es wichtig, dass die Plattform, auf der die Datenbank vorhanden ist, so sicher wie möglich ist. Am besten ist es, einen dedizierten Server für die Passwörter bereitzustellen. Die Anzahl an Personen, die mit Administratorrechten darauf zugreifen können, sollte so minimal wie möglich sein. Auch ist eine Autorisierung der Personen, die Zugriff auf das System erhalten sollen, nötig, sodass keine unbefugten Personen das System nutzen können.

Die gespeicherten Daten müssen geschützt werden. Wie eine sichere Aufbewahrung der Passwörter aussehen könnte, behandelt Kapitel 4 näher.

Außerdem sollte das System Mehrbenutzerbetrieb erlauben. Sowohl das Anlegen mehrerer Benutzer als auch das Einschränken der Zugriffsberechtigungen muss möglich sein, damit Benutzer nur Einblick in die Daten erhalten, die sie für die Ausführung ihrer Arbeit benötigen. Das bedeutet, dass es eine Verwaltung der Zugriffsberechtigungen geben muss.

3.2.2 Berechtigungsverwaltung und Zugriffskontrolle

Innerhalb eines Systems zur Passwortverwaltung existieren verschiedene Berechtigungen. Darunter fallen die Berechtigungen

- ein Passwort lesen zu können (Leseberechtigung)
- ein Passwort ändern zu können (Schreibberechtigung)
- einen Passwort-Eintrag anlegen und löschen zu können, dies ist verbunden mit der Berechtigung, Zugriffsberechtigungen für diesen Eintrag vergeben zu können
- Auditinformationen einsehen zu können
- Benutzer und Gruppen anlegen zu können
- die Datenbank verwalten zu können

Um die Verwaltung von Berechtigungen zu erleichtern, ist die Existenz von Gruppen oder Rollen wünschenswert. Die Vergabe von Berechtigungen auf Grundlage von Rollen wird Role Based Access Control genannt. Dabei werden den Benutzern eines Systems Rollen zugewiesen und der Zugriff auf die Ressourcen erfolgt anhand dieser Rollen. Ein Benutzer kann dabei mehrere Rollen übernehmen.

Eine mögliche Rollenverteilung, die sich an den oben aufgezählten Berechtigungen orientiert, könnte folgendermaßen aussehen:

- Administratoren haben Lese- und Schreibberechtigung. Die Passwort-Einträge, für die sie Zugriffsberechtigung zugewiesen erhalten haben, können von ihnen eingesehen werden. Sie können aber nicht selber Einträge anlegen oder löschen.
- Das Anlegen und Löschen von Passwort-Einträgen sowie die Vergabe von Berechtigungen fällt in den Aufgabenbereich einer weiteren Rolle. Benutzer mit dieser Rolle verwalten und kontrollieren die Zugriffsberechtigungen für die von ihnen angelegten Passwort-Einträge und können auch temporären Zugriff auf die Passwörter erlauben (siehe 3.2.4). Diese Rolle könnte von den Leitern bestimmter Gruppen übernommen werden oder von System-Hauptverantwortlichen, denen diese Aufgaben übertragen werden.
- Die Administration der Datenbank wird von einer dritten Rolle übernommen. In diesen Bereich würden allgemeine Einstellungen wie beispielsweise die Richtlinien zur Passwort-Vergabe oder die Abstände der automatischen Änderung von Passwörtern fallen. Auch das Anlegen von Gruppen und Benutzern fällt unter die Aufgaben dieser Rolle.
- Die Berechtigung dazu, Audits einsehen zu können, kann an eine eigene Rolle gehen. Auditinformationen können nur von Personen mit dieser Rolle gesehen werden.

Das Verwalten von Berechtigungen wird auch durch Gruppen vereinfacht. Die Berechtigungen können dann auf der Basis von Gruppenzugehörigkeit zugewiesen werden. Das ist

leichter, als jedem einzelnen Benutzer eigene Berechtigungen zuzuweisen, diese zu kontrollieren und bei Bedarf zu ändern (beispielsweise beim Wechsel eines Mitarbeiters in eine andere Abteilung).

Die Gruppen sollten sich dabei nach den benötigten Zugriffsberechtigungen der einzelnen Mitarbeiter richten. Vorteilhaft wäre es, wenn die Gruppen direkt aus Verzeichnisdiensten wie Active Directory oder Lightweight Directory Access Protocol (LDAP) übernommen werden könnten.

Der Zugriff auf die Datenbank muss beschränkt und nur autorisierten Personen möglich sein. Zur Autorisierung eines Benutzers, also zur Vergabe seiner Zugriffsberechtigungen, muss jeder Benutzer authentifiziert werden. Dabei muss sich der Benutzer am System authentisieren, was über die Eingabe von Benutzername und Passwort erfolgt. Möglich ist auch Multi-Factor-Authentication, bei der mehr als eine Methode zur Authentisierung des Benutzers verwendet wird. Wurden Verzeichnisdienste wie Active Directory oder LDAP integriert, kann ein Benutzer sich auch mit den dort genutzten Login-Informationen am System authentisieren.

Die Authentisierung eines Benutzers dient auch dazu, diesen zu identifizieren. Damit können die Aktionen eines Benutzers ihm jederzeit zugeordnet werden.

Allgemein sollte der Zugriff auf die Datenbank nur innerhalb des LRZ-Netzwerks oder über spezielle Gateway-Systeme möglich sein. Das erschwert externen Angreifern den Zugriff zusätzlich. Außerdem sollten Login-Versuche überwacht werden, um bei einer gewissen Anzahl an fehlgeschlagenen Versuchen das Login zu sperren.

3.2.3 Erstellen der Passwörter

Das Anlegen von Passwörtern kann nur durch autorisierte Benutzer erfolgen. Grundsätzlich kann nur eine Person das Passwort für einen Account anlegen, die überhaupt weiß, dass dieser Account existiert und die Zugriff darauf hat. Es könnte eine Möglichkeit sein, das Anlegen von Passwörtern nur bestimmten Personen oder Rollen zu erlauben.

Es gibt zwei Möglichkeiten, wie die Wahl des Passworts erfolgen kann, einerseits durch den Benutzer selbst, andererseits durch das System. Werden die Passwörter von den Nutzern selbst festgelegt, ist es sinnvoll, allgemeine Richtlinien aufzustellen, nach welchen Kriterien sich ein Passwort richten sollte und die Einhaltung dieser Kriterien zu überprüfen. Beispielsweise kann eine bestimmte Mindestlänge für ein Passwort vorgegeben sein. Dann sollte das System das Anlegen von Passwörtern nicht erlauben, die diese Mindestlänge unterschreiten. Erfolgt das Ändern der Passwörter allein durch den Benutzer und muss das Passwort periodisch geändert werden, sollte das System einerseits verhindern, dass das alte Passwort erneut verwendet wird und andererseits eine Historie alter Passwörter anlegen, damit kein Kennwort gewählt wird, das dem alten sehr ähnlich ist.

Die Generierung der Passwörter kann aber auch durch das System selber erfolgen. In diesem Fall ist natürlich keine Historie nötig. Der Passwort-Generator sollte dabei in der Lage sein, ein Passwort mit der nötigen Komplexität und Zufälligkeit zu generieren. Auf keinen Fall

darf der Generator Passwörter nach einem bestimmten Muster anlegen. Falls der Benutzer angeben kann, nach welchen Kriterien die Erstellung eines Passworts erfolgt, also beispielsweise die Länge des Passworts oder aus welchen Zeichen es sich zusammensetzen soll, sollten ebenfalls Vorgaben zu einer guten Passwort-Sicherheit gemacht werden und deren Einhaltung geprüft werden.

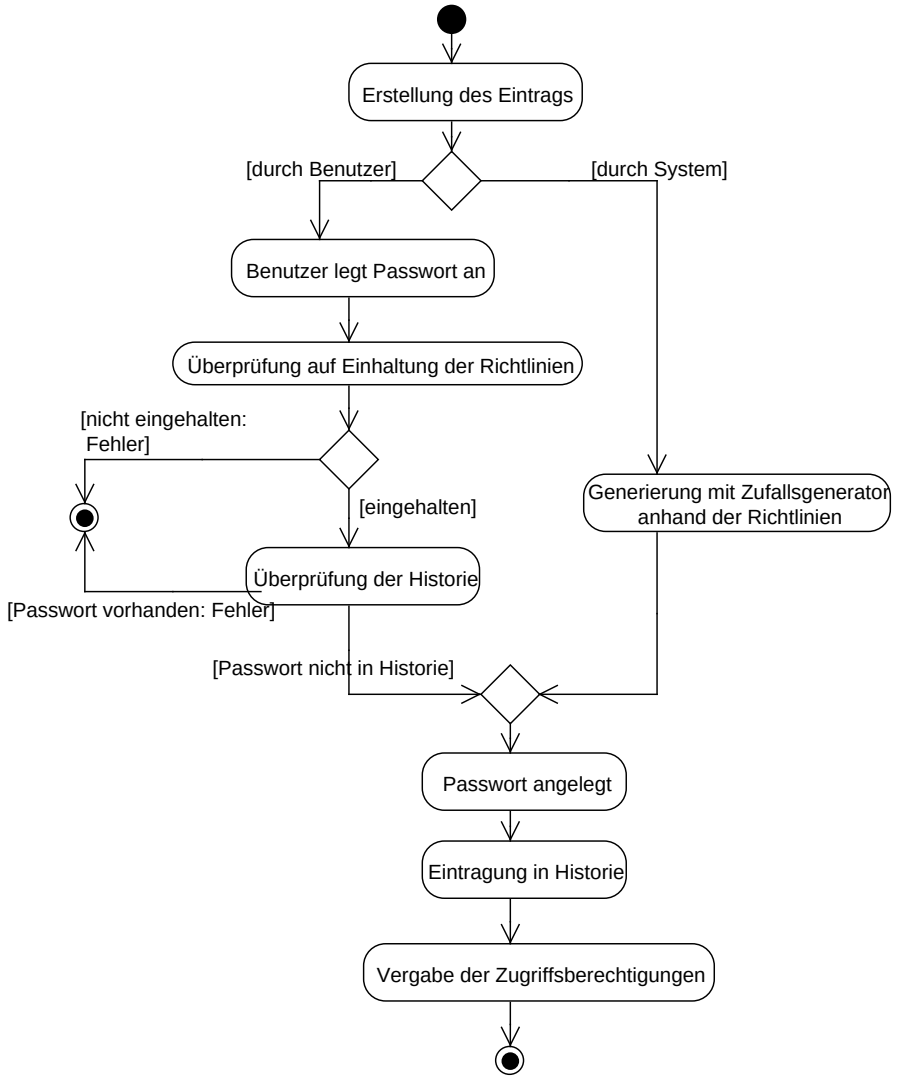

Abbildung 3.2: Aktivitätsdiagramm zum Vorgangs des Anlegens eines Passworts

Von diesen zwei Alternativen zur Erstellung der Passwörter ist die Generierung durch das System zu bevorzugen. Die automatische Erstellung garantiert die Wahl guter Passwörter und bedeutet auch weniger Aufwand für den Eigentümer, der dann nicht mühsam ein eigenes Passwort erstellen muss.

Die Vergabe von Berechtigungen erfolgt über den Eigentümer des Passworts. Er kann einzelnen Benutzern und Gruppen den Zugriff erlauben. Es sollte auch möglich sein, einer Gruppe den Zugriff zu erlauben, aber ein bestimmtes Mitglied auszuschließen. Die Berechtigung zum Ändern des Passworts sollte nicht allein beim Eigentümer liegen, denn es kann der Fall auftreten, dass der Eigentümer nicht verfügbar ist, das Passwort aber geändert werden muss. Außerdem sollte es entweder möglich sein, einen Eintrag zu löschen und neu zu erstellen oder aber die Eigentumsberechtigungen an einem Passwort-Eintrag übertragen zu können, falls der Eigentümer eines Passworts beispielsweise das Unternehmen verlässt.

Das Ändern des Passworts sollte aber auch automatisch über das System funktionieren können. Einerseits sollte es möglich sein, dass eine regelmäßige Passwort-Änderung nach einem gewissen festzulegendem Zeitintervall automatisch durchgeführt wird, andererseits sollte das System in manchen Fällen von sich aus ein Passwort ändern (siehe 3.2.4). Die Änderung des Passworts sollte also sowohl Personen als auch dem System möglich sein.

Außerdem kann bei Anlegen des Passworts auch eingestellt werden, ob der Eintrag offen oder verdeckt ist. Nicht jeder Benutzer sollte jeden Eintrag sehen können. Unter Umständen sollten nur bestimmte Benutzer oder Gruppen den Passwort-Eintrag sehen können. Ob ein Eintrag verdeckt sein sollte oder nicht, hängt von der Identität und Wichtigkeit des dahinter liegenden Systems ab.

Aus der Identität und Wichtigkeit des Systems leitet sich auch der Autorisierungsprozess ab, im Besonderen, welche Personen die Freigabe eines Passworts erlauben können und wie viele Zustimmungen vorliegen müssen, bevor ein Benutzer zeitlich begrenzten Zugriff auf ein Passwort erhält.

Der Eigentümer eines Passworts sollte über verschiedene Aktionen bezüglich seines Passworts informiert werden. Er muss erfahren, wenn ein Passwort geändert wurde, ob auf das Passwort zugegriffen wurde oder wann die nächste automatische Änderung des Passworts fällig ist. Der Eigentümer kann beispielsweise per E-Mail oder SMS informiert werden.

Auch einige Auditinformationen müssen bei Anlegen des Passworts generiert werden. Es muss festgehalten werden, wer wann das Passwort angelegt hat und ähnliche Informationen (siehe 3.2.6).

3.2.4 Autorisierung einer temporärer Zugriffsberechtigung

Der dauerhafte Zugriff auf ein Passwort wird durch den Eigentümer des Passworts gewährt. Aber es kann auch vorkommen, dass ein Administrator vorübergehenden Zugriff auf ein Kennwort erhalten muss, für das er keine Berechtigung hat. Dies kann beispielsweise in folgenden Fällen notwendig sein:

- Ein für ein System zuständiger Administrator ist nicht verfügbar, aber es liegt ein Fehler vor, der behoben werden muss. Ein anderer Administrator benötigt die Berechtigung, auf dieses System zugreifen zu können. Ob die temporäre Übertragung von Zugriffsrechten sinnvoll ist, hängt ab von

  - der Zeitspanne, die benötigt wird, bis eine zuständige Person wieder verfügbar ist.
  - der Identität und der Wichtigkeit des Systems. Bei einem weniger wichtigen System kann mit der Behebung des Fehlers gewartet werden, bis eine zuständige Person sich darum kümmern kann.
  - der Schwere des Fehlers.

  Es muss also eine Priorisierung vorgenommen werden und nach Dringlichkeit der Fehlerbehebung entschieden werden.

- Ein Administrator scheidet aus dem Dienst aus, ohne dass eine geordnete Nachfolge organisiert wurde.
- Unter Umständen müssen bestimmte Teams auf ein System zugreifen können. Das IT-Sicherheitsteam muss bei Sicherheitsvorfällen Zugriff auf diese Accounts erhalten. Ebenso kann es vorkommen, dass zur Bearbeitung von Major Incidents das ITSM-Team Zugriff braucht.

Benötigt eine Person also zeitlich beschränkten Zugriff auf einen Account, muss eine Anfrage auf temporäre Freigabe des Passworts gestellt werden. Über diese Anfrage wird von den Bevollmächtigten für ein Passwort entschieden, Entscheidungsgrundlage für eine Freigabe sollte die Identität des Anfragestellers, die Identität und Wichtigkeit des Systems und die Schwere des Notfalls sein.

Wie viele Bevollmächtigte zur Freigabe eines Passwort notwendig sind, kann unterschiedlich sein und hängt ebenfalls von diesen Faktoren ab. Unter Umständen muss die Zustimmung mehrerer Personen vorliegen, bevor das Passwort freigegeben wird. Besonders wichtige Systeme sollten mehr vorliegende Autorisierungen benötigen, bevor eine Freigabe gewährt wird.

Die Autorisierung von einer einzigen Person abhängig zu machen, ist wenig sinnvoll, denn es kann auch vorkommen, dass eben diese Person ebenfalls nicht verfügbar ist. Es sollten also mehrere Berechtigte in der Lage sein, Freigabe für ein Passworts zu erteilen. Ebenso ist es grundsätzlich empfehlenswert, einen Antrag auf Offenlegung an mehr Personen weiterzuleiten, als zur Autorisierung nötig sind. Das beschleunigt das Verfahren und verhindert auch, dass die Abwesenheit einer Person dazu führt, dass keine Freigabe erfolgen kann. Ein besonderer Fall ergibt sich, wenn die Autorisierung einer bestimmten Person benötigt wird, diese aber nicht verfügbar ist. In diesem Fall muss an den vorgesetzten Manager eskaliert werden.

Damit eine Anfrage nicht auf Dauer im System bleibt, sollte sie nach einem gewissen Zeitintervall wieder gelöscht werden. Außerdem sollte jede Anfrage geloggt werden. Wichtige festzuhaltende Informationen sind die Identität des Anfragestellers, wer der Autorisierung zugestimmt oder sie abgelehnt hat und natürlich, ob Zugriff gewährt wurde oder nicht (siehe 3.2.6).

Es ist sinnvoll, den Zugriff auf die Accounts außerhalb der normalen Berechtigungen eines Administrators zeitlich zu beschränken und die Berechtigung anschließend wieder zu entzie-

hen. Deswegen ist es wichtig, darüber zu entscheiden, für welchen Zeitraum Zugriff gewährt wird. Die Information, wie lange Zugriff benötigt wird, sollte in der Anfrage enthalten sein.

Wurde Zugriff gewährt und ist das Passwort für den Benutzer sichtbar, sollte es nach einem gewissen Zeitraum wieder ausgeblendet werden. Man kann annehmen, dass es nach einem Zeitintervall gesehen und genutzt wurde. Wie das Passwort freigegeben werden sollte, ist in 3.2.5 näher beschrieben.

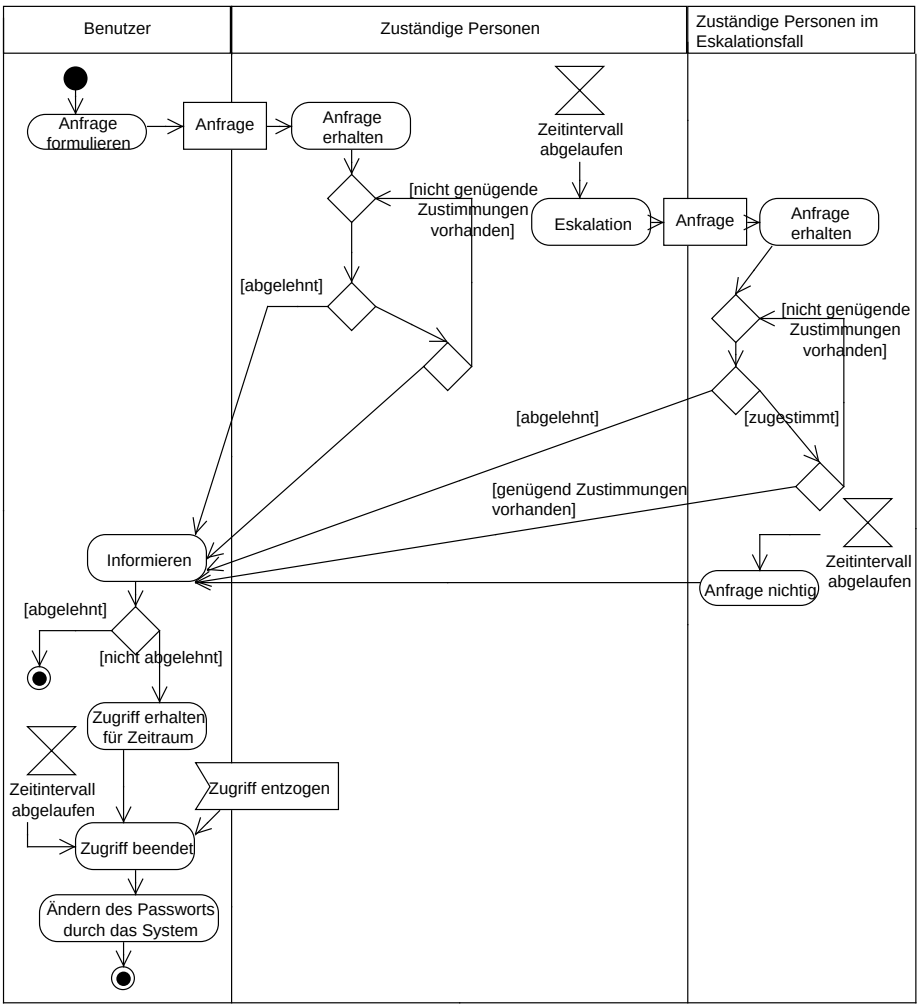

Abbildung 3.3: Aktivitätsdiagramm zur Darstellung des Vorgangs der Autorisierung zur temporären Freigabe

Nach Beenden der Sitzung des Administrators oder bei Auslaufen des Zeitintervalls sollte das benutzte Passwort geändert werden (außer natürlich wenn dem Administrator das Passwort nicht angezeigt wurde). Das geschieht am besten direkt durch das System, das in diesem

Fall automatisch ein neues, zufälliges Passwort für den privilegierten Account generiert. Die Möglichkeit, das Passwort durch den Eigentümer ändern zu lassen, ist nicht vorzuziehen. Es kann sein, dass ein Eigentümer eine Zeitlang nicht verfügbar ist, zudem bedeutet das manuelle Ändern Aufwand und könnte leicht vergessen werden.

Das Ändern des Passworts durch das System kann natürlich nur erfolgen, falls eine automatische Änderung des Passworts am Zielsystem möglich ist. Es muss eine Verbindung zum Zielsystem bestehen und das neue Passwort muss übermittelt werden können. Das System könnte auch periodisch alle Passwörter ändern, einige Privileged-Password-Management-Lösungen bieten beispielsweise das Ändern aller Passwörter alle 24 Stunden oder in ähnlichen Zeiträumen an.

Ein möglicher Ablauf dieses Verfahrens könnte folgendermaßen aussehen (siehe auch Abbildung 3.3):

- Der Benutzer stellt eine Anfrage mit den dafür benötigten Informationen: Wer stellt Anfrage, wer soll Zugriff erhalten, um welches System handelt es sich, wann und wie lange wird Zugriff benötigt.
- Für diesen Account existiert eine Liste an Personen, die autorisieren können. Die Anfrage wird per E-Mail oder SMS an diese Personen weitergeleitet, dabei an mehr Personen, als zur Autorisierung nötig sind.
- Melden sich diese Personen nicht, wird die Anfrage eskaliert.
- Wird die Anfrage in einem Zeitintervall nicht bearbeitet, wird sie nichtig und der Anfragesteller informiert.
- Die Bevollmächtigten können sich am System anmelden, um die Anfrage anzusehen und ihre Zustimmung zu geben oder abzulehnen.
- Lehnt die zuständige Person die Anfrage ab, wird die Anfrage nichtig und der Anfragesteller informiert.
- Liegt die Zustimmung von genügend Personen vor, wird der Benutzer per E-Mail benachrichtigt und kann das Passwort einsehen.
- Der Benutzer hat Zugriff für einen bestimmten Zeitraum.
- Ist die Anzahl gleichzeitig zugelassener Nutzer beschränkt, wird eine weitere Person, die eine Anfrage stellt, nicht zugelassen.
- Bei Beenden der Sitzung oder Auslaufen des Zeitintervalls oder Entzug der Zugriffsberechtigung wird das Passwort zurückgesetzt.
- Alle Vorgänge werden geloggt.

3.2.5 Freigabe der Passwörter

Wenn ein Benutzer entweder dauerhaften oder vorübergehenden Zugriff auf ein Passwort erhalten hat, kann er das Kennwort verwenden. Die einfache Möglichkeit ist, ihm das Passwort anzuzeigen, sodass er es zur Anmeldung eingeben kann. Aber es ist auch möglich, dem Administrator das Passwort nicht offenzulegen, sondern ihn automatisch einzuloggen, indem eine Terminal-Sitzung gestartet wird (beispielsweise mit RDP, SSH, vSphere...). Der Vorteil dieser Methode ist, dass das Passwort geschützt bleibt. Ein Kennwort, das nicht bekannt ist, kann auch nicht mit Kollegen geteilt werden. Auch dem Administrator bleibt der Aufwand erspart, Loginname und Passwort einzugeben.

Eine weitere Möglichkeit, um das Anzeigen des Passworts zu vermeiden, ist das Kopieren des Passworts in einen Puffer, sodass der Administrator es einfach zum Login nutzen kann. Auch so bleibt das Passwort geschützt.

Außerdem muss festgelegt werden, wie viele Administratoren gleichzeitig Zugriff auf einen privilegierten Account haben. Wenn mehrere Personen zur gleichen Zeit in einem System eingeloggt sind, ist nicht mehr ersichtlich, wer für Änderungen im System verantwortlich ist. Auch für die Administratoren ist es besser, wenn nicht parallel Änderungen von anderen Personen durchgeführt werden. Zur Beschränkung könnte man das Passwort einfach sperren, sodass keine andere Person Zugriff bekommt.

Bei der Beschränkung des Zugriffs auf eine Person ergibt sich allerdings ein anderes Problem. Es kann sein, dass die einzelne Person mit Zugriffsberechtigung für einen Zeitraum nicht verfügbar ist, in diesem Intervall aber Zugriff benötigt wird, weil beispielsweise ein Fehler im System vorliegt. Dies macht es nötig, den Zugriff einer Person auf den Account auf einen gewissen Zeitraum zu begrenzen, damit Passwort und Login nach Ablauf dieses Zeitrahmens wieder anderen Administratoren zur Verfügung steht.

Besonders problematisch wird die Beschränkung des Zugriffs, wenn der Administrator mit Berechtigung nicht in der Lage ist, auf den Account zuzugreifen, beispielsweise bei Verbindungsproblemen. In diesem Fall kann sich auch keine weitere Person einloggen. Deswegen ist eine Beschränkung des Zugriffs generell nicht zu empfehlen.

3.2.6 Audit

In Audit-Logs werden Informationen gesammelt, die Aufschluss darüber geben, welcher Benutzer wann welche Aktivität durchgeführt hat. Diese Informationen sollten nicht frei verfügbar, sondern nur für autorisierte Benutzer einsehbar sein. Außerdem müssen diese Daten vor Manipulation geschützt sein, sodass niemand die Daten ändern kann.

Die Aktionen aller Benutzer sollten geloggt werden, ebenso alle Zugriffe auf die geschützten Daten. Informationen, die in den Audit-Logs enthalten sein sollten, umfassen:

- Jegliche Informationen über Zugriffsrechte bei einem Eintrag, insbesondere welche Person welche Berechtigungen erhalten hat oder wem sie entzogen wurden.

- Ebenso alle Informationen, die sich auf ein Passwort beziehen: ein Passwort wurde hinzugefügt, geändert, gelöscht oder eingesehen.
- Wichtig ist das Loggen vor allem im Falle einer Anfrage auf temporäre Autorisierung. Hier sollte geloggt werden, wer wann für wen die Anfrage gestellt hat, an wen die Anfrage gegangen ist, ob Zustimmung erteilt wurde oder nicht und natürlich, ob Zugriff gewährt wurde oder nicht.
- Weitere Informationen beziehen sich auf die Verwaltung der Datenbank. Darunter fallen beispielsweise Informationen wie der Erfolg oder das Fehlschlagen eines Logins; das Anlegen, Ändern, Löschen, Aktivieren oder Deaktivieren eines Benutzeraccounts; das Hinzufügen oder Entfernen eines Benutzer zu einer Gruppe; das Anlegen, Ändern oder Löschen einer Gruppe.

Vor allem ist wichtig, dass die Aktionen eines Benutzers ihm zu jedem Zeitpunkt klar zuzuordnen sind. Es sollte nachvollziehbar sein, wer zu welchem Zeitpunkt Zugriff auf ein Passwort hatte.

Es wäre von Vorteil, wenn zusätzlich zu den Audit-Logs Tools existieren, die es ermöglichen, die Daten zu analysieren, beispielsweise um die Aktionen eines bestimmten Benutzers anzeigen zu können. Ebenfalls sollte das System automatisch in der Lage sein, bei manchen Ereignissen Benachrichtigungen an zuständige Personen zu verschicken.

Das Einsehen der Audits könnte auch nach dem Vier-Augen-Prinzip erfolgen, sodass mindestens zwei Personen zusammen Einblick in die Informationen haben. Da es sich bei Auditinformationen unter Umständen um personenbezogene Daten handelt, könnte ein Mitglied des Personalrats hinzugezogen werden.

Es gibt viele Möglichkeiten, die Audits zu gestalten. Allgemein bietet es sich an, das Audit-Log entweder auf die Aktionen der einzelnen Benutzer oder die Aktivitäten im Zusammenhang mit einem bestimmten Passwort-Eintrag zu beziehen. Ein nutzerbezogenes Audit-Log kann für jeden einzelnen Benutzer die durchgeführten Aktionen anzeigen, also beispielsweise welche Passwörter von ihm angelegt, eingesehen oder geändert wurden. Ein Log, das die Einträge im Mittelpunkt hat, würde ausgehend von jedem Datensatz die einzelnen mit diesem Eintrag erfolgten Aktivitäten darstellen, also beispielsweise wann es angelegt wurde oder welcher Benutzer wann Zugriff darauf hatte.

Natürlich können auch mehrere Audit-Logs angelegt werden. Lösungen im Bereich von Privileged Password Management bieten zusätzlich oft auch an, die Sitzungen der Administrators aufzunehmen, um diese zu einem späteren Zeitpunkt abspielen zu können. Das macht es möglich, jede einzelne Aktion des Administrators zurückzuverfolgen.

Eine weitere Fragestellung ist, wie lange die Logs aufbewahrt werden sollten. Diese Frage hängt im Grunde davon ab, wie lange man die Aktivitäten einzelner Benutzer zurückverfolgen will und kann von Unternehmen zu Unternehmen unterschiedlich beantwortet werden. Auf jeden Fall sollten Logs für den temporären Zugriff und Informationen über die Aktionen eines Benutzers über einen längeren Zeitraum aufbewahrt werden. Verwaltungsinformationen zu den Logins von Benutzern am System und zum Anlegen und Löschen von Gruppen

und Benutzern sind weniger wichtig und müssen nicht so lang aufbewahrt bleiben. Generell kann sich die Aufbewahrung von Audit-Logs an internen Audit-Zyklen orientieren. Für externe Audits beträgt die von vielen Richtlinien geforderte Aufbewahrungszeit ein Jahr. [RSA07]

Zur Zeit gibt es noch keine Gesetze, die spezifische Zeiträume für die Aufbewahrung von Audit-Logs vorgeben. Oft wird jedoch ein entsprechender Schutz der Auditinformationen gefordert. Wie sich die gesetzlichen Vorgaben in diese Richtung entwickeln, ist zur Zeit nicht abzusehen. Es ist aber vorstellbar, dass zur Aufbewahrung von Audit-Logs schon allein aus datenschutzrechtlichen Gründen in Zukunft rechtliche Regelungen erfolgen werden.

3.3 Resultierender Kriterienkatalog

Aus den dargelegten Anwendungsfällen und Anforderungen lässt sich ein Katalog entwickeln, in dem die Kriterien an ein System zur Passwortverwaltung zusammengefasst sind. Ein Teil der aufgeführten Kriterien sind dabei Anforderungen, die erfüllt werden müssen, um ein gut funktionierendes System zum Passwort-Management zu haben. Andere Kriterien wiederum sind zwar wünschenswert und dienen der vereinfachten Verwaltung, ein System ist aber auch ohne ihre Erfüllung denkbar.

Aufbau des Systems

- Zentrale Speicherung der Passwörter: Um jederzeit Zugriff auf alle Passwörter gewährleisten zu können, sollten die Kennwörter zentral gespeichert werden.
  - Ausschlusskriterium: Es existiert keine zentrale Struktur, in der die Passwörter gespeichert werden.
- Automatische Änderung der Passwörter: Das System sollte in der Lage sein, automatisch die Kennwörter zu ändern und die geänderten Kennwörter sicher an die Zielsysteme weiterzuleiten.
  - Ausschlusskriterium: Die Passwort-Änderung ist nicht automatisch möglich.
- Erfüllung verschiedener Sicherheitskriterien: Die Kommunikation zwischen Benutzer, System und den Komponenten muss verschlüsselt ablaufen und die Passwörter müssen verschlüsselt abgespeichert werden.
  - Ausschlusskriterium: Diese Sicherheitsanforderungen werden nicht erfüllt.
- Mehrbenutzerbetrieb: Es gibt die Möglichkeit, mehrere Benutzer anzulegen und den Zugriff der einzelnen Benutzer zu beschränken.
  - Ausschlusskriterium: Es gibt nur ein zentrales Login, das sich mehrere Benutzer teilen müssen.
  - Ausschlusskriterium: Es ist nicht möglich, den Zugriff der einzelnen Benutzer einzuschränken.

Wünschenswert wäre weiterhin:

- Sicherung der Daten auch bei Ausfall des Servers: Das System sollte hoch verfügbar sein, die Ausfallzeit also möglichst gering. Außerdem sollte es möglichst fehlerfrei arbeiten. Im Idealfall gibt es die Möglichkeit, die Daten so zu sichern, dass auch bei einem Ausfall des Servers weiterhin auf sie zugegriffen werden kann.
- Log mit alten Passwörtern vorhanden: Es sollte ein Log geben, in dem alte Passwörter gespeichert werden, um diese Passwörter im Falle der Wiederherstellung eines System mit veralteten Archivdaten nutzen und auf das System zugreifen zu können. Außerdem kann damit verhindert werden, dass alte Passwörter erneut genutzt werden.

Berechtigungsverwaltung und Zugriffskontrolle

- Zugriffskontrolle: Der Zugriff auf die Datenbank ist nur eingeschränkt möglich. Benutzer werden vom System geeignet authentifiziert und autorisiert, bevor sie Zugriff erhalten.
  - Ausschlusskriterium: Die Datenbank mit den Passwörtern ist offen einsehbar und es existieren keine Mechanismen, um den Zugriff auf die Datenbank nur bestimmten Personen zu ermöglichen.
- Anlegen von Gruppen oder Rollen ist möglich: Für eine vereinfachte Verwaltung von Berechtigungen ist es von Vorteil, wenn Gruppen angelegt werden können. Eine Alternative ist die Existenz verschiedener Rollen, die den Benutzern zugeteilt werden können.

Wünschenswert wäre weiterhin:

- Übernahme der Benutzer und Gruppen aus Verzeichnisdiensten: Es ist möglich, Benutzer und Gruppen von Verzeichnisdiensten wie Active Directory oder LDAP zu übernehmen.

Erstellen der Passwörter

- Vorgabe von Passwort-Richtlinien bei der Erstellung von Passwörtern durch den Benutzer: Werden Kennwörter durch den Benutzer angelegt, muss es möglich sein, Richtlinien zur Erstellung vorzugeben und die Einhaltung dieser Richtlinien kontrollieren zu können. Außerdem muss ein Log vorliegen, das alte Passwörter enthält, damit nicht schon verwendete Passwörter erneut gewählt werden können.
  - Ausschlusskriterium: Die Passwort-Vergabe ist allein den Benutzern möglich und die gewählten Passwörter werden nicht kontrolliert.
- Vorgabe von Passwort-Richtlinien und Generierung geeigneter Passwörter bei Nutzung eines Generators: Bei der Erstellung von Passwörtern mit einem Passwort-Generator muss dieser Generator in der Lage sein, geeignete Passwörter zu erstellen. Im Besonderen sollten die Richtlinien einer guten Passwort-Sicherheit eingehalten werden. Kann der Benutzer selbst Vorgaben an das vom System zu erstellende Passwort wie beispielsweise die Länge des Kennworts machen, sollte es auch hier die Möglichkeit geben, Kriterien an das Passwort vorzugeben und auf deren Einhaltung zu achten.

  - Ausschlusskriterium: Der Passwort-Generator ist unzureichend.

- Kontrolle des Passwort-Zugriffs: Der Zugriff auf die Passwörter darf nur autorisierten Personen möglich sein.
  - Ausschlusskriterium: Passwörter sind für alle Benutzer offen einsehbar.
- Vergabe von Zugriffsberechtigungen auf Passwörtern: Es muss möglich sein, anderen Benutzern verschiedene Berechtigungen zu übertragen. So sollten andere Benutzer Lese- und Schreibberechtigung erhalten können.
  - Ausschlusskriterium: Es gibt keine Möglichkeit, anderen Benutzern entsprechende Zugriffsberechtigungen zu übertragen.
- Verbergen von Einträgen: Die Möglichkeit existiert, Passwort-Einträge zu verbergen, sodass nur ausgewählte Benutzer sie sehen können.
  - Ausschlusskriterium: Alle Einträge können eingesehen werden.

Wünschenswert wäre weiterhin:

- Vergabe von Passwörtern durch das System: Allgemein ist es zu bevorzugen, wenn die Passwörter vom System erstellt werden.

Autorisierung einer temporären Zugriffsberechtigung

- Notfall-Zugriff ist möglich: In Notfällen können Administratoren Zugriffsberechtigungen auf Passwörter und Accounts erhalten, die sie normalerweise nicht besitzen.
  - Ausschlusskriterium: Es ist Benutzern nicht möglich, in Notfällen einen Passwort-Eintrag einsehen zu können.
- Über eine Freigabe kann von mehreren Personen entschieden werden: Die Bevollmächtigung des Zugriffs hängt nicht von einer einzelnen Person ab. Es ist möglich, dass mehrere zuständige Personen über die Freigabe eines Passworts entscheiden.
  - Ausschlusskriterium: Die Bevollmächtigung wird nur von einer einzigen Person erteilt.
- Zeitliche Beschränkung der Freigabe: Ein Passwort wird nur für einen gewissen Zeitraum freigegeben und die Zugriffsberechtigung wird anschließend wieder entzogen.
  - Ausschlusskriterium: Der Zugriff im Notfall kann zeitlich nicht beschränkt werden und es findet kein automatischer Entzug der Zugriffsrechte statt.
- Automatisches Ändern des Passworts: Nach Beenden des temporären Zugriffs wird vom System ein neues Passwort erstellt und an das Zielsystem übertragen, sofern Verbindung zu dem Zielsystem besteht.
  - Ausschlusskriterium: Das Passwort gilt nach Notfallzugriff weiterhin.
- Loggen des Notfall-Zugriffs: Jede temporäre Freigabe wird automatisch vom System geloggt.

  - Ausschlusskriterium: Es findet kein Logging statt und im Audit sind keine Informationen über den Notfallzugriff enthalten.

Freigabe der Passwörter

- Zugriff auf einen Account ist mehreren Personen möglich.
  - Ausschlusskriterium: die Berechtigung für Zugriff auf den Account liegt bei nur einer Person.

Wünschenswert wäre weiterhin:

- Einloggen ohne Anzeigen des Passworts ist möglich: Der Nutzer kann sich auf einem Zielsystem einloggen, ohne dass ihm das Passwort angezeigt wird.

Audit

- Existenz eines Audit-Logs: Es wird ein Audit-Log erstellt, das die wichtigsten Informationen darüber enthält, wer wann welche Aktion durchgeführt hat.
  - Ausschlusskriterium: Es gibt kein zentrales Log, in dem diese Informationen gesammelt werden.
- Das Log kann nicht von Administratoren geändert werden: Das Log sollte möglichst resistent gegenüber Manipulationen sein.
  - Ausschlusskriterium: Auditinformationen können manipuliert werden, die Manipulationen können nicht festgestellt werden.
- Der Zugriff auf die Logs ist nur eingeschränkt möglich.
  - Ausschlusskriterium: Die Informationen der Logs sind offen einsehbar.

Wünschenswert wäre weiterhin:

- Automatisches Versenden von Benachrichtigungen: Das System ist in der Lage, Benachrichtigungen zu verschicken, um zuständige Personen über wichtige Ereignisse zu informieren.

4 Entwurf eines kryptografischen Modells zur Passwortverwaltung

Da sich Sicherheit in IT-Systemen derzeit nicht mit Hilfe von standardisierten Metriken messen lässt, werden Schutzziele aufgestellt, durch deren Einhaltung die Sicherheit des Systems gewährleistet werden soll. Diese Schutzziele sind

Vertraulichkeit (confidentiality). Dieses Schutzziel besagt, dass geschützte Daten nur von Berechtigten eingesehen werden dürfen. Das wird oft durch den Einsatz von Verschlüsselung erreicht.

Integrität (integrity). Das bedeutet, dass geschützte Daten nicht von unautorisierten Personen geändert werden können. Dazu werden meist kryptographische Hashverfahren verwendet.

Verfügbarkeit (availability). Services und Daten können von dazu autorisierten Personen ohne Störung genutzt werden.

Bei einem Modell der Passwortverwaltung, wie es in Kapitel 3 aufgestellt wurde, ist vor allem die Vertraulichkeit der Daten wichtig, nur autorisierte Personen dürfen Zugriff auf die Passwörter haben. Um ein kryptografisches Modell zu entwerfen, muss das Management der Passwörter also entsprechend mit kryptografischen Mechanismen abgesichert werden. In einem ersten Schritt muss festgestellt werden, welche Daten überhaupt von einem solchen System gespeichert werden, welche Daten besonders schützenswert sind und wie der Schutz dieser Daten erfolgen kann (siehe 4.2). Dazu werden sowohl die Daten als auch die Passwort-Einträge je nach benötigtem Schutz in Vertraulichkeitsstufen eingeordnet.

In einem zweiten Schritt werden kryptografische Maßnahmen besprochen, die der Absicherung der Passwortverwaltung dienen. Im Besonderen muss ein System in der Lage sein, den Zugriff auf schützenswerte Einträge und Passwörter zu beschränken und die Passwörter sicher zu übertragen. Falls die Zugriffskontrolle von einer Anwendung übernommen wird, muss sich der Benutzer bei der Anwendung authentisieren (siehe 4.3.1). Allgemein müssen die Daten sicher gespeichert werden (siehe 4.3.2) und die Kommunikation zwischen den Beteiligten im System muss geschützt ablaufen (siehe 4.3.3). Insgesamt ist also auch eine Verwaltung der verwendeten Schlüssel erforderlich (siehe 4.3.4).

In einem dritten Schritt wird eine theoretische Lösung für das Passwort-Management aufgestellt, die auf einen vertrauenswürdigen Dritten zur Zugriffskontrolle verzichtet. Das wird mit der Verwendung des kryptografischen Primitivs Attribute-Based Encryption möglich gemacht (siehe 4.4). Dieses Modell wird anschließend mithilfe des Kriterienkatalogs bewertet (siehe 4.4.2).

4.1 Kryptografische Grundbegriffe

Zur Einhaltung der Schutzziele gibt es mehrere kryptografische Mechanismen, über die die folgenden Kapitel einen kurzen Überblick geben.

4.1.1 Symmetrische und asymmetrische Verschlüsselungsverfahren

Kryptografische Verfahren lassen sich in symmetrische und asymmetrische Verfahren unterteilen. Bei symmetrischen Verfahren wird ein Schlüssel für die Absicherung der Kommunikation genutzt, der sowohl für Ver- als auch Entschlüsselung verwendet wird. Alle Beteiligten müssen also zur gemeinsamen Kommunikation den Schlüssel kennen. Das macht den Schlüsselaustausch problematisch, da allen Partnern der gemeinsam genutzte Schlüssel sicher übermittelt werden muss, ohne dass Dritte den Schlüssel erfahren.

Symmetrische Verfahren lassen sich in Strom- und Blockchiffren unterteilen. Bei Stromchiffren wird ein Bitstrom generiert, mit dem die Nachricht Bit für Bit verschlüsselt wird. Bei Blockchiffren wird die Nachricht in Blöcke unterteilt und die einzelnen Blöcke werden mit dem Schlüssel verschlüsselt.

Bei asymmetrischer Verschlüsselung wird ein Schlüsselpaar aus einem öffentlichen und einem geheimen Schlüssel verwendet. Dabei darf der geheime Schlüssel nicht aus dem öffentlichen abzuleiten sein und eine mit dem öffentlichen Schlüssel verschlüsselte Nachricht darf nur allein mit Kenntnis des privaten Schlüssels entschlüsselbar sein. Soll eine Nachricht an den Besitzer eines bestimmten öffentlichen Schlüssels gesendet werden, wird diese mit dem bekannten öffentlichen Schlüssel verschlüsselt. Der Besitzer des öffentlichen Schlüssels kann diese dann mit seinem geheimen Schlüssel wieder entschlüsseln. Asymmetrische Verfahren haben den Vorteil, dass sie nicht auf sichere Kanäle zum Schlüsselaustausch angewiesen sind wie symmetrische Verfahren, sie sind aber auch langsamer als symmetrische Verfahren. Bei asymmetrischen Verfahren ist es allerdings wichtig, dass Sender und Empfänger sich entsprechend authentisieren.

Hybride Verfahren kombinieren symmetrische und asymmetrische Verfahren miteinander. Zum Schlüsselaustausch werden asymmetrische Verfahren angewandt, für die anschließende Kommunikation werden symmetrische Schlüssel verwendet.

4.1.2 Hashverfahren und Integritätssicherung

Hashverfahren werden verwendet, um Datenintegrität sicherzustellen. Eine Hashfunktion erzeugt aus einer beliebig langen Eingabe eine Ausgabe fester Länge, den Hashwert. Gute Hashfunktionen stellen sicher, dass es nahezu unmöglich ist, zu einem vorgegebenen Hashwert eine Nachricht zu erzeugen, die dem Hashwert entspricht. Zudem sollten die Funktionen möglichst kollisionsresistent sein. Das bedeutet, dass zu zwei verschiedenen Nachrichten nicht der gleiche Hashwert berechnet wird.

Aus der Kollisionsresistenz von Hashwerten folgt, dass bei gleichen Hashwerten mit sehr hoher Wahrscheinlichkeit die gleiche Eingabe vorliegt. Dies wird bei bestimmten Angriffen auf Passwörter ausgenutzt, indem eine Liste mit bereits berechneten Hashes mit den gehashten Passwörtern verglichen wird. Deswegen werden Passwörter mit zufälligen Zeichenfolgen, sogenannten Salts, versehen und anschließend gehasht, um die Entropie zu erhöhen.

Die Integritätssicherung erfolgt über die Berechnung einer Prüfsumme. Dabei wird zu den Daten ein bestimmter Wert berechnet und an die Nachricht angehängt. Der Sender berechnet eine Prüfsumme und schickt die Daten zusammen mit dieser Summe. Der Empfänger berechnet zu der erhaltenden Nachricht ebenfalls die Prüfsumme und vergleicht sie mit der empfangenen. Bei symmetrischen Verfahren wird dazu der gleiche Schlüssel benutzt, bei asymmetrischen Verfahren wird der private Schlüssel zur Berechnung der Prüfsumme verwendet und mithilfe des öffentlichen Schlüssels überprüft.

Bei Signatur-Verfahren werden die Daten zunächst gehasht, dann mithilfe des privaten Schlüssels signiert. Der Empfänger prüft die Signatur mit dem entsprechenden öffentlichen Schlüssel. Bei Message Authentication Codes (MACs) wird stattdessen ein symmetrisches Verfahren verwendet, bei dem Sender und Empfänger sich zuerst auf einen geheimen Schlüssel einigen müssen. MACs setzen entweder Blockchiffren oder Hash-Funktionen ein.

4.1.3 Secret Sharing

Secret Sharing behandelt Verfahren, bei denen ein Geheimnis unter mehreren Teilnehmern aufgeteilt wird. Zur Rekonstruktion dieses Geheimnisses müssen die Teilnehmer miteinander kooperieren, kein Teilnehmer kann alleine das Geheimnis bestimmen. Dabei werden (t, n)-Schwellenschemata verwendet: das Geheimnis s wird auf n Personen aufgeteilt, die Kooperation von t oder mehr Personen wird zur Rekonstruktion benötigt. Es muss also gelten, dass $t \leq n$ ist.

Ein häufig genutztes Secret-Sharing-Verfahren ist das Schwellenschema von Shamir. Bei diesem Schema wählt der Dealer zur Aufteilung des Geheimnisses s ein Polynom von Grad $t-1$

$$f(x) = s + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_{t-1} \cdot x^{t-1}$$

Die Koeffizienten a_i werden zufällig gewählt. Das Geheimnis entspricht dabei $s = f(0)$. Jeder der n Teilnehmer erhält ein $(x_i, f(x_i))$, $i = 1, ..., n$. $f(x_i) = s_i$ ist dabei als das Teilgeheimnis zu verstehen, das selbstverständlich geheim gehalten werden muss.

Mindestens t Teilnehmer müssen kooperieren, um das Geheimnis rekonstruieren zu können. Die Rekonstruktion kann mithilfe der Lagrange-Interpolation erfolgen. Dabei wird

$$f(x) = \sum_{i=1}^{t} \frac{(x - x_1)...(x - x_{i-1})(x - x_{i+1})...(x - x_t)}{(x_i - x_1)...(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_t)} s_i$$

berechnet. [ABS10]

Zum Beispiel soll das Geheimnis a_0 geschützt werden. Es soll auf $n = 6$ Personen aufgeteilt werden, $t = 4$ Teilgeheimnisse sollen zur Rekonstruktion des Geheimnisses genügen.

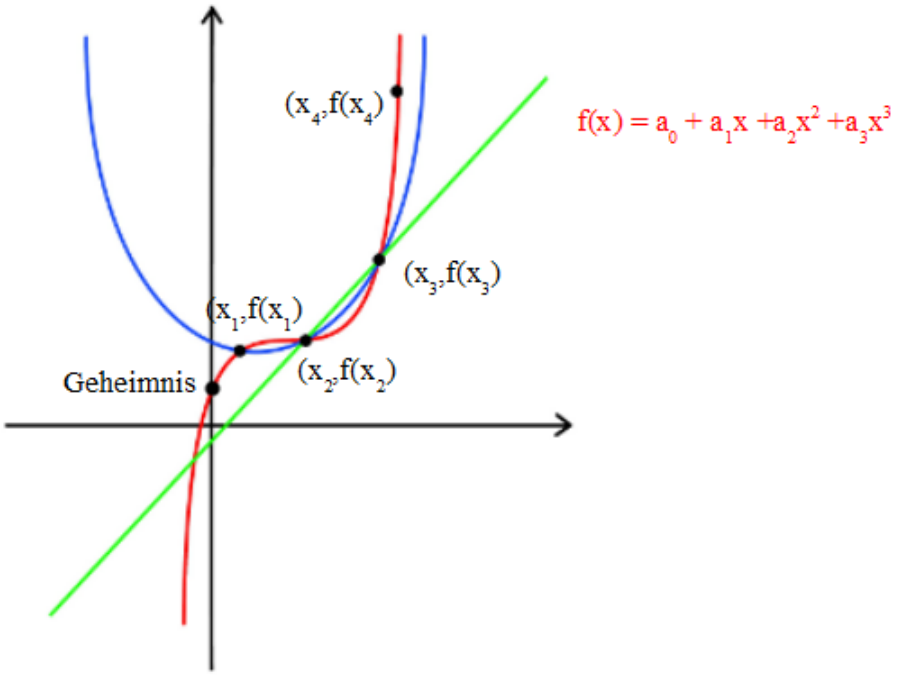

Abbildung 4.1: Ein Beispielgraph für Shamirs Secret Sharing. Die grüne Linie kennzeichnet die Gerade, die durch die Kenntnis zweier Punkte gebildet werden kann, die blaue Parabel ergibt sich durch die Kenntnis dreier Punkte.

Folgendes Polynom wird gebildet:

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$$

mit zufällig gewählten Koeffizienten a_1, a_2, a_3. Im nächsten Schritt werden die Teilgeheimnisse $(x_1, f(x_1))$ bis $(x_6, f(x_6))$ berechnet und auf die Teilnehmer aufgeteilt. Die Teilgeheimnisse entsprechen einem Punkt auf dem Graphen.

Nur durch die Kenntnis von vier Teilgeheimnissen kann nun das Geheimnis mithilfe der Lagrange-Interpolation rekonstruiert werden. Ein Beispiel dazu wird in Abbildung 4.1 gezeigt. Ziel der Rekonstruktion ist es, das Polynom bilden zu können. Sind zwei Teilgeheimnisse bekannt, kann eine Gerade gebildet werden. Sind drei Teilgeheimnisse bekannt, kann eine Parabel gebildet werden. Aber erst durch die Freigabe von vier Teilgeheimnissen kann das Polynom in diesem Beispiel dargestellt werden.

Secret-Sharing-Verfahren können im Zusammenhang mit der Verwaltung von Passwörtern dazu genutzt werden, ein Passwort in Teilgeheimnisse aufzuteilen, sodass mehrere Personen kooperieren müssen, um es zu rekonstruieren. Auch das Passwort kann erst eingesehen werden, wenn genügend Personen ihre Teilgeheimnisse freigeben.

Eine weitere Möglichkeit ist die Aufteilung eines Schlüssels. Dabei ist der Schlüssel das Geheimnis, das in Teilgeheimnisse aufgeteilt wird. Auch der Schlüssel kann nur dann zur Entschlüsselung genutzt werden, wenn genügend Personen kooperieren.

4.1.4 Attribute-Based Encryption

Mit der Frage, wie Zugriffskontrolle durch Verschlüsselung und ohne einen vertrauenswürdigen Dritten geregelt werden kann, beschäftigt sich das kryptografische Primitiv Attribute-Based Encryption (ABE). Das ursprüngliche Konzept wurde 2005 von Sahai und Waters unter dem Titel Fuzzy Identity Based Encryption vorgeschlagen [SW05] und in die zwei Formen Key-Policy Attribute-Based Encryption (KP-ABE) [GPSW06] und Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [BSW07] weiterentwickelt. Das Besondere dabei ist, dass sowohl Schlüssel als auch Geheimtext mit Attributen versehen werden. Ein Schlüssel kann einen Text nur entschlüsseln, falls bestimmte Attribute von Schlüssel und Text übereinstimmen. Welche Nutzer also einen Text entschlüsseln können, hängt von den festgelegten Attributen ab.

Bei KP-ABE wird dem geheimen Schlüssel eines Nutzers eine Zugriffsstruktur zugeordnet und der Geheimtext mit Attributen versehen. Die Zugriffsstruktur ist dabei ein monotoner Zugriffsbaum, bei dem die Attribute die Blätter bilden und die Knoten Threshold Gates sind. Der Geheimtext kann entschlüsselt werden, falls die Zugriffsstruktur des Schlüssels die Attribute des Geheimtextes erfüllen. CP-ABE ist genau anders herum aufgebaut: dem Geheimtext ist eine Zugriffsstruktur zugeordnet und der geheime Schlüssel des Nutzers enthält die Attribute.

Beispielsweise lässt sich CP-ABE gut in Role-Based-Access-Control-Szenarien einsetzen. Ein Nutzer erhält Attribute, die seiner Rolle entsprechen, für jede Rolle gibt es ein eigenes Attribut. Die Daten werden dann mit einem Set an Attributen (einer sogenannten Policy) verschlüsselt. Zum Beispiel wird ein Datensatz mit der Policy RolleA OR (RolleB AND RolleC) verschlüsselt. Auf den Datensatz könnten Personen zugreifen, die entweder die Rolle A haben oder sowohl Rolle B als auch Rolle C ausfüllen.

Der große Vorteil von ABE ist, dass sich nicht mehrere Benutzer zusammenschließen können, um gemeinsam auf verschlüsselte Daten zuzugreifen, für die sie allein nicht die Berechtigung haben. Wie dies umgesetzt werden kann, ist in [GPSW06] und [BSW07] näher beschrieben.

4.2 Gespeicherte Daten eines Passwort-Management-Tools

In einem ersten Schritt wird untersucht, welche Daten von einem Passwort-Management-Tool gespeichert werden. Generell muss sich der Benutzer bei dem System authentifizieren können. Dazu braucht er im Normalfall:

- einen Benutzernamen, der ihn eindeutig identifiziert
- ein Passwort, mit dem er sich authentifizieren kann

Die Authentifizierung beim System kann auch über Multi-Factor-Authentication erfolgen, was bedeutet, dass unter Umständen neben einem Passwort weitere Daten zur Authentifizierung des Benutzers vorliegen müssen. Der Normalfall ist aber die Authentifizierung allein über das Passwort.

Weitere Verwaltungsinformationen für einen Benutzer können eine E-Mail-Adresse enthalten oder die Zugehörigkeit zu bestimmten Benutzergruppen. Da Enterprise Password Manager das Anlegen von Rollen oder Gruppen unterstützen, müssen auch diese Informationen verwaltet werden. Wichtig ist, welche Gruppen mit welchem Namen es gibt und welche Zugriffsberechtigungen die einzelnen Gruppen haben (dies gilt ebenfalls für Rollen). Ein Passwort-Eintrag in der Datenbank besteht mindestens aus

- Passwort
- zugehörigem Account oder Komponente
- Titel des Eintrags

Oft gibt es auch die Möglichkeit, zusätzliche Notizen anzulegen. Weitere Daten sind zum Beispiel das Datum der Erstellung des Kennworts, der letzten Modifikation, des letzten Zugriffs oder auch wann das Passwort wieder geändert werden muss. Ebenfalls verwaltet werden müssen Daten zur Zugriffsberechtigung auf dieses Passwort.

Gerade für Auditzwecke müssen noch zusätzlich weitere Daten gespeichert werden. Allgemein ist es wichtig zu sehen, welcher Benutzer eine bestimmte Aktivität durchgeführt hat (siehe 3.2.6). Im Besonderen ist zum Beispiel erforderlich, nachverfolgen zu können, ob ein Benutzer eine Anfrage auf die Freigabe eines Passworts gestellt hat, ob diese Anfrage zugelassen wurde und wer einen Zugriff autorisiert hat.

4.2.1 Vertraulichkeitsstufen der gespeicherten Daten

Grundsätzlich sollten diese Daten nicht extern bekannt sein. Einige Informationen sind aber auch vor externen Personen nicht zu verbergen, so ist beispielsweise die Organisationsstruktur mit den verschiedenen Abteilungen und Arbeitskreisen und ihren Aufgaben offen auf der Webseite einsehbar. Auch die Namen mancher Mitarbeiter und ihre Abteilungszugehörigkeit lassen sich herausfinden, außerdem die Existenz mehrerer Accounts und Komponenten, deren Passwörter verwaltet werden. Ein Beispiel dafür wäre die Existenz verschiedener Mailinglisten, die auch extern bekannt sind. Es ist klar, dass zu der Verwaltung der Listen die entsprechenden Accounts und Passwörter benötigt werden.

Diese Fälle sollen hier aber nicht weiter betrachtet werden. Wichtig ist vielmehr die interne Verwaltung der Daten. Dabei kann die Verfügbarkeit der Daten in drei grobe Kategorien eingeteilt werden. Manche Informationen sind intern allen Mitarbeitern vollständig bekannt oder lassen sich leicht herausfinden, dazu gehören die Organisationsstruktur, die Namen und Tätigkeiten anderer Mitarbeiter sowie die Existenz mancher Accounts und Komponenten.

Andere Daten wiederum sind nur einem eingeschränkten Kreis bekannt. Manche Accounts werden generell von einem beschränkten Personenkreis verwaltet und nur Mitglieder dieser Gruppe wissen über diesen Account Bescheid. Sowohl die Information, dass dieser Account existiert als auch die zugehörigen Verwaltungsinformationen sind dann nur in diesem Kreis bekannt. Diese Verwaltungsinformationen sind die mit einem Account verbunden Informationen wie beispielsweise der Zeitpunkt der nächsten automatischen Passwortänderung. Eben-

falls verborgene Informationen sind die Daten, die zu Auditzwecken gesammelt werden. Auch diese Daten sind nur einer bestimmten Personengruppe einsehbar. Für diese Daten müssen also Mechanismen zur Zugriffskontrolle existieren.

In eine dritte Kategorie lassen sich die Daten einteilen, die innerhalb einer Gruppe nur bestimmten Mitgliedern bekannt sind. Dazu zählen die Passwörter selbst, die nur den Personen bekannt sein sollten, die auch Zugriff auf den Account haben. Ebenfalls in diese Kategorie fällt das User-Passwort, das zum Login des einzelnen Benutzers nötig ist und auch nur dieser einen Person bekannt ist. Diese Daten müssen verborgen werden und sie haben besonderen Schutzbedarf.

Eine visuelle Darstellung dieser Vertraulichkeitsstufen findet sich in Abbildung 4.2.

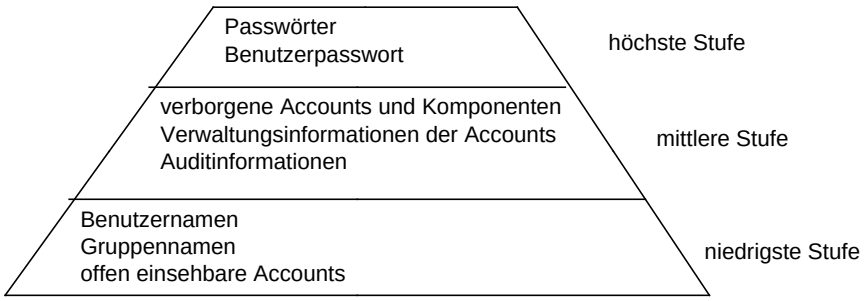

Abbildung 4.2: Die drei Vertraulichkeitsstufen.

Da nicht alle Daten allen Mitarbeitern bekannt sein sollen, ergibt sich die Forderung nach einer Zugriffskontrolle. Um den Zugriff einschränken und kontrollieren zu können, müssen Benutzer vom System authentifiziert und autorisiert werden. Um den Schutz der Daten zu gewährleisten, muss Vertraulichkeit gegeben sein.

4.2.2 Vertraulichkeitsstufen der Passwort-Einträge

Eine weitere Frage an dieser Stelle ist, auf welcher Ebene sich bestimmte Passwort-Einträge einstufen lassen. Für die Einstufung eines Eintrags sind die mit diesem verbundenen Metadaten erforderlich, vor allem die Informationen zu den Zugriffsberechtigungen. Ein Eintrag, der von vielen Personen eingesehen werden kann, ist auf einer eher niedrigen Stufe einzuordnen. Es ergibt wenig Sinn, einen solchen bekannten Account zu verbergen. Daraus folgt, dass ein Eintrag, der nur wenigen Personen bekannt ist, einer höheren Stufe zugewiesen wird.

Ein Account, dessen Passwort von vielen Mitarbeitern geändert werden kann, ist ebenfalls eher auf einer niedrigen Stufe eingestuft, wohingegen ein Account, für den hauptsächlich Leseberechtigungen vorliegen, höher zu klassifizieren ist.

Trotzdem kann es auch wichtig sein, einen Account, für den viele Personen das Passwort benötigen, auf einer höheren Stufe einzuordnen, falls das dahinter stehende System kritisch ist. Die Identität und die Wichtigkeit des Systems müssen also ebenfalls in die Bewertung miteinbezogen werden. Die Frage, welche Funktionen von einem System ausgeführt werden, ist also auch zu betrachten.

Allgemein kann sich also eine Einstufung auf folgende Punkte stützen:

- die Identität des Systems: welche Funktionen von einem System ausgeübt werden
- die Wichtigkeit des Systems: wie kritisch wäre ein Ausfall dieses Systems
- die Anzahl der Zugriffsberechtigungen: je mehr Personen Berechtigung haben, desto niedriger die Stufe.
- und die Frage, welche Berechtigungen vergeben wurden: je beschränkter die Berechtigungen, desto höher die Stufe.

Zugriffsberechtigungen werden oft in einer sogenannten Zugriffsmatrix dargestellt. Ein Beispiel für eine solche Matrix ist in Abbildung 4.3 zu sehen. In diesem Fall wäre Eintrag 1

	Eintrag 1	Eintrag 2	Eintrag 3
Person A	lesen, schreiben		lesen
Person B	lesen, schreiben, löschen	lesen, schreiben, löschen	lesen
Person C	lesen, schreiben	lesen	
Person D	lesen, schreiben, löschen		

Abbildung 4.3: Beispiel einer Zugriffsmatrix.

auf der niedrigsten Stufe einzuordnen, Eintrag 2 auf der mittleren Stufe und Eintrag 3 auf der höchsten Vertraulichkeitsstufe. Eine solche Zugriffsmatrix würde sich auch mithilfe eines Algorithmus auslesen lassen, um die Stufen der Einträge zu bestimmten. Um die Einträge einzuordnen, muss also zunächst festgestellt werden:

- die Anzahl der Benutzer, die von einem Ausfall des Systems betroffen wären
- die Anzahl der betroffenen Dienste
- die Anzahl der Personen, die Zugriffsberechtigung haben
- welche Berechtigungen für diesen Eintrag vergeben wurden

Ein Eintrag, der auf der niedrigen Stufe eingeordnet ist, ist nicht verborgen, alle Mitarbeiter können sehen, dass dieser Eintrag existiert. Ein Eintrag auf der mittleren Stufe ist verborgen, nur bestimmte Personen können ihn überhaupt sehen. Weil der Eintrag damit nur wenigen Personen bekannt ist, wird zusätzlicher Schutz für den dahinter liegenden Account erreicht. Für das Verbergen des Eintrags ist eine gute Zugriffskontrolle unbedingt erforderlich.

Auf der höchsten Stufe kann man zusätzlich zum Verbergen des Eintrags noch Verfahren des Secret-Sharing verwenden. Der zum Entschlüsseln des Passworts benötigte Schlüssel kann in Teilgeheimnisse aufgeteilt werden, sodass keine Person alleine auf den Account zugreifen kann. Das bedeutet, dass zur Entschlüsselung des Eintrags die Kooperation mehrerer Berechtigter erforderlich ist. Die Aufteilung des Schlüssels kann beispielsweise nach dem Schwellenschema von Shamir erfolgen (siehe 4.1.3). Die Teilgeheimnisse müssen entsprechend verteilt und von den berechtigten Personen sicher aufbewahrt werden.

Dabei kann sich das Problem ergeben, dass eine berechtigte Person nicht mehr verfügbar ist, zum Beispiel weil sie das Unternehmen verlassen hat. In diesem Fall gibt es keine Möglichkeit mehr, an das Teilgeheimnis dieser Person zu kommen. Dann muss das Geheimnis rekonstruiert und neu aufgeteilt werden.

Selbstverständlich sind die Passwörter in den Einträgen auf jeden Fall verschlüsselt. Die Zugriffskontrolle wird bei Passwort-Management-Tools im Normalfall von der Anwendung selbst übernommen, die in diesem Fall ein vertrauenswürdiger Dritter ist. Dabei werden die Zugriffsberechtigungen entweder pro Benutzer auf Basis seiner Identität vergeben (Discretionary Access Control) oder anhand festgelegter Regeln und Richtlinien (Mandatory Access Control). Eine weitere Möglichkeit ist die Vergabe von Berechtigungen anhand der Rollen, die eine Person im System einnimmt (Role-Based Access Control).

Soll auf einen vertrauenswürdigen Dritten verzichtet werden, gestaltet sich die Zugriffskontrolle schwieriger. In kryptografischen Modellen wird das durch Attribute-Based Encryption möglich gemacht (siehe 4.4).

4.3 Kryptografische Absicherung

Ein System der Passwortverwaltung ist grundsätzlich so aufgebaut, dass es eine zentrale Struktur gibt, in der Passwörter gespeichert werden. Mehrere Benutzer können entsprechend ihrer Berechtigungen darauf zugreifen.

Im Normalfall wird bei Passwort-Management-Tools die Zugriffskontrolle von der Anwendung übernommen. Der Benutzer muss sich bei ihr authentisieren und wird von ihr entsprechend seiner Berechtigungen autorisiert. Die Passwörter liegen meist in einer Datenbank vor, wo sie von der Anwendung abgerufen und an den Benutzer übermittelt werden. Kryptografisch abgesichert werden muss also die Authentifizierung der Benutzer, die Speicherung der Daten, die Kommunikation und schließlich die Verwaltung der Schlüssel.

Soll auf eine Anwendung als vertrauenswürdigen Dritten verzichtet werden, ist die Modellierung eines solchen Systems deutlich schwieriger. Wie ein Modell aussehen könnte, das auf einen vertrauenswürdigen Dritten verzichtet, ist in 4.4 beschrieben.

Im Folgenden wird die kryptografische Absicherung eines Modells der Passwortverwaltung dargestellt, wobei davon ausgegangen wird, dass eine Anwendung existiert, die das Management der Kennwörter übernimmt.

4.3.1 Authentifizierung

Die Authentifizierung des Benutzers ist wichtig zur Zugriffskontrolle. Der Benutzer authentisiert sich am Server, dieser verifiziert seine Identität und autorisiert den Benutzer, weist ihm also auf dieser Grundlage seine Zugriffsberechtigungen zu. Die Authentifizierung kann mehrere Mechanismen umfassen:

- Wissen: eine Information, die der Benutzer besitzt, beispielsweise ein Passwort, eine PIN oder die Antwort auf eine Sicherheitsfrage
- Besitz: ein Gegenstand, der sich im Besitz des Benutzers befindet, beispielsweise ein Hardware-Token
- Eigenschaft: bestimmte Eigenschaften, über die ein Benutzer verfügt, beispielsweise biometrische Merkmale wie ein Fingerabdruck

Authentifizierung, die sich auf zwei oder mehr Faktoren stützt, wird entsprechend Two-Factor-Authentication oder Multi-Factor-Authentication genannt.

Im Normalfall erfolgt die Authentifizierung über die Eingabe von Login und Passwort. Grundsätzlich können aber alle diese Mechanismen zur Authentifizierung eingesetzt werden und viele Tools zum Passwort-Management unterstützen Two-Factor-Authentication.

Das vom Benutzer eingegebene Passwort wird vom Server überprüft. Dazu sollte auf Serverseite ein Hash gespeichert werden, das Passwort sollte auf keinen Fall im Klartext auf dem Server vorliegen. Wenn der Benutzer das Passwort eingegeben hat, berechnet der Server den zugehörigen Hashwert und vergleicht diesen mit dem vorliegenden Hash. Stimmen die Werte überein, wird Zugriff auf die Datenbank je nach Berechtigung gewährt.

Vorteil dieses Verfahrens ist, dass ein Angreifer, der den Server kompromittiert, nicht Zugriff auf die Passwörter bekommt. Aus dem Hash des Passworts kann (zumindest wenn ein geeignetes Verfahren gewählt wird), nicht das Passwort berechnet werden. Die einzige Möglichkeit für einen Angreifer, auf das Passwort zu kommen, wäre ein Brute-Force-Angriff, bei dem der Hashwert von vielen möglichen Passwörtern berechnet und verglichen wird. Das macht es einerseits nötig, ein Passwort gut zu wählen (siehe 2.1), andererseits kann ein guter Schutz der Passwörter dadurch gewährleistet werden, dass zusätzlich Salzen verwendet wird. Dabei wird das Passwort vor dem Hashen mit einer zusätzlichen zufälligen Zeichenfolge versehen.

Zum Hashen des Passwort empfiehlt das BSI folgende Algorithmen: [Bun08]

- SHA-224
- SHA-256
- SHA-384
- SHA-512

Selbstverständlich sollte das Passwort nur verschlüsselt und keinesfalls offen über einen ungesicherten Kanal übertragen werden. Das schließt aus, dass ein Angreifer die Kommunikation abhört und das Passwort dadurch erhalten kann.

4.3.2 Speicherung der Daten

Es muss die Frage entschieden werden, in welchem Umfang Verschlüsselungstechniken bei der Speicherung von Passwörtern und Daten allgemein eingesetzt werden. Im Normalfall werden die Passwörter in einer Datenbank gespeichert. Es gibt die Möglichkeit, sowohl die gesamte Datenbank zu verschlüsseln als auch die Verschlüsselung auf bestimmte Daten zu beschränken. Für den zweiten Fall muss überlegt werden, welche Daten verschlüsselt werden sollten und welche nicht.

Die Verschlüsselung der gesamten Datenbank schützt alle dort gespeicherten Daten. Das dient vor allem dazu, die Daten zu sichern, falls der Datenträger, auf dem sie vorhanden sind, verloren geht. Bei jedem Zugriff muss die Datenbank dann zuerst entschlüsselt werden, bevor die Daten genutzt werden können. Die Verschlüsselung der gesamten Datenbank ergibt also wenig Sinn, da eine Entschlüsselung der Datenbank zu zeitintensiv ist.

Das Beschränken der Verschlüsselung auf bestimmte Daten ist also sinnvoller. Es ist klar, dass die in der Datenbank gespeicherten Passwörter besonderen Schutz genießen müssen. Die Vertraulichkeit dieser Daten ist besonders wichtig, sie sollten also auf jeden Fall verschlüsselt werden. Wird die Datenbank entwendet, sind die Passwörter also weiterhin geschützt und die Vertraulichkeit der Daten ist gewährleistet. Vor allem im Fall eines erfolgreichen Angriffs per SQL Injection ist die Verschlüsselung der Kennwörter wichtig.

Der Schutz von verborgenen Accounts muss durch Zugriffskontrolle erreicht werden. Der Server muss die Benutzer geeignet autorisieren, damit nur diejenigen Benutzer Einträge sehen können, die dazu Berechtigung haben. Ebenso sollte die Zugriffskontrolle gewährleisten, dass Auditinformationen nur von berechtigten Personen eingesehen werden können.

Eine weitere Möglichkeit zum Schutz von Daten und Einträgen wäre die zusätzliche Verschlüsselung von verborgenen Passwort-Einträgen und Auditinformationen. Eine solche Maßnahme würde allein der Abwehr externer Angreifer dienen, gegen internen Missbrauch kann Verschlüsselung nicht schützen. Eine zum Zugriff berechtigte Person hat in einem System nicht mehr oder weniger Berechtigungen durch zusätzliche Verschlüsselung der Daten.

Verschlüsselung kann aber die Daten und die Informationen zu den dahinter liegenden Accounts im Falle der Kompromittierung des Servers zusätzlich sichern. Ebenfalls kann mit Verschlüsselung der Audit-Logs verhindert werden, dass Angreifer zusätzliche Informationen über im System vorhandene Accounts und die Zugriffsberechtigungen darauf erhalten. Der Angreifer müsste erst diese Daten entschlüsseln, bevor er sie einsehen kann.

Die Verschlüsselung der verborgenen Accounts und Auditinformationen ist also eine Maßnahme, die zusätzlich zum Schutz der Daten vor externen Angreifern erfolgen kann.

Eine weitere Frage ist, auf welcher Ebene Ver- und Entschlüsselung ablaufen. Auch dafür gibt es grundsätzlich zwei Möglichkeiten. Diese Prozesse können entweder von der Datenbank oder durch die Anwendung ausgeführt werden. Wenn auf Datenbankebene Ver- und Entschlüsselung durchgeführt werden, müssen die Verschlüsselungsschlüssel entweder zur Datenbank übertragen werden oder gleich mit den Daten gespeichert werden. Daten werden dann entschlüsselt und zur Anwendung gesendet.

Wenn diese Verfahren auf Anwendungsebene ablaufen, liegen die benötigten Schlüssel auf dieser Ebene vor. Die Anwendung ruft die verschlüsselten Daten aus der Datenbank ab und entschlüsselt sie selbst. Ebenso werden die Daten von der Anwendung selbst schon verschlüsselt gespeichert. Werden die Schlüssel mit den Daten gespeichert, bekommt ein Angreifer, der sich Zugriff auf die Datenbank verschafft, auch Zugriff auf die Schlüssel und kann sie gleich nutzen, um die Daten zu entschlüsseln. Aber auch eine Anwendung kann gehackt werden und die benötigten Schlüssel abgegriffen werden. Es ist also äußerst wichtig, entsprechende Schutzmaßnahmen zu ergreifen, um die Anwendung abzusichern.

Eine weitere Möglichkeit besteht darin, den Schlüssel vom Benutzerpasswort abhängig zu machen. In dem Fall ist der Verschlüsselungsschlüssel zusätzlich mit dem Passwort gesichert. Das nützt allerdings wenig, falls der Benutzer ein schwaches Passwort besitzt oder der Angreifer den Account des Benutzers kompromittiert hat. Alle diese Alternativen haben also Nachteile. Eine vollständige Sicherheit der Verschlüsselungsschlüssel ist schwer zu gewährleisten.

Die Sicherheit der Verschlüsselung hängt auch davon ab, welches Verfahren genutzt wird. Zur Verschlüsselung würden sich sowohl symmetrische als auch asymmetrische Verfahren eignen. In diesem Fall ist ein symmetrisches Verschlüsselungsverfahren sinnvoller, da dieses Verfahren performanter ist als asymmetrische Verfahren. Die Problematik bei der Verwendung von symmetrischer Verschlüsselung, der Schlüsselaustausch, fällt hier nicht ins Gewicht, da Ver- und Entschlüsselung auf dem Server stattfinden. Die Schlüssel müssen also auf Serverseite gespeichert werden. Asymmetrische Verschlüsselung bietet also keinerlei Vorteile.

Das BSI empfiehlt für symmetrische Verfahren folgende Algorithmen bei Einsatz von Blockchiffren: [Bun08]

- AES-128, AES-192, AES-256
- SERPENT-128, SERPENT-192, SERPENT-256
- Twofish-128, Twofish-192, Twofish-256

Stromchiffren werden generell nicht empfohlen. Weitere Empfehlungen für die Wahl der Betriebsmodi, des Initialisierungsvektor und Padding-Verfahren sind auf der Webseite des BSI zu finden (siehe [Bun08]).

Verschlüsselungsverfahren nützen allerdings nur bedingt. Wenn ein Angreifer einen Account kompromittiert hat, hat er Zugriff auf alle Passwörter, für die ein Benutzer Berechtigungen hat. Es ist also wichtig, den Zugriff auf Ressourcen sowohl zur Abwehr von externen Angreifern als auch zur Verhinderung von internem Missbrauch einzuschränken. Zu einer geeigneten Zugriffskontrolle gehört also das Beschränken von Berechtigungen auf die für die Ausführung einer Aufgabe benötigten Berechtigungen und zusätzlich die Überwachung von Zugriffen, beispielsweise mit Audit-Logs.

4.3.3 Kommunikation

Grundsätzlich ist hier die Kommunikation zwischen dem Benutzer und dem System zu betrachten. Bei Passwort-Management-Lösungen, die dem System das automatische Update des Passworts auf Zielsystemen erlauben, muss auch die Kommunikation zwischen System und Zielsystem miteinbezogen werden. Um System und Komponenten miteinander kommunizieren zu lassen, müssen native Protokolle genutzt werden, die von den Komponenten unterstützt werden. Diese Protokolle sind mal mehr, mal weniger sicher. Zur Absicherung der Übertragung der Passwörter sollten bei wenig sicheren Protokollen zusätzlich Protokolle wie IPSec benutzt werden, um die Vertraulichkeit der Daten zu schützen. IPSec macht es möglich, Informationen sicher auch über ungesicherte Kanäle zu übertragen.

Die Kommunikation zwischen Benutzer und Server sollte verschlüsselt ablaufen. Besonders wichtig ist, dass die Passwörter nur verschlüsselt übertragen werden. Das bezieht sich sowohl auf das Passwort, das der Benutzer zum Login verwendet, als auch auf die abgerufenen Passwörter aus der Datenbank. Beide Kommunikationspartner müssen sich vorher authentisieren, der Benutzer kann dies über die Eingabe von Login und Passwort machen. Eine Möglichkeit zur gegenseitigen Authentisierung ist Kerberos, bei der eine zentrale Autorität Tickets ausstellt, die zur gegenseitigen Authentisierung genutzt werden können.

Allgemein können zur Verschlüsselung der Kommunikation symmetrische, asymmetrische und hybride Verfahren genutzt werden. Aus den in 4.1.1 besprochenen Gründen eignen sich hybride Verfahren am Besten. Werden symmetrische Verfahren eingesetzt, eignen sich laut BSI die genannten Verfahren von 4.3.2. Bei asymmetrischer Verschlüsselung können laut BSI folgende Verfahren eingesetzt werden: [Bun08]

- RSA-Verschlüsselungsverfahren
- Elliptic Curve Integrated Encryption Schemes
- Discrete Logarithm Integrated Encryption Schemes

Für den Austausch von Schlüssel wird allgemein das Diffie-Hellman-Verfahren oder EC Diffie-Hellman empfohlen.

Mithilfe der Verschlüsselung wird die Vertraulichkeit der übertragenen Daten gewährleistet, zum Absichern der Integrität können Signaturen oder Message Authentication Codes verwendet werden. Bei Einsatz von MAC-Verfahren empfiehlt das BSI eine Schlüssellänge von mindestens 16 Byte sowie entweder eine der unter 4.3.2 aufgeführten Blockchiffren oder eine der unter 4.3.1 genannten Hash-Verfahren. [Bun08]

Es bietet sich an, bereits vorhandene Protokolle wie beispielsweise SSL zur Kommunikation zu verwenden. Bei SSL authentifiziert sich der Server mittels eines Zertifikats, zum Schlüsselaustausch wird Diffie-Hellman angewendet. Die Kommunikation zwischen Server und Benutzer läuft danach symmetrisch ab.

4.3.4 Verwaltung der Schlüssel

Schlüsselmanagement beschäftigt sich mit all den Stadien, die ein Schlüssel während seiner Lebensspanne durchläuft. Dazu gehören

- Generierung
- Verteilung und Austausch
- Aufbewahrung
- Wechsel
- Archivierung und Zerstörung des Schlüssels.

Zur Passwortverwaltung wird auf jeden Fall ein Schlüssel benötigt, der die Daten verschlüsselt. Ebenfalls muss die Kommunikation zwischen Benutzer und System verschlüsselt ablaufen, es werden also auch Schlüssel zur Sicherung dieser Kommunikation gebraucht.

Bei der Generierung der Schlüssel ist es vor allem wichtig, den Schlüssel möglichst zufällig zu wählen. Das bedeutet, dass ein geeigneter Generator verwendet werden muss, der zufällige Schlüssel generieren kann. Am besten eignet sich ein physikalischer Generator, der beispielsweise elektromagnetische, elektromechanische oder quantenmechanische Effekte nutzt. Auf gar keinen Fall sollte der Schlüssel etwa durch Eingabe des Benutzerpassworts erfolgen.

Zufälligkeit ist zur Wahl des Initialisierungsvektors nötig. Manche Verschlüsselungsverfahren beruhen zudem auch auf bestimmten mathematischen Operationen, für die Zufälligkeit nötig ist, etwa die Wahl der Primzahlen bei RSA.

Ein Schlüssel sollte in möglichst sicherer Umgebung erzeugt werden. In der Praxis geschieht die Erzeugung entweder durch eine zentrale Stelle oder durch den Benutzer selber. Wird der Schlüssel von einer zentralen Stelle erzeugt, muss dieser sicher an den Benutzer verteilt werden. Zentrale Stellen haben dabei den Vorteil, dass die Erzeugung von Schlüsseln durch virenfreie Rechner und geeignete Generatoren garantiert werden kann, allerdings muss der Schlüssel erst sicher zum Benutzer transportiert werden. Dafür ist bei der Generierung des Schlüssel durch den Benutzer selbst die sichere Umgebung unter Umständen nicht gegeben. Allerdings ist es heutzutage möglich, geeignete Hardware wie beispielsweise bestimmte Chipkarten zu verwenden, die ebenfalls eine sichere Umgebung bieten.

Wichtig bei der Generierung von Schlüsseln ist auch, dass der erzeugte Schlüssel eine ausreichend große Länge hat. Wie lang dabei ein Schlüssel sein sollte, hängt davon ab, welche Funktion er erfüllen soll. Gerade Schlüssel, die über einen langen Zeitraum gespeicherte Daten sichern sollen, müssen eine entsprechende Länge haben. Die zur symmetrischen Verschlüsselung empfohlenen Algorithmen in 4.3.2 sollten eine Mindestlänge von 128 Bit haben. Welche Länge die Schlüssel haben sollten, die zur Kommunikation verwendet werden, hängt von dem verwendeten Verfahren ab.

Die Verteilung von Schlüsseln ist bei geheimen Schlüsseln schwierig. Geheime Schlüssel müssen dem entsprechenden Empfänger sicher überbracht werden. Dazu muss der Empfänger

sich entsprechend authentisieren. Die Verteilung kann über geeignete Datenträger oder verschlüsselte Verbindungen erfolgen. Bei der Übertragung von symmetrischen Schlüsseln sollten diese asymmetrisch verschlüsselt werden und auf gar keinen Fall über einen ungesicherten Kanal gesendet werden.

Bei asymmetrischen Verfahren muss zusätzlich der öffentliche Schlüssel öffentlich gemacht werden. Das kann beispielsweise individuelle Verteilung oder durch eine Public Key Infrastructure (PKI) geschehen.

Die Schlüssel sollten selbstverständlich so aufbewahrt werden, dass nur autorisierte Personen darauf zugreifen können. Dazu werden die Schlüssel häufig selbst verschlüsselt, der dafür genutzte key-encryption-key muss dann ebenfalls sicher verwahrt werden.

Die Schlüssel können auch auf entsprechender Hardware gespeichert werden, beispielsweise auf Chipkarten oder speziellen PC-Einsteckkarten. Auf gar keinen Fall sollten Schlüssel unverschlüsselt im System vorhanden sein. Es sollte ebenfalls vermieden werden, dass der zur Verschlüsselung der Daten genutzte Schlüssel zusammen mit den Daten gespeichert wird. Das bedeutet, dass der Verschlüsselungsschlüssel am Besten separat abgespeichert wird, wobei aber beachtet werden muss, dass die Anwendung zur Entschlüsselung Zugriff auf den Schlüssel benötigt.

Da der Verlust eines Schlüssels dazu führt, dass die damit verschlüsselten Daten nicht mehr entschlüsselt werden können, muss überlegt werden, ob ein Backup der Schlüssel sinnvoll ist. Jede angelegte Kopie der Schlüssel muss aber wieder entsprechend gesichert werden, damit sie nicht Angreifern in die Hände fällt. Es ergibt Sinn, von dem zur Verschlüsselung der Daten benötigten Schlüssel eine Sicherheitskopie anzulegen, da der Verlust dieses Schlüssels fatal wäre.

Wie oft ein Schlüsselwechsel stattfinden sollte, hängt von mehreren Faktoren ab, beispielsweise welche Funktionen er erfüllt, welcher Algorithmus verwendet wird und wie häufig er eingesetzt wird. Die Sitzungsschlüssel, die dazu verwendet werden, die Kommunikation zwischen den Beteiligten zu sichern, sollten nach jeder Sitzung geändert werden.

Wird der für die Verschlüsselung der Daten verwendete Schlüssel gewechselt, müssen alle Daten neu verschlüsselt werden. In diesem Fall sollte der neue Schlüssel eine Zeitlang getestet werden, bevor der alte Schlüssel gelöscht wird. Es ist wichtig, in diesem Fall sicherzugehen, dass keine Daten mehr vorhanden sind, die noch mit dem alten Schlüssel verschlüsselt wurden und die nach der Zerstörung nicht mehr zu entschlüsseln sind. Der alte Schlüssel sollte also für einen gewissen Zeitraum archiviert werden, bevor er zerstört wird.

Dabei muss das Löschen des Schlüssels auf sichere Art und Weise erfolgen. Der Datenträger, auf dem der Schlüssel vorliegt, sollte überschrieben oder zerstört werden.

4.4 Theoretisches Modell zur Zugriffskontrolle ohne vertrauenswürdigen Dritten

Mithilfe der genannten kryptografischen Maßnahmen wird eine Absicherung der Passwortverwaltung bei der Übernahme der Zugriffskontrolle durch einer Anwendung möglich gemacht. Ein grundlegendes Problem bei der Zugriffskontrolle durch eine Anwendung oder einem Server besteht aber darin, dass die Anwendung oder der Server die Rolle eines vertrauenswürdigen Dritten übernimmt. Geht man aber davon aus, dass die Daten auf einem Server liegen, dem nicht vertraut werden kann oder der kompromittiert werden könnte, muss der Zugriff auf die Daten anders geregelt werden. Es müssen also Verschlüsselungstechniken so eingesetzt werden, dass selbst auf einem nicht vertrauenswürdigen Server die Daten sicher sind.

Das Problem an dieser Stelle ergibt sich aus der Forderung nach Zugriffskontrolle auf einer feinen Ebene. Wenn ein Nutzer Passwort-Einträge anlegt und verschlüsselt, müsste er den Schlüssel an alle Personen weitergeben, die Zugriff auf die Daten haben sollen. Diese Personen könnten anschließend mit dem Schlüssel alle Passwort-Einträge des Nutzers entschlüsseln. Eine einfache Verschlüsselung der Daten ohne feinmaschige Zugriffskontrolle ist also in diesem Fall nicht umzusetzen.

In kryptografischen Modellen wird eine feingranulare Zugriffskontrolle durch Attribute-Based Encryption möglich gemacht.

4.4.1 Aufbau des Systems

Grundsätzlich eignen sich sowohl KP-ABE als auch CP-ABE-Verfahren für das Management von Passwörtern. Das hier verwendete Modell setzt das von Yu et al vorgeschlagenen Schema ein, das auf KP-ABE aufbaut. [SYL10] Dieses Schema hat den Vorteil, dass es gut skalierbar und auch für Modelle mit vielen Benutzern verwendbar ist. Das wird erreicht, indem ein Teil der Operationen auf den Server ausgelagert wird. Eine weitere Umsetzung dieses Schemas wird in [SC12] dargestellt.

Dieses Schema setzt hybride Verschlüsselungstechnik ein. Die Daten auf dem Server werden symmetrisch verschlüsselt, der zur Verschlüsselung benötigte Schlüssel mit KP-ABE. Das Ziel ist, die Vertraulichkeit der Daten in jedem Fall zu schützen. Der Server darf die Daten nicht selbst entschlüsseln können, ebenso muss vermieden werden, dass mehrere Benutzer miteinander kooperieren, um Daten zu entschlüsseln, für die sie allein keinen Berechtigung hätten.

Das System besteht aus einem oder mehreren Servern, mehreren Benutzern, die Lese- und Schreibberechtigung haben können, sowie einem Ersteller, der die Berechtigung hat, neue Passwort-Einträge anzulegen und zu löschen. Die Verwaltung der Passwort-Einträge wird also vom Ersteller übernommen. Es ist möglich, dass Benutzer ihre Zugriffsberechtigung verlieren. Die Server sind ständig verfügbar und es wird angenommen, dass sie curious-but-honest sind. Das bedeutet, dass die Server einerseits versuchen, so viel wie möglich über

die auf ihnen gespeicherten Daten zu erfahren, andererseits aber alle Operationen, die ihnen übertragen werden, ehrlich durchführen.

Der Ersteller hat die Berechtigung, Daten anzulegen und zu verschlüsseln. Anhand der Attribute, mit denen ein Eintrag verschlüsselt wird, kann er festlegen, welche Benutzer auf die Daten zugreifen können. Um Zugriff zu erhalten, müssen Benutzer die Daten von dem Server herunterladen und mithilfe ihres geheimen Schlüssels wieder entschlüsseln. Der geheime Schlüssel ist dabei als Zugriffsstruktur zu verstehen, der ihre Berechtigungen festlegt. Ändern die Benutzer die Daten, müssen sie diese signieren und wieder verschlüsselt hochladen.

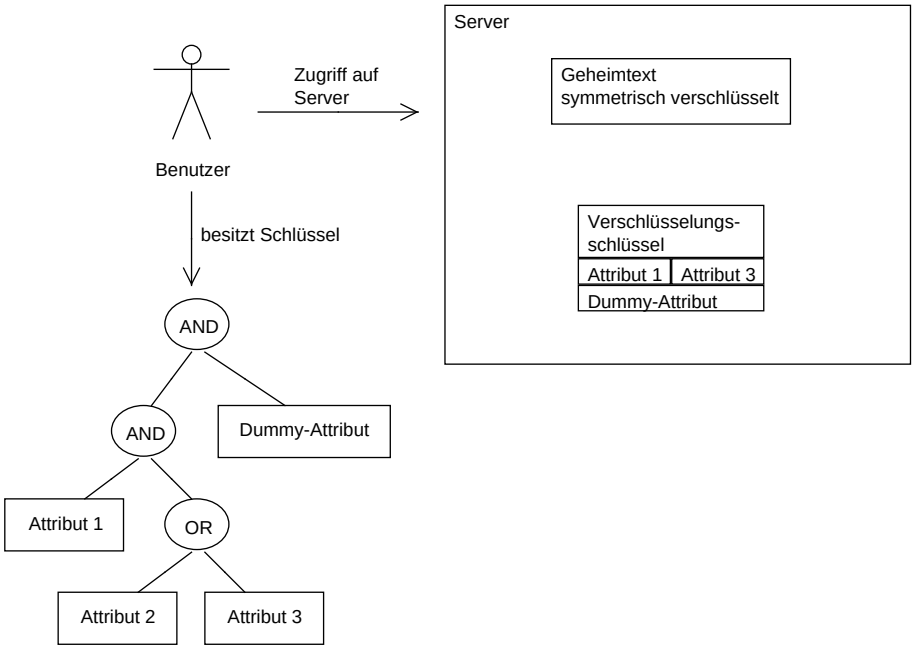

Abbildung 4.4: Beispiel: Entschlüsselung eines Geheimtextes durch den Benutzer.

Ein Beispiel dieses Modells ist in Abbildung 4.4 zu sehen. Der Benutzer besitzt einen geheimen Schlüssel, der aus der Zugriffsstruktur Attribut 1 AND (Attribut 2 OR Attribut 3) sowie einem zusätzlichen Dummy-Attribut besteht. Mit diesem Schlüssel kann der Verschlüsselungsschlüssel (und damit dann anschließend der symmetrisch verschlüsselte Geheimtext) entschlüsselt werden, der mit Attribut 1 und Attribut 2 verschlüsselt ist oder ein Schlüssel, der mit Attribut 1 und Attribut 3 verschlüsselt ist. Der in diesem Beispiel auf dem Server vorliegende Geheimtext kann also entschlüsselt werden.

Ein sich dabei ergebendes Problem ist der hohe Verwaltungsaufwand für den Ersteller der Daten. Er muss die Schlüssel und Daten verwalten, was aufwändig ist. Beispielsweise muss der Ersteller Schlüssel generieren, verteilen und bei einer Änderung der Zugriffsberechtigungen diese aktualisieren und Daten neu verschlüsseln. Dies ist ein besonderes Problem, wenn ein Benutzer gelöscht werden soll. Um diese Aufgaben durchzuführen, müsste ein Ersteller unter Umständen ständig online und verfügbar sein.

Das widerspricht dem Wunsch nach einem effizienten System. Um also den Erstellern eine leichtere Verwaltung der Daten und Schlüssel zu ermöglichen, werden manche Operationen an den Server ausgelagert. Trotz dieser Auslagerung ist es den Servern aber unmöglich, die Daten entschlüsseln zu können. Die Vertraulichkeit bleibt also geschützt. So kann dem Server die Aufgabe, bei einer Änderung der Zugriffsberechtigung die Schlüssel zu aktualisieren und die Daten neu zu verschlüsseln, übertragen werden.

Dieser Vorgang wird als Proxy Re-Encryption (PRE) bezeichnet. Dabei wandelt der Proxy einen Geheimtext, der mit einem bestimmten öffentlichen Schlüssel verschlüsselt wurde, in einen Geheimtext um, der von dem geheimen Schlüssel eines anderen Benutzers entschlüsselt werden kann.

Die Server speichern eine Benutzerliste, die die IDs der validen Benutzer enthält. Die Benutzerliste wird damit zum Verwalten aller Personen im System verwendet. Ebenfalls auf dem Server gespeichert wird eine teilweise Kopie der geheimen Schlüssel der Benutzer, um den Entzug von Zugriffsberechtigungen möglich zu machen. Außerdem existiert für jedes Attribut eine eigene Historie, in der die Attribute mit ihrer aktuellen Versionsnummer sowie PRE-Schlüssel aufgelistet sind.

Alle Attribute im System erhalten eine Versionsnummer, mit Ausnahme des sogenannten Dummy-Attributs. Diese Versionsnummer wird benötigt, um den Entzug von Zugriffsberechtigungen für einen Benutzer möglich zu machen. Zusätzlich zu diesen Attributen existiert auch noch ein Dummy-Attribut, das für die Verschlüsselung aller Daten benutzt wird. In jedem Set an Attributen, mit dem Daten verschlüsselt werden, muss dieses Dummy-Attribut enthalten sein. Dieses zusätzliche Attribut dient der vereinfachten Verwaltung der Schlüssel und ist das einzige Attribut, das keine Versionsnummer erhält. Das Dummy-Attribut ist außerdem nicht in den Kopien der geheimen Schlüssel der Benutzer auf dem Server enthalten.

Jedes Attribut entspricht einer Komponente, die im öffentlichen Schlüssel des Systems enthalten ist, bei der Verschlüsselung werden die Dateien mit Attributen verschlüsselt, indem die zugehörigen Komponenten des öffentlichen Schlüssels hinzugefügt werden.

Ein Überblick über das Schema wird in Abbildung 4.5 gegeben.

Je nach Organisationsstruktur bietet es sich an, die Methoden von Role-Based-Access-Control zu nutzen und für jede Rolle ein eigenes Attribut zu erstellen. Benutzer, die mehrere Rollen übernehmen, bekommen für jede Rolle das entsprechende Attribut zugewiesen und können alle Einträge entschlüsseln, die zu ihrer Rolle gehören. Attribute könnten sich auch orientieren an

- der Organisationsstruktur
- dem Dienst, der vom System bereitgestellt wird
- der Identität des Systems
- dem Ort, an dem das System aufgestellt ist
- dem Ersteller der Daten

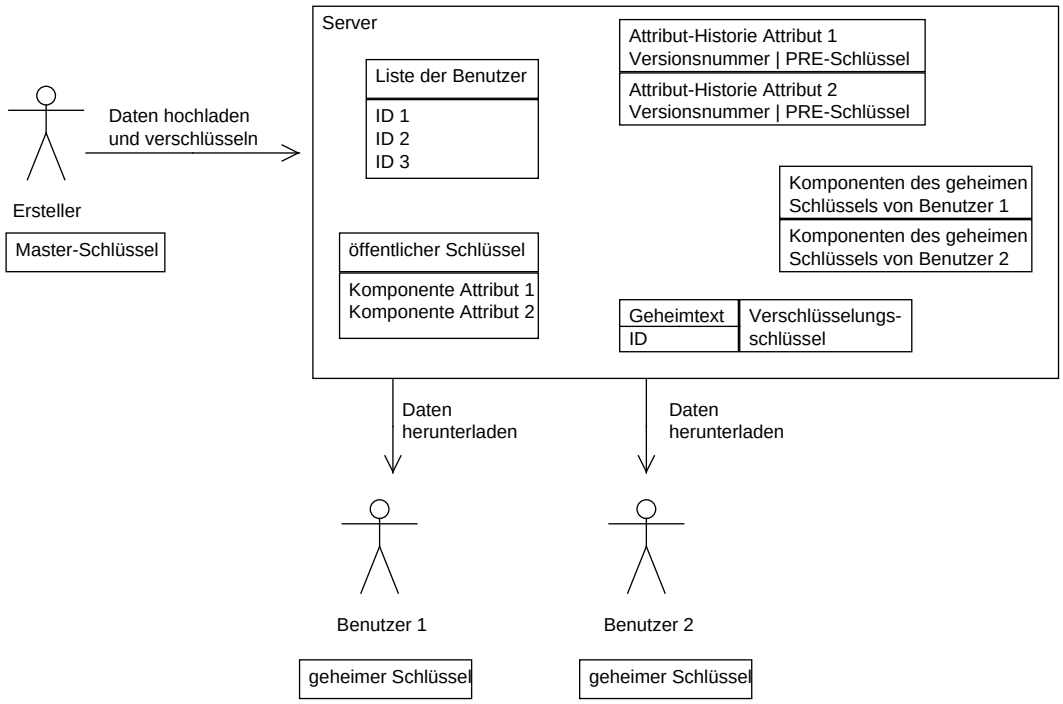

Abbildung 4.5: Darstellung des Modells. Auf dem Server liegen eine Liste der Benutzer, die Attribut-Historien der verschiedenen Attribute, der öffentliche Schlüssel, die Komponenten der geheimen Schlüssel der Benutzer sowie die Geheimtexte mit den entsprechenden Verschlüsselungsschlüsseln vor. Der Ersteller lädt verschlüsselte Daten hoch, die Benutzer können die Daten herunterladen.

Initialisierung des Systems

Der Ersteller der Daten erzeugt den Sicherheitsparameter p, den öffentlichen Schlüssel des Systems PK und den Master-Schlüssel des Systems MK. Alle Komponenten des öffentlichen Schlüssels PK werden vom Ersteller signiert und anschließend an den Server gesendet.

Erstellen und Löschen eines Datensatzes

Der neu erzeugte Eintrag wird mit einer ID versehen, die systemweit einzigartig ist. Zur Verschlüsselung wird ein symmetrischer Algorithmus benutzt, mit dem die Daten verschlüsselt werden. Der Verschlüsselungsschlüssel DEK wird ebenfalls verschlüsselt. Dazu wird ein Set an Attributen definiert und mit KP-ABE der Verschlüsselungsschlüssel verschlüsselt. Die Daten und der zugehörige Verschlüsselungsschlüssel werden anschließend auf den Server hochgeladen, wo Benutzer sie dann herunterladen können.

Das Löschen eines Datensatzes kann nur vom Ersteller durchgeführt werden. Dazu schickt

dieser die ID der Datei signiert an den Server. Kann die Signatur vom Server verifiziert werden, wird der Eintrag gelöscht.

Um eine zusätzliche Sicherheit von Passwort-Einträgen auf einer hohen Vertraulichkeitsstufe zu gewährleisten, kann der Verschlüsselungsschlüssel für diese Einträge nach Secret Sharing in Teilschlüssel aufgeteilt werden, sodass kein Benutzer allein Zugriff auf den Eintrag bekommt, sondern mit anderen Benutzern kooperieren muss, um den Schlüssel wieder zusammenzusetzen.

Generierung und Verteilung der Schlüssel

Im System existieren mehrere Schlüssel. Jeder Benutzer besitzt ein Schlüsselpaar aus öffentlichem und privaten Schlüssel. Außerdem gibt es den öffentlichen Schlüssel des Systems sowie den Master-Schlüssel, der vom Ersteller geheim gehalten wird.

Um einen geheimen Schlüssel für einen Benutzer zu erstellen, wird aus einer entsprechenden Zugriffsstruktur, dem Master-Schlüssel und dem öffentlichen Schlüssel PK ein Schlüssel generiert. Diese Schlüssel müssen den Benutzern übermittelt werden, was über den Server erfolgt (siehe 4.4.1).

Der Verschlüsselungsschlüssel DEK muss verschlüsselt mit den Daten auf dem Server vorliegen, da zur Entschlüsselung sowohl der geheime Schlüssel eines Nutzers als auch der DEK gebraucht werden. Der öffentliche Schlüssel PK muss offen vorliegen, ebenso müssen die öffentlichen Schlüssel der Benutzer frei einsehbar sein.

Sowohl Verschlüsselungsschlüssel als auch geheimer Schlüssel können sich beim Entzug von Zugriffsberechtigungen für einen Benutzer ändern. In diesem Fall werden sie vom Server erneuert.

Erstellen neuer Benutzer im System

Neue Benutzer müssen zunächst eine neue ID, Zugriffsstruktur und einen geheimen Schlüssel erhalten, bevor sie Daten entschlüsseln können. Der Ersteller muss eine ID wählen, dem Benutzer eine Zugriffsstruktur P zuweisen und dann den geheimen Schlüssel erstellen. Zugriffsstruktur, geheimer Schlüssel und öffentlicher Schlüssel des Systems müssen mit dem öffentlichen Schlüssels des neuen Benutzers verschlüsselt und vom Ersteller signiert werden (genannt C). Das wird zusammen mit der ID des Benutzers, der Signatur und den Komponenten des geheimen Schlüssels ohne Dummy-Attribut D an den Server geschickt.

Der Server überprüft dann die Signatur und erstellt bei Korrektheit einen neuen Eintrag für den Benutzer in der Benutzerliste. C wird anschließend an den neuen Benutzer übermittelt, der die Daten dann mit seinem geheimen Schlüssel entschlüsselt. Auch er kann die Signatur verifizieren, um den Ursprung der Daten zu überprüfen. Nach Entschlüsselung hat er seine Zugriffsstruktur, seinen geheimen Schlüssel und den öffentlichen Schlüssel des Systems erhalten. Mit diesem geheimen Schlüssel können nun die Daten auf dem Server entschlüsselt werden.

Der Server speichert dabei alle Komponenten des geheimen Schlüssels, abgesehen von dem Dummy-Attribut D. Er muss diese Daten speichern, um später den Entzug von Zugriffsberechtigungen durchzuführen. Da ihm aber das Dummy-Attribut nicht bekannt ist, kann er nicht mithilfe des geheimen Schlüssels im System gespeicherten Daten entschlüsseln.

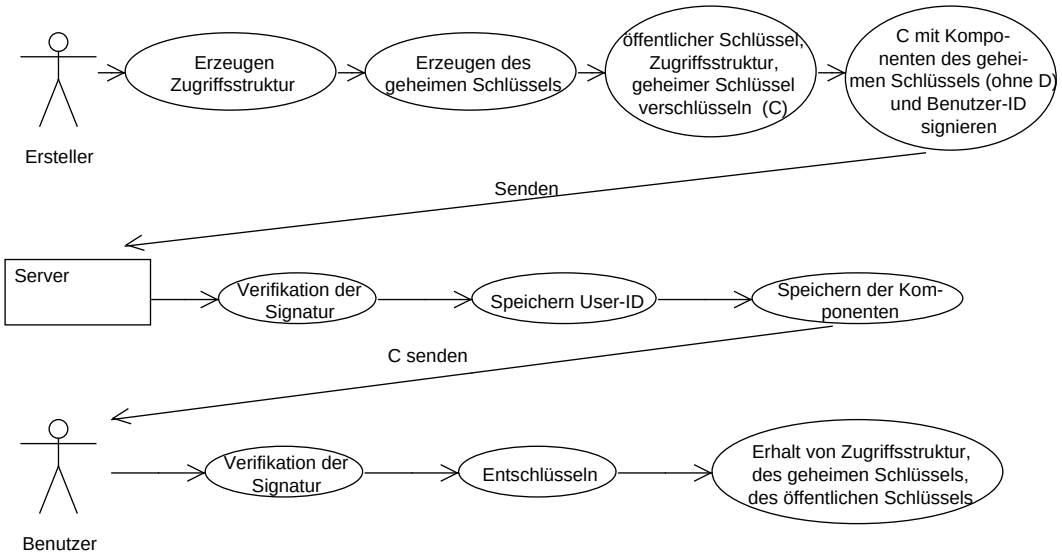

Abbildung 4.6: Beispiel: Anlegen eines neuen Benutzers.

Eine Auflistung dieses Vorgangs findet sich in Abbildung 4.6.

Entzug von Zugriffsberechtigungen

Das Entziehen von Zugriffsberechtigungen gestaltet sich schwierig, da im Normalfall mehrere Benutzer Zugriff auf ein Attribut haben. Das bedeutet, dass der Ersteller alle betroffenen Attribute für alle Benutzer aktualisieren muss. Eine Möglichkeit, diesen Vorgang durchzuführen, wäre, die geheimen Schlüssel mit einem Datum-Attribut zu versehen, an dem sie auslaufen. Das bedeutet allerdings, dass die Nutzer bei Ablauf des Schlüssels immer wieder neue Schlüssel vom Ersteller bekommen müssen. Außerdem ist es nicht möglich, einem Benutzer Berechtigungen von einem Moment auf den anderen zu entziehen, sondern nur zu einem vorher festgesetzten Zeitpunkt.

Alle Operationen, die zum Entzug von Zugriffsberechtigungen notwendig sind, müssten vom Ersteller durchgeführt werden, wenn kein PRE verwendet wird. Dazu sind folgende Schritte erforderlich:

- Die zu aktualisierenden Attribute müssen bestimmt werden.
- Die zugehörigen Komponenten von MK und PK müssen aktualisiert werden.

- Die geheimen Schlüssel aller Benutzer müssen aktualisiert werden.
- Die betroffenen DEKs müssen aktualisiert und neu verschlüsselt werden.

Um diesen Aufwand für den Ersteller möglichst gering zu halten, können diese Operationen mithilfe von PRE vom Server übernommen werden. Auch in dem Fall muss der Ersteller festlegen, welche Attribute und zugehörige Komponenten von MK und PK aktualisiert werden müssen. Dafür werden PRE-Schlüssel generiert, die zusammen mit der ID des betroffenen Nutzers, den gewählten Attributen und den aktualisierten Komponenten signiert an den Server geschickt werden. Die ID des Benutzers wird dann aus der Benutzerliste gelöscht. Die aktualisierten Komponenten des öffentlichen Schlüssels müssen gespeichert werden, ebenso der PRE-Schlüssel in der Historie des betroffenen Attributs. Die Aktualisierung der betroffenen geheimen Schlüssel der Benutzer und der Geheimtexte kann nun vom Server übernommen werden.

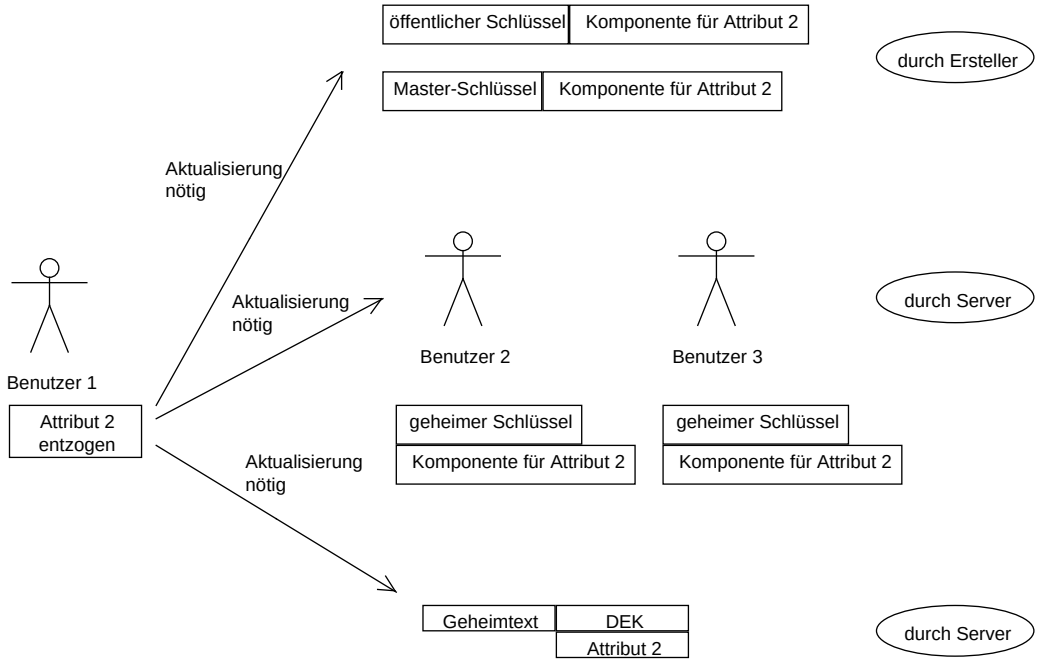

Abbildung 4.7: Eine beispielhafte Darstellung der notwendigen Schritte für den Entzug von Zugriffsberechtigungen.

Es ist auch möglich, die PRE-Schlüssel auf dem Server zu speichern. Dazu könnte ein PRE-Schlüssel gleich bei der Verschlüsselung der Daten erstellt und mit den Daten hochgeladen werden. Das ist allerdings keine gute Alternative, da in diesem Fall bei Kompromittierung des Servers die PRE-Schlüssel für Angreifer offen liegen und ein Angreifer alle Daten neu verschlüsseln könnte.

Die notwendigen Aktualisierungen für den Entzug von Berechtigungen werden in Abbildung 4.7 dargestellt.

Zugriff auf die Daten

Wenn ein Benutzer Daten anfordert, überprüft der Server zunächst, ob der Benutzer in der Liste der validen Nutzer enthalten ist. Ist das der Fall, bekommt der Benutzer die Daten und den Verschlüsselungsschlüssel übertragen. Wenn sich zwischenzeitlich Berechtigungen geändert haben, was anhand der Versionsnummern der Attribute erkannt werden kann, werden die Komponenten des geheimen Schlüssels des Benutzers vor dem Abruf der Daten auf die neue Version aktualisiert und der Verschlüsselungsschlüssel neu verschlüsselt.

Nach Aktualisierung der Komponenten (falls notwendig) kann der Benutzer nun den Verschlüsselungsschlüssel entschlüsseln. Mithilfe dieses Schlüssels kann dann der Datensatz entschlüsselt werden und der Benutzer erhält das Passwort.

Notfall-Zugriff

Um eine temporäre Freigabe möglich zu machen, kann der Schlüssel zusätzlich mit Notfall-Attributen verschlüsselt werden. Wenn ein Notfall vorliegt, kann ein Schlüssel mit entsprechender Zugriffsstruktur von einer zuständigen Stelle nach der Authentisierung des Benutzers und der Verifizierung des Notfalls abgerufen werden. Dazu muss eine zentrale Stelle eingerichtet werden, bei der die Schlüssel für Notfall-Zugriff gespeichert werden. Bei der Erstellung eines Eintrags wird dabei ein zusätzlicher Notfall-Schlüssel erzeugt, der an diese zentralen Stelle gesendet wird. Nach Nutzung des Schlüssels können die Daten dann vom Ersteller neu verschlüsselt und dabei neue Notfall-Attribute verwendet werden, um die temporäre Freigabe rückgängig zu machen.[LYZ+10]

Eine solche zentrale Stelle müsste allerdings vertrauenswürdig und zudem immer verfügbar sein, um den Schlüssel freigeben zu können. Außerdem wäre ein solcher zentraler Punkt ein beliebtes Ziel für Angreifer, da dort alle Schlüssel für sämtliche Einträge gespeichert werden würden.

Eine weitere Möglichkeit wäre es, den Notfall-Schlüssel auf mehrere zuständige Personen aufzuteilen, sodass jede Person einen Teil des Schlüssels erhält. Der Mitarbeiter, der einen Notfall-Zugriff benötigt, muss dann von verschiedenen Personen die Teilgeheimnisse bekommen, um den Schlüssel zusammenzusetzen und den Passwort-Eintrag entschlüsseln zu können. Auch hier müsste ein Schlüssel zusätzlich mit einem Notfall-Attribut versehen werden.

Der Notfall-Schlüssel, der zur Entschlüsselung benötigt wird, liegt dann symmetrisch verschlüsselt auf dem Server. Der Verschlüsselungsschlüssel wird mit Secret Sharing in Teilgeheimnisse aufgeteilt, die mit KP-ABE verschlüsselt werden. Dabei wird darauf geachtet, dass jeweils eine zuständige Person auf einen Teilschlüssel zugreifen kann. Benötigt ein Benutzer Zugriff im Notfall, setzt er sich mit den zuständigen Personen in Verbindung, die ihm ihre Teilgeheimnisse übermitteln können. Der Benutzer kann anschließend, sobald genügend Zustimmungen in Form von Teilgeheimnissen vorliegen, den Schlüssel zusammensetzen. Mit diesem Schlüssel kann er den Notfall-Schlüssel entschlüsseln. Mit dem Notfall-Schlüssel be-

kommt er Zugriff auf den Verschlüsselungsschlüssel DEK und kann damit die Daten entschlüsseln.

In diesem Fall ist eine Kooperation der Benutzer natürlich nicht mehr ausgeschlossen, sondern erwünscht.

Logging

Logging kann vom Server durchgeführt werden. Dabei übernimmt der Server die Aufgabe, automatisch bei jedem Hoch- beziehungsweise Herunterladen von Daten ein Log mit der ID des Benutzers, der ID des Datensatzes und einen Zeitstempel anzulegen. Es muss allerdings darauf vertraut werden, dass der Server diese Aufgabe übernimmt und ausführt.

Zugriff auf das Server-Log haben die Administratoren des Servers. Die Log-Daten können von ihnen ebenfalls verschlüsselt auf den Server hochgeladen werden, um Personen mit Audit-Berechtigung Zugriff zu gewähren.

4.4.2 Analyse anhand des Kriterienkatalogs

Die aufgestellte theoretische Lösung wird im Folgenden anhand der Kriterien von Kapitel 3.3 geprüft. Dabei wird gezeigt, welche Anforderungen an ein gutes Passwort-Management erfüllt und welche nicht eingehalten werden können. $\surd$ kennzeichnet dabei erfüllte, $\times$ nicht erfüllte und $\diamondsuit$ teilweise erfüllte Anforderungen. Ein tabellarischer Gesamtüberblick findet sich in 5.1.

Aufbau des Systems

Anforderung: zentrale Speicherung der Passwörter $\surd$

Alle Passwörter können zentral auf einem Server gespeichert werden. Es ist aber auch problemlos möglich, mehrere Server zur Speicherung der Daten zu verwenden.

Anforderung: Automatische Änderung der Passwörter $\times$

Um dieses Kriterium zu erfüllen, müsste das System in der Lage sein, automatisch auf die Passwörter zuzugreifen und diese zu ändern. Da aber gerade der Zugriff des Systems auf die Passwörter verhindert werden soll, ist diese Anforderung nicht zu erfüllen.

Anforderung: Erfüllung verschiedener Sicherheitskriterien $\surd$

Die Passwörter liegen nur verschlüsselt auf dem Server vor, sind also sicher gespeichert. Im Besonderen kann der Server selber nicht auf die Daten zugreifen. Die Kommunikation zwischen Benutzer und Server kann selbstverständlich auch über dafür geeignete Protokolle, beispielsweise SSL, erfolgen. Eine Zugriffskontrolle ist durch die Verschlüsselung der Daten gewährleistet.

Anforderung: Mehrbenutzerbetrieb √

Die Verwaltung mehrerer Benutzer ist kein Problem. Grundsätzlich können beliebig viele Benutzer angelegt werden, ihnen muss nur ein geheimer Schlüssel entsprechend ihrer Berechtigungen zugewiesen werden. Dieser geheime Schlüssel reicht aus, um die Zugriffskontrolle zu gewährleisten, sodass nur berechtigte Benutzer Zugriff auf die Ressourcen haben.

Ebenso ist es möglich, den Zugriff der Benutzer wieder beliebig einzuschränken und Benutzer zu löschen. Im Besonderen ist es mehreren Benutzern nicht möglich, miteinander zu kooperieren, um Zugriff auf geschützte Daten zu erhalten, für die ihnen allein die Berechtigung fehlt (mit der Ausnahme vom Notfall-Zugriff).

Wünschenswert: Sicherung der Daten auch bei Ausfall des Servers √

Es können beliebig viele Backups der Daten angelegt werden. Beim Ausfall eines Servers kann dann mit einem Backup dieser Daten weitergearbeitet werden.

Wünschenswert: Log mit alten Passwörtern vorhanden ♢

Für ein solches Log müssten alte Passwörter vom Server gespeichert werden. Das ist natürlich nicht erwünscht. Es ist allerdings problemlos möglich, alte Kopien geänderter Daten zu behalten.

Dank PRE werden die Verschlüsselungsschlüssel automatisch vom Server im Fall eines Entzugs von Zugriffsberechtigungen aktualisiert, sodass auch für alte Kopien die aktuellen Zugriffsberechtigungen gelten.

Berechtigungsverwaltung und Zugriffskontrolle

Anforderung: Zugriffskontrolle √

Die Zugriffskontrolle wird durch die Verschlüsselung der Daten gewährleistet. Die Einträge sind nicht offen einsehbar, sondern können nur von berechtigten Personen entschlüsselt werden.

Anforderung: Anlegen von Gruppen oder Rollen ist möglich. ♢

Gruppen können dadurch definiert werden, dass alle Mitglieder einer Gruppe den gleichen geheimen Schlüssel erhalten. Rollen können bestimmten Attributen entsprechen, die einem Benutzer zugeteilt werden. Trotzdem muss jedem Benutzer einzeln die benötigte Zugriffsstruktur zugewiesen werden.

Wünschenswert: Übernahme der Benutzer und Gruppen aus Verzeichnisdiensten ×

Die Übernahme ist nicht möglich, da die Benutzer eigens vom Ersteller angelegt werden müssen.

Erstellen der Passwörter

Anforderung: Vorgabe von Passwort-Richtlinien bei der Erstellung von Passwörtern durch den Benutzer ×

Es ist zwar möglich, dem Ersteller von Passwörtern abstrakt Richtlinien vorzugeben, aber ihre Einhaltung kann nicht kontrolliert werden. Ebenso ist es nicht möglich, die Wahl eines alten oder eines einem alten sehr ähnlichen Passworts zu verhindern. Dafür müsste ein Log existieren, in dem alte Passwörter gespeichert werden.

Anforderung: Vorgabe von Passwort-Richtlinien und Generierung geeigneter Passwörter bei Nutzung eines Generators ×

Die Passwörter werden durch den Ersteller angelegt, es wird kein Generator verwendet. Natürlich kann ein Generator zur Erstellung empfohlen werden, es ist aber nicht möglich, die Passwörter zu überprüfen.

Anforderung: Kontrolle des Passwort-Zugriffs √

Die Zugriffskontrolle auf die Passwörter ist durch Verschlüsselung gewährleistet. Es ist ausgeschlossen, dass ein Benutzer ein Passwort einsehen kann, für das er keine Berechtigung hat, noch nicht einmal durch Kooperation mit anderen Benutzern oder mit dem Server.

Anforderung: Vergabe von Zugriffsberechtigungen auf Passwörtern √

Die Benutzer können sowohl Lese- als auch Schreibberechtigungen haben. Die Zugriffsberechtigungen werden durch den Ersteller vorgegeben, der die geheimen Schlüssel der Benutzer generiert und den Verschlüsselungsschlüssel entsprechend mit Attributen verschlüsselt. Der Ersteller hat also die volle Kontrolle darüber, welcher Benutzer welche Daten einsehen kann.

Anforderung: Verbergen von Einträgen ♢

Es ist nicht möglich, Einträge auf einem Server vor den Benutzern zu verbergen. Aber es besteht die Möglichkeit, solche Einträge auf einen anderen Server auszulagern, um sie so geheim zu halten.

Wünschenswert: Vergabe von Passwörtern durch das System ×

Die Erstellung von Passwörtern durch das System ist nicht möglich und auch nicht erwünscht, da auch in diesem Fall das System wieder Zugriff auf die verschlüsselten Daten haben müsste.

Autorisierung einer temporären Zugriffsberechtigung

Anforderung: Notfall-Zugriff ist möglich. ♢

Der Notfall-Zugriff ist bei einem solchen Modell nur schwer umzusetzen. Eine Möglichkeit ist es, den Verschlüsselungsschlüssel zusätzlich mit einem Notfalls-Attribut zu verschlüsseln. Im Notfall kann der dem Attribut entsprechende Notfall-Schlüssel zur Entschlüsselung genutzt werden. Diese Möglichkeit hat aber wieder andere Nachteile.

Anforderung: Über eine Freigabe kann von mehreren Personen entschieden werden. ♢

Das hängt davon ab, wie der Prozess gestaltet wird, um Zugriff auf den Notfall-Schlüssel zu bekommen.

Anforderung: Zeitliche Beschränkung der Freigabe ×

Es ist möglich, dem Benutzer die gewährten Zugriffsberechtigungen wieder zu entziehen, dazu muss der Ersteller den Schlüssel wieder mit neuen Notfall-Attributen verschlüsseln. Damit wird der alte Notfall-Schlüssel ungültig und der Benutzer hat keinen Zugriff mehr.

Es ist allerdings nicht möglich, den Zugriff auf ein bestimmtes Zeitintervall zu beschränken, der Zugriff muss vom Ersteller selbst wieder entzogen werden.

Anforderung: Automatisches Ändern des Passworts ×

Eine automatische Änderung des Passworts durch das System ist nicht möglich.

Anforderung: Loggen des Notfall-Zugriffs ♢

Das Herunterladen von Verschlüsselungsschlüssel und verschlüsselten Daten wird vom Server geloggt. Da aber nicht aufgezeichnet wird, wer Berechtigung erteilt hat und wer nicht, werden nicht alle notwendigen Informationen gespeichert.

Freigabe der Passwörter

Anforderung: Zugriff auf einen Account ist mehreren Personen möglich. √

Es können beliebig viele Personen auf einen Account zugreifen. Dafür muss nur das erforderliche Passwort entschlüsselt werden.

Wünschenswert: Einloggen ohne Anzeigen des Passworts ist möglich. ×

Eine solche Möglichkeit ist nicht gegeben. Das Passwort kann allein durch die Entschlüsselung des entsprechenden Eintrags erhalten werden.

Audit

Anforderung: Existenz eines Audit-Logs ♢

Ein solches Log kann durch den Server angelegt werden. Dieser kann Informationen darüber loggen, wann und von wem ein verschlüsselter Datensatz hochgeladen wurde und wer wann welche Daten heruntergeladen hat.

Weitere Informationen können nicht gesammelt werden.

Anforderung: Das Log kann nicht von Administratoren geändert werden. ×

Das kann nicht garantiert werden.

Anforderung: Der Zugriff auf die Logs ist nur eingeschränkt möglich. √

Auf Server-Logs können nur die Administratoren des Servers zugreifen. Die Logs können von ihnen auch verschlüsselt auf den Server hochgeladen werden, sodass nur berechtigte Personen darauf Zugriff erhalten.

Wünschenswert: Automatisches Versenden von Benachrichtigungen ×

Eine automatische Benachrichtigung über wichtige Ereignisse ist nicht möglich.

Zusammenfassung der Analyse

Ein guter Teil der in Kapitel 3.3 aufgestellten Anforderungen an das Passwort-Management wird von diesem Modell ganz oder teilweise erfüllt. Die Umsetzung eines Teils der genannten Kriterien gestaltet sich aber schwierig, da auf einen vertrauenswürdigen Dritten verzichtet wird. Folgende Anforderungen werden nicht eingehalten:

- Es ist nicht möglich, Passwörter automatisch durch das System generieren oder ändern zu lassen. Im Besonderen ist es nach einem Notfall-Zugriff nicht möglich, das verwendete Passwort sofort automatisch zu ändern.
- Es können keine Richtlinien zur Passwort-Vergabe erstellt und kontrolliert werden. Deswegen kann nicht verhindert werden, dass ein schwaches Passwort gewählt wird. Ebenfalls ist es möglich, dass ein altes Passwort oder ein ähnliches Passwort erneut gewählt wird.
- Das Verbergen von Einträgen kann nur durch die Auslagerung der Daten auf einen anderen Server gewährleistet werden.
- Der Notfall-Zugriff ist nur schwer zu modellieren. Nach einem Notfall-Zugriff muss der Ersteller selbst die Berechtigung wieder entziehen, es gibt kein festgesetztes Zeitlimit, nach dem die Berechtigung automatisch wieder entzogen wird.
- Es werden nicht alle benötigten Informationen geloggt.

Ein weiterer Nachteil ist, dass die Verwaltung der Schlüssel und der Daten allein von den Benutzern beziehungsweise den Erstellern der Daten übernommen werden muss. Dieses Problem ergibt sich aber bei allen Lösungen, bei denen kein vertrauenswürdiger Dritter das entsprechende Management übernimmt. Der Benutzer ist dafür zuständig, den eigenen Schlüssel sicher aufzubewahren, wofür sich eine Verschlüsselung dieses Schlüssels empfiehlt. Der Ersteller ist für die Erstellung und Verschlüsselung von Passwörtern zuständig, zudem muss er die geheimen Schlüssel der Benutzer generieren. Außerdem muss er den Prozess des Entzuges von Zugriffsberechtigungen verwalten. Der Verwaltungsaufwand für die Beteiligten im System ist also höher als bei einer Verwaltung der Passwort-Einträge durch einen vertrauenswürdigen Dritten.

Zudem ist das System nicht gefeit dagegen, dass ein Benutzer seinen Schlüssel an einen Angreifer weitergibt. In diesem Fall hat der Angreifer Zugriff auf alle Daten, die mit diesem

Schlüssel entschlüsselt werden können. Dieses Problem tritt aber auch auf, wenn die Zugriffskontrolle von einem vertrauenswürdigen Dritten übernommen wird. In dem Fall hat ein Angreifer, der einen Benutzeraccount kompromittiert, auch Zugriff auf alle Passwörter, für die der Benutzer Berechtigung hat.

Ein Modell, das einen vertrauenswürdigen Dritten zur Zugriffskontrolle einsetzt, kann hingegen alle Kriterien (je nach Implementierung) erfüllen. Allerdings ergeben sich dabei einige Sicherheitsprobleme. Ein vertrauenswürdiger Dritter stellt selbst eine potentielle Schwachstelle in einem System dar. Es muss darauf vertraut werden, dass der vertrauenswürdige Dritte die übernommenen Aufgaben korrekt ausführt und kein Fehlverhalten auftritt.

Eine Anwendung kann aber Bugs aufweisen, die zu falschem Verhalten führen, sodass beispielsweise Berechtigungen falsch vergeben werden. Außerdem können Sicherheitslücken auftreten, die von einem Angreifer ausgenutzt werden können. Wird die Sicherheit des vertrauenswürdigen Dritten gefährdet, ist aber die Sicherheit des gesamten Systems in Gefahr. Die Benutzer haben keine andere Wahl, als auf das korrekte Funktionieren des vertrauenswürdigen Dritten zu vertrauen.

Wird die Anwendung, die in einem System der Passwortverwaltung die Rolle des vertrauenswürdigen Dritten einnimmt, von einem Angreifer kompromittiert, ist auch die Sicherheit aller Passwörter gefährdet. Liegen beispielsweise die Verschlüsselungsschlüssel bei der Anwendung vor und kann sich ein Angreifer Zugriff auf die Anwendung verschaffen, kann er alle Verschlüsselungsschlüssel zur Entschlüsselung der Daten nutzen.

Das ABE-Schema hingegen bietet einen großen Gewinn an Sicherheit. Selbst im Falle der Kompromittierung des Servers bleibt die Vertraulichkeit der Daten gewahrt, ein Angreifer kann die Passwörter nicht entschlüsseln oder an die Verschlüsselungsschlüssel gelangen, die ebenfalls nur verschlüsselt vorliegen. Die Daten können damit sogar auf nicht vertrauenswürdigen Servern sicher gespeichert werden.

Auch ist es mehreren Personen nicht möglich, zu kooperieren und Zugriff auf Daten zu erhalten, den sie allein nicht hätten. Das schützt die Daten also auch vor unerwünschter Zusammenarbeit durch interne Angreifer. Selbst wenn Server und interne Benutzer miteinander kooperieren, können sie keinen unberechtigten Zugriff erlangen.

Ein Modell mit vertrauenswürdigem Dritten weist somit einige Sicherheitslücken auf. ABE hingegen bietet also große Sicherheit sowohl gegenüber externen Angreifern als auch gegenüber internem Missbrauch.

5 Analyse am Markt vorhandener Passwort-Tools

Im folgenden Kapitel werden zwei kommerzielle Programme analysiert, die eine effiziente und effektive Verwaltung von Passwörtern im Unternehmensbereich zum Ziel haben. Für diese Bewertung wird der in Kapitel 3.3 aufgestellte Katalog an Kriterien und Anforderungen verwendet. Warum sich Passwort-Manager für die private Nutzung nur schlecht für den Einsatz in Unternehmen eignen, wurde in 2.2.1 erläutert.

Die zwei ausgewählten Programme sind die Enterprise Password Safe Edition mit dem Enterprise Server von MATESO ([MAT]) und die Password Manager Pro Premium Edition von ManageEngine ([Man]). Die Enterprise Edition baut dabei auf ein Programm zur Passwort-Verwaltung auf, das auch für die private Nutzung verwendet werden kann, wohingegen der Password Manager Pro auf die Verwaltung von Passwörtern privilegierter Accounts spezialisiert ist.

Die Enterprise Password Safe Edition wurde gewählt, da dieses Programm bereits in kleinerem Rahmen am LRZ eingesetzt wird und demzufolge ein Kandidat für eine Einführung in größerem Umfang ist. Der Password Manager Pro eignet sich für die Bewertung aufgrund der Spezialisierung auf die Verwaltung privilegierter Accounts sowie aufgrund der vorhandenen ausführlichen Dokumentation und Materialien.

Für die Analyse wurden die Programme installiert und getestet sowie im Fall von Password Manager Pro die vorhandene Online-Demoversion des Tools genutzt. Mithilfe des Kriterienkatalogs werden die Anforderungen an das Management von Passwörtern untersucht und die Bewertung erfolgt auf Basis der eingehalteten beziehungsweise der nicht eingehaltenen Kriterien. Dabei kennzeichnet auch hier wieder $\surd$ eine erfüllte, $\times$ eine nicht erfüllte und $\diamondsuit$ eine in Ansätzen erfüllte Anforderung.

5.1 Bewertung von Enterprise Password Safe Edition mit Enterprise Server

PasswordSafe wird in verschiedenen Editionen angeboten, sowohl für die private Nutzung als auch für den Einsatz in Unternehmen. Vier Editionen stehen zur Verfügung, wobei jede Edition neue zusätzliche Features beinhaltet.

Für Unternehmen steht sowohl die Professional Edition als auch als die Enterprise Edition zur Verfügung. Die Enterprise Edition verfügt über mehr Funktionen als die Professional

Edition. Zusätzlich gibt es noch den Enterprise Server, der in Verbindung mit der Enterprise Edition genutzt werden kann. Die Enterprise Edition kann bis zu 20 Benutzer unterstützen, danach ist aus Performanz-Gründen der Einsatz des Enterprise Servers nötig. Die Analyse konzentriert sich also auf den Einsatz der Enterprise Edition in Verbindung mit dem Enterprise Server. Enterprise Edition und Enterprise Server machen auch eine Client-/Server-Architektur möglich. Die Enterprise Edition kann zusätzlich mit kostenpflichtigen Modulen und Plugins erweitert werden.

Da die Enterprise Edition auch die Features der anderen Editionen enthält, werden einige Funktionen angeboten, die für den Einsatz in Unternehmen und zur Verwaltung privilegierter Accounts nicht notwendig sind. Zum Beispiel erlaubt es die Enterprise Edition, TAN-Listen zu speichern oder automatisch Passwörter auf Webseiten einzutragen. Solche Features sind eher für den privaten Gebrauch von Nutzen und werden in der folgenden Evaluation nicht berücksichtigt.

5.1.1 Analyse

Aufbau des Systems

Anforderung: zentrale Speicherung der Passwörter √

Passwörter werden zentral in einer Datenbank gespeichert, die von dem Enterprise Server bereitgestellt wird.

Anforderung: Automatische Änderung der Passwörter ×

Die Passwörter können nicht automatisch geändert werden. Es kann bei der Erstellung eines Eintrags angegeben werden, wie lange ein Passwort gültig sein soll. Läuft das Kennwort ab, kann eine automatische Benachrichtigung verschickt werden.

Anforderung: Erfüllung verschiedener Sicherheitskriterien √

Die Passwörter und Kommunikation werden mit AES-256 gesichert. Zum Schlüsselaustausch wird RSA-1024 verwendet.

Anforderung: Mehrbenutzerbetrieb √

Es können mehrere Benutzer angelegt werden und es gibt die Möglichkeit, den Zugriff der Benutzer zu beschränken.

Wünschenswert: Sicherung der Daten auch bei Ausfall des Servers √

Es ist möglich, Backups von den Daten anzulegen. Dazu kann ein genauer Zeitplan erstellt werden, sodass die Backups automatisch durchgeführt werden. Ebenfalls wird die Möglichkeit einer Synchronisation zwischen Datenbanken auf zwei Servern angeboten.

Der angebotene Offline-Modus erlaubt es zudem, lokal eine verschlüsselte Datenbank mit den Passwörtern zu speichern. Das macht auch dann den Zugriff auf die Passwörter möglich, falls keine Verbindung zum Server besteht.

Wünschenswert: Log mit alten Passwörtern vorhanden √

Die Enterprise Edition bietet eine Historie, die alle Änderungen an einem Eintrag aufzeichnet. Damit ist es auch möglich, eine alte Version des Eintrags wiederherzustellen. Allgemein kann die Historie alte und neue Versionen von Daten anzeigen und Änderungen darstellen.

Berechtigungsverwaltung und Zugriffskontrolle

Anforderung: Zugriffskontrolle √

Die Authentifizierung erfolgt über die Eingabe des Passworts. Es ist auch eine Two-Factor-Authentication mittels Zertifikaten, Smartcards oder Token möglich, wenn eine entsprechende Public Key Infrastructure im Unternehmen vorhanden ist und das entsprechende zusätzliche kostenpflichtige PKI-Modul erworben wurde.

Anforderung: Anlegen von Gruppen oder Rollen ist möglich. √

Die Erstellung von Gruppen ist möglich. Berechtigungen können auch einer ganzen Gruppe übertragen werden.

Wünschenswert: Übernahme der Benutzer und Gruppen aus Verzeichnisdiensten √

Sowohl Active Directory als auch LDAP können integriert und Benutzer sowie Gruppen aus diesen Verzeichnisdiensten übernommen werden.

Erstellen der Passwörter

Anforderung: Vorgabe von Passwort-Richtlinien bei der Erstellung von Passwörtern durch den Benutzer ♢

Es ist möglich, Richtlinien vorzugeben und nur das Anlegen von Passwörtern zu erlauben, die den Richtlinien entsprechen. Allerdings können bereits gewählte oder ähnliche Passwörter erneut verwendet werden.

Anforderung: Vorgabe von Passwort-Richtlinien und Generierung geeigneter Passwörter bei Nutzung eines Generators √

Passwörter können automatisch durch das System erzeugt werden. Für die Generierung können Richtlinien vorgegeben werden. Außerdem wird ein Passwort-Generator bereitgestellt, der mithilfe von Mausbewegungen auf einer grafischen Oberfläche ein Passwort erzeugen kann.

Anforderung: Kontrolle des Passwort-Zugriffs √

Nur bei entsprechender Berechtigung kann ein Passwort eingesehen oder geändert werden.

Anforderung: Vergabe von Zugriffsberechtigungen auf Passwörtern √

Es können verschiedene Berechtigungen auf die Passwörter vergeben werden. Lese- und Schreibberechtigung oder Berechtigung zum Löschen eines Datensatzes können erteilt werden.

Anforderung: Verbergen von Einträgen √

Benutzer können nur die Einträge sehen, für die sie die entsprechende Berechtigung haben.

Wünschenswert: Vergabe von Passwörtern durch das System √

Passwörter können automatisch durch einen Generator erstellt werden.

Autorisierung einer temporären Zugriffsberechtigung

Anforderung: Notfall-Zugriff ist möglich. √

Für den Notfall-Zugriff existiert ein Siegelsystem. Ein bestimmter Passwort-Eintrag kann mit einem Siegel versehen werden. Dieses Siegel kann nur von einem Benutzer gebrochen werden, wenn er einen Antrag stellt und dieser Antrag von einer oder mehreren berechtigten Personen positiv beschieden wird. Die Benachrichtigung der zuständigen Personen erfolgt intern über ein integriertes Nachrichtensystem.

Anforderung: Über eine Freigabe kann von mehreren Personen entschieden werden. √

Diese Möglichkeit ist gegeben.

Anforderung: Zeitliche Beschränkung der Freigabe ♢

Bei der Erstellung des Siegels muss angegeben werden, wie lange eine Freigabe gilt. Es ist nicht möglich, die Berechtigung vor Ablauf des Zeitlimits der temporären Freigabe wieder zu entziehen. Anschließend muss das Siegel neu angebracht werden, damit anderen Benutzern der Notfall-Zugriff möglich ist. Da das Passwort nicht automatisch geändert wird, kann es nach Ablaufen des Zeitlimits auch weiter verwendet werden.

Anforderung: Automatisches Ändern des Passworts ×

Das Passwort wird nicht automatisch geändert.

Anforderung: Loggen des Notfall-Zugriffs √

Wird ein Siegel gebrochen, werden zuständige Personen automatisch von dem System informiert. Alle Informationen darüber, wer die Anfrage gestellt hat, wer zugestimmt oder abgelehnt hat und ob eine Freigabe erteilt wurde, werden vom System aufgezeichnet.

Freigabe der Passwörter

Anforderung: Zugriff auf einen Account ist mehreren Personen möglich. ♢

Wenn ein Passwort-Eintrag versiegelt wird, damit nur im Notfall Zugriff darauf möglich ist, kann nur eine einzige Person dieses Siegel brechen und das Passwort erhalten.

Nur diese Person kann das Passwort sehen und nutzen. Anderen Benutzern abgesehen von den Administratoren ist es nicht möglich, Zugriff auf das Passwort zu bekommen.

Wünschenswert: Einloggen ohne Anzeigen des Passworts ist möglich. √

Das Starten von RDP-Sitzungen ist möglich. In diesem Fall muss das Passwort dem Benutzer nicht angezeigt werden. Es ist möglich, sich direkt mit dem entsprechendem System zu verbinden.

Ein bestimmter Passwort-Eintrag kann auch gesperrt werden. In dem Fall wird den berechtigten Benutzern das Passwort nicht angezeigt, es kann aber trotzdem zum automatischen Einloggen genutzt werden.

Audit

Anforderung: Existenz eines Audit-Logs √

Die Enterprise Edition bietet ein Logbuch an, das alle Aktionen, die an einem bestimmten Datensatz durchgeführt werden, aufzeichnet. Die Informationen können gefiltert werden, es ist möglich, sich die Aktionen eines bestimmten Benutzers oder spezielle Ereignisse anzeigen zu lassen. Insbesondere werden alle Aktionen, die mit dem Siegelsystem im Zusammenhang stehen, geloggt.

Zusätzlich können vordefinierte Reports angelegt werden, die beispielsweise die Anzahl der versiegelten Passwörter im System anzeigen können.

Anforderung: Das Log kann nicht von Administratoren geändert werden. √

Das Log wird automatisch von der Anwendung angelegt.

Anforderung: Der Zugriff auf die Logs ist nur eingeschränkt möglich. √

Es haben nur Benutzer Zugriff auf das Logbuch, denen die entsprechende Berechtigung erteilt wurde.

Wünschenswert: Automatisches Versenden von Benachrichtigungen ♢

Zuständige Personen werden über alle Aktionen, die mit dem Siegelsystem in Zusammenhang stehen, benachrichtigt. Ebenso ist es möglich, Reports automatisch zu versenden. Zu Benachrichtigungen bei Aktionen wie zum Beispiel das Ändern eines Passworts ist das Workflow-System erforderlich, das von einem zusätzlichen kostenpflichtigen Modul möglich gemacht wird.

5.1.2 Zusammenfassung der Bewertung

Ein großer Teil der aufgeführten Anforderungen wird von der Enterprise Edition erfüllt. So erlaubt es diese Edition unter anderem, Passwörter zentral zu speichern und einer beliebigen Anzahl an Benutzern Zugriff darauf zu geben, Passwort-Einträge zu verbergen oder nur im

Notfall nutzen zu können, mithilfe von Gruppen eine vereinfachte Verwaltung von Berechtigungen umzusetzen und ausführliche Audit-Logs anzulegen. Die Berechtigungen können sehr feingranular vergeben werden, sodass es möglich ist, einem Benutzer genau die und nur die Berechtigungen zuzuweisen, die er für die Ausführung seiner Aufgaben benötigt.

In dem System nimmt der Administrator eine zentrale Rolle ein. Er ist für das Anlegen von Benutzern und Gruppen sowie die Konfiguration der Datenbank und die Vergabe von Berechtigungen zuständig. Alle Passwörter und Log-Informationen können von ihm eingesehen werden und er besitzt alle im System möglichen Berechtigungen. Es ist möglich, die Datenbank so einzustellen, dass ein Administrator nicht mehr automatisch Siegel und Sperren entfernen kann. Da allerdings auch diese Konfiguration durch den Administrator selbst erfolgt, trägt das wenig dazu bei, die Berechtigungen des Administrators einzuschränken.

Entsprechende Berechtigungen zur Verwaltung der Benutzer und Gruppen und zum Konfigurieren der Datenbank können auch an andere Benutzer übertragen werden. Es können auch andere Benutzer in die Administratoren-Gruppen aufgenommen und ihnen alle Berechtigungen gegeben werden.

Da ein Administrator alle Berechtigungen ändern, alle Einstellungen an der Datenbank konfigurieren und alle Passwörter einsehen kann, sollte der Gebrauch dieses Accounts nur eingeschränkt möglich sein. Eine Möglichkeit wäre es, diesen Account nur in Notfällen zu verwenden.

Ein Teil der aufgeführten Kriterien wird von der Enterprise Edition allerdings nicht erfüllt. Im Besonderen:

- Das automatische Ändern von Passwörtern durch das System ist nicht möglich. Alle Passwörter müssen direkt vom Benutzer in der Datenbank und am Zielsystem geändert werden. Das bedeutet einen höheren Arbeitsaufwand für die Benutzer.
- Es ist problemlos möglich, bereits verwendete oder ähnliche Passwörter zu wählen.
- Wichtige Benachrichtigungen, beispielsweise über die Anforderung einer Freigabe, werden nur intern über die Anwendung versandt. Es ist nicht möglich, diese Nachrichten per Mail zu versenden. Personen, die also nicht eingeloggt sind, werden über die Anfrage nicht informiert. Der Anfragesteller muss also unter Umständen die berechtigten Personen selbst benachrichtigen, sofern diese nicht eingeloggt sind.

  Der Versand von Benachrichtigungen per Mail bei Ereignissen wie der Änderung eines Passworts ist nur möglich, wenn zusätzlich noch ein weiteres kostenpflichtiges Modul erworben wurde.
- Der Notfall-Zugriff, der über das Siegel-System ermöglicht wird, weist ebenfalls einige Schwachstellen auf.
  - Das Passwort kann nur von einem einzigen Benutzer eingesehen werden, wenn das Siegel gebrochen wurde. Das könnte unter Umständen Probleme bereiten, beispielsweise wenn diese eine Person aus irgendeinem Grund nicht auf den mit dem Passwort verbundenen Account zugreifen kann. Dann kann das Passwort

von niemandem mehr verwendet werden. Nur der Administrator kann noch das Passwort einsehen.

- Das Siegel muss von berechtigten Personen erst wieder entfernt und neu angebracht werden, bevor weitere Benutzer auf das Passwort zugreifen können. Auch das ist mit zusätzlichem Aufwand für die betroffenen Personen verbunden.
- Das Passwort wird nach der Nutzung nicht automatisch geändert und bleibt weiterhin gültig. Der Benutzer, der eine temporäre Freigabe hatte, kann das damit verbundene Passwort ohne Probleme weiterhin verwenden.

Das Programm weist also einige Nachteile auf, die in Kauf genommen werden müssen. Insbesondere das Siegelsystem für den Notfall-Zugriff besitzt einige Schwächen. Trotzdem eignet es sich grundsätzlich für den Einsatz, fast alle Kriterien werden von der Enterprise Edition erfüllt.

5.2 Bewertung von Password Manager Pro

Password Manager Pro ist eine Lösung im Bereich des Privileged Password Managements. Verfügbar sind zwei Editionen, die Standard Edition und die Premium Edition. Die Premium Edition bietet im Vergleich mit der Standard Edition einige zusätzliche Features. Für diese Bewertung wurde die Premium Edition verwendet, da die Standard Edition nicht über einige wichtige Funktionen wie etwa einen Prozess zum Notfall-Zugriff verfügt.

5.2.1 Analyse

Aufbau des Systems

Anforderung: zentrale Speicherung der Passwörter √

Die Passwörter werden in einer zentralen MySQL-Datenbank gespeichert.

Anforderung: Automatische Änderung der Passwörter √

Das System ist in der Lage, die Passwörter automatisch zu ändern. Es kann ein genauer Zeitplan angelegt werden, der spezifiziert, wann welches Passwort vom System geändert werden soll. Die Passwörter werden dann am Zielsystem geändert, sofern das System darauf zugreifen kann oder spezielle Agenten-Software auf den Zielsystemen installiert ist.

Anforderung: Erfüllung verschiedener Sicherheitskriterien √

Sowohl die Daten als auch die Datenbank werden verschlüsselt. Dabei werden grundsätzlich alle vorhandenen Informationen, nicht nur die Passwörter, verschlüsselt. Als Verschlüsselungsverfahren wird AES-256 verwendet.

Die Kommunikation zwischen Benutzer-Interface und Server verläuft über HTTPS.

Anforderung: Mehrbenutzerbetrieb √

Password Manager Pro ist auf Mehrbenutzerbetrieb ausgelegt.

Wünschenswert: Sicherung der Daten auch bei Ausfall des Servers √

Für die Sicherung der Daten bietet das Programm gleich mehrere Möglichkeiten. Es kann eine zweite Datenbank angelegt werden, deren Daten mit der Haupt-Datenbank synchronisiert werden. Außerdem wird angeboten, automatisch Backups der Daten vom System anlegen zu lassen.

Eine weitere Möglichkeit ist es, zusätzlich einen zweiten Server zu betreiben, mit dem die Daten synchronisiert werden. Zugriff auf den zweiten Server ist nur möglich, wenn der erste Server nicht mehr funktioniert.

Wünschenswert: Log mit alten Passwörtern vorhanden √

Es wird eine Historie alter Passwörter aufgezeichnet und es ist möglich, sich alte Passwörter erneut anzeigen zu lassen.

Berechtigungsverwaltung und Zugriffskontrolle

Anforderung: Zugriffskontrolle √

Der Zugriff auf die Datenbank ist nur autorisierten Personen möglich. Zur Authentisierung kann auch Two-Factor-Authentication verwendet werden. Es kann etwa ein RSA SecurID Token verwendet werden oder es ist die zusätzliche Eingabe eines weiteren von der Anwendung an den Benutzer gesendeten Passworts gefragt.

Anforderung: Anlegen von Gruppen oder Rollen ist möglich. √

Password Manager Pro setzt Role Based Access Control für die Verwaltung von Benutzerberechtigungen ein.

Es existieren vier Rollen: Administrators konfigurieren und verwalten die Anwendung, außerdem sind sie für die Verwaltung der Benutzer zuständig. Sie können die Passwörter einsehen, die sie selber angelegt haben oder für die sie die Berechtigung haben. Zusätzlich haben sie Zugriff auf die Audit-Logs.

Password Administrators können Passwörter verwalten und entsprechend ihrer Berechtigungen einsehen und ändern. Ein Administrator oder Password Administrator kann von anderen Benutzern zu einem Super Administrator gemacht werden, der alle im System vorhandenen Ressourcen verwalten darf. Password User können nur die Passwörter einsehen oder ändern, für die sie die entsprechenden Berechtigungen haben. Password Auditors können Audits einsehen.

ManageEngine empfiehlt das Anlegen eines einzigen Super Administrators, der als Notfall-Account eingesetzt werden kann. Dieser Account sollte nicht verwendet werden und das zugehörige Passwort sorgfältig aufbewahrt werden, beispielsweise in einem Tresor.

Wünschenswert: Übernahme der Benutzer und Gruppen aus Verzeichnisdiensten √

Benutzer und Gruppen können von Active Directory und Lightweight Access Directory Protocol übernommen werden. Eine Synchronisation zwischen AD/LDAP und Password Manager Pro ist auch möglich, sodass neue Benutzer automatisch hinzugefügt werden.

Erstellen der Passwörter

Anforderung: Vorgabe von Passwort-Richtlinien bei der Erstellung von Passwörtern durch den Benutzer ♢

Entsprechende Richtlinien zur Komplexität von Passwörtern können vergeben werden. Es ist auch möglich, die Nutzung bereits verwendeter Passwörter zu verbieten. Allerdings lassen sich Passwörter erstellen, die den bereits gewählten ähnlich sind.

Anforderung: Vorgabe von Passwort-Richtlinien und Generierung geeigneter Passwörter bei Nutzung eines Generators √

Die Generierung der Passwörter ist auch durch das System möglich. Das Passwort folgt dann den entsprechend vorgegebenen Richtlinien.

Anforderung: Kontrolle des Passwort-Zugriffs √

Passwörter können nur von einem Benutzer eingesehen werden, falls diese von ihm erstellt wurden oder er die entsprechenden Berechtigungen besitzt.

Anforderung: Vergabe von Zugriffsberechtigungen auf Passwörtern √

Es können sowohl Lese- als auch Schreibberechtigungen übertragen werden. Außerdem ist es möglich, die Eigentumsberechtigung für ein Passwort zu übertragen, sodass dieses Passwort von einem anderen Benutzer verwaltet werden kann.

Anforderung: Verbergen von Einträgen √

Es können nur Passwörter und Ressourcen gesehen werden, für die eine Person die entsprechende Berechtigung hat. Andere Einträge sind automatisch verborgen.

Wünschenswert: Vergabe von Passwörtern durch das System √

Passwörter können automatisch generiert werden.

Autorisierung einer temporären Zugriffsberechtigung

Anforderung: Notfall-Zugriff ist möglich. √

Es ist möglich, für ein Passwort eine strikte Zugriffskontrolle festzulegen. Dann müssen eine oder mehrere zuständige Personen zustimmen, bevor ein Benutzer ein Passwort einsehen kann. Der Zugriff ist dann zeitlich beschränkt.

Es kann auch eine Liste an Benutzern angegeben werden, die trotz der strikten Zugriffskontrolle auch ohne zusätzliche Berechtigung auf ein Passwort zugreifen können.

Anforderung: Über eine Freigabe kann von mehreren Personen entschieden werden. ♢

Es kann eine Liste an Personen angegeben werden, die über eine Freigabe entscheiden können. Je nach Einstellung ist die Zustimmung von ein oder zwei Personen erforderlich.

Anforderung: Zeitliche Beschränkung der Freigabe √

Die temporäre Freigabe ist auf ein festgesetztes Zeitlimit beschränkt, der Zugriff wird automatisch bei Überschreiten des Limits beendet. Die zuständigen Personen können auch die Berechtigung wieder entziehen. Außerdem kann der Benutzer selbst den Zugriff beenden und das Passwort damit wieder anderen Nutzern zur Verfügung stellen.

Anforderung: Automatisches Ändern des Passworts √

Das System ändert das Passwort automatisch, wenn der temporäre Zugriff beendet wurde.

Anforderung: Loggen des Notfall-Zugriffs √

In den Audit-Logs wird genau aufgezeichnet, wer wann welches Passwort abgerufen hat. Es wird auch geloggt, ob einer Anfrage auf Freigabe zugestimmt wurde oder nicht.

Freigabe der Passwörter

Anforderung: Zugriff auf einen Account ist mehreren Personen möglich. ♢

Es gibt keine Beschränkung gleichzeitiger Benutzer, außer es wurde eine strikte Zugriffskontrolle für das Passwort festgelegt. In dem Fall kann nur eine einzige Person Zugriff auf den Account haben. Es kann allerdings problematisch sein, wenn nur eine einzige Person zu einem Zeitpunkt auf einen Account zugreifen kann. Beispielsweise kann diese eine Person Verbindungsprobleme haben und den Account nicht nutzen. Damit können alle anderen Benutzer ebenfalls nicht auf den Account zugreifen.

Password Manager Pro erlaubt es allerdings, die Zugriffsberechtigung wieder zu entziehen, bevor der Zugriff beendet wurde. Dann kann ein anderer Benutzer Zugriff bekommen. Außerdem können Administratoren, wenn die Einstellungen der Datenbank entsprechend konfiguriert wurden, trotzdem weiterhin das Passwort einsehen.

Wünschenswert: Einloggen ohne Anzeigen des Passworts ist möglich. √

Password Manager Pro kann Benutzer automatisch einloggen, ohne dass ihnen das Passwort angezeigt wird. Dazu kann RDP, SSH, oder Telnet genutzt werden.

Audit

Anforderung: Existenz eines Audit-Logs √

Es gibt drei verschiedene Audits, Resource Audit, User Audit und Task Audit.

Resource Audits zeichnen alle Aktionen auf, die mit einer bestimmten Ressource im Zusammenhang stehen. Ressourcen bezeichnen bestimmte Systeme wie beispielsweise eine Datenbank. Darunter fallen zum Beispiel eine Änderung der Zugriffsrechte oder des Passworts eines bestimmten Systems. User Audits beziehen sich auf die Aktionen der Benutzer im System. In diesen Audits sind beispielsweise Informationen zum Anlegen und Löschen oder Änderungen der Berechtigungen von Benutzern vorhanden. Task Audits zeigen Zeitpläne an, zum Beispiel den nächsten Zeitpunkt eines Backups.

Zusätzlich können auch automatisch Berichte erstellt werden, die beispielsweise alle Benutzer und ihre Berechtigungen im System anzeigen.

Anforderung: Das Log kann nicht von Administratoren geändert werden. √

Die Logs werden durch die Anwendung generiert und können nicht geändert werden.

Anforderung: Der Zugriff auf die Logs ist nur eingeschränkt möglich. √

Password Administrators und Password User können nicht auf die Audit-Logs zugreifen.

Wünschenswert: Automatisches Versenden von Benachrichtigungen √

Das System kann automatisch Benachrichtigungen per E-Mail verschicken, beispielsweise an die Eigentümer von Passwörtern, wenn auf das Passwort zugegriffen oder es geändert wird.

5.2.2 Zusammenfassung der Bewertung

Password Manager Pro ist auf die Verwaltung von privilegierten Accounts in Unternehmen ausgelegt und bietet die entsprechenden Features. Ebenso wie bei der Enterprise Edition von MATESO wird das zentrale Management von Passwörtern möglich gemacht, Notfall-Zugriff auf die Kennwörter erlaubt und ausführliche Audit-Logs angelegt. Auch erlaubt Password Manager Pro es, Terminal-Sitzungen auf Zielsystemen zu starten, wobei hierfür noch mehr Optionen zur Verfügung stehen als bei der Enterprise Edition.

Zusätzlich kann Password Manager Pro die automatische Änderung von Passwörtern übernehmen. Dazu wird spezielle Agenten-Software bereitgestellt, die auf den Zielsystemen installiert werden kann. Das ermöglicht es, genaue Zeitpläne für den regelmäßigen und automatisch vom System durchgeführten Wechsel von Kennwörtern zu erstellen. Auch nach einem Notfall-Zugriff wird das entsprechende Passwort sofort geändert.

Es können auch verschiedene Password Policies angelegt werden, sodass für unterschiedliche Accounts unterschiedliche Passwort-Richtlinien gelten. Das macht es möglich, die Vergabe von Passwörtern flexibel zu gestalten.

Auch werden mehr Informationen zum Passwort-Management bereitgestellt. So wird beispielsweise genau aufgelistet, wie viele Passwörter nicht synchronisiert mit einem Zielsystem sind, nicht genutzt werden oder eine bestimmte Richtlinie verletzen.

Password Manager Pro kann auch das Management von Passwörtern zwischen Anwendungen übernehmen. Zur Kommunikation zwischen Anwendungen werden Kennwörter oft beispiels-

weise in den Code der Anwendung eingebettet. Das Programm macht es möglich, dass Anwendungen Passwörter einfach aus der Datenbank abrufen können. Zu diesem Zweck werden entsprechende APIs bereitgestellt.

Nahezu alle aufgelisteten Kriterien werden von dem Programm erfüllt. Problematisch ist aber:

- Die Wahl bereits verwendeter Passwörter kann ausgeschlossen werden, es können aber ähnliche Passwörter verwendet werden.
- Die Berechtigung für einen temporären Zugriff kann von maximal zwei Personen erteilt werden. Unter Umständen kann gefordert werden, dass eine Freigabe von mehr Personen abhängt.
- Wird ein Passwort, für das eine zusätzliche Berechtigung nötig ist, temporär freigegeben, kann es nur von einer Person genutzt werden. Die Administratoren können es aber parallel einsehen, außerdem kann die Berechtigung wieder entzogen werden. Es ist also trotzdem möglich, im Notfall das Passwort anderen Personen freizugeben.

Insgesamt eignet sich Password Manager Pro also gut für den Einsatz in Unternehmen. Es stellt die benötigten Funktionen bereit und ist damit eine sehr gute Alternative.

5.3 Gesamtübersicht der bewerteten Lösungen

Die folgende Tabelle 5.1 fasst die gesammelten Ergebnisse zusammen und gibt einen Überblick über die erfüllten, teilweise erfüllten und nicht erfüllten Anforderungen der in 4.4 dargestellten Lösung sowie der beiden in diesem Kapitel bewerteten Programme. Die Abkürzung ABE steht für das mithilfe von Attribute-Based Encryption entworfene Modell, EE für die Enterprise Password Safe Edition von MATESO und PMP für den Password Manager Pro von ManageEngine.

Tabelle 5.1: Eine Übersicht der anhand des Kriterienkatalogs bewerteten Lösungen.

	ABE	EE	PMP
Aufbau des Systems			
Anforderung: zentrale Speicherung der Passwörter	√	√	√
Anforderung: Automatische Änderung der Passwörter	×	×	√
Anforderung: Erfüllung verschiedener Sicherheitskriterien	√	√	√
Anforderung: Mehrbenutzerbetrieb	√	√	√
Wünschenswert: Sicherung der Daten auch bei Ausfall des Servers	√	√	√
Wünschenswert: Log mit alten Passwörtern vorhanden	♢	√	√
Berechtigungsverwaltung und Zugriffskontrolle			
Anforderung: Zugriffskontrolle	√	√	√
Anforderung: Anlegen von Gruppen oder Rollen ist möglich.	♢	√	√
Wünschenswert: Übernahme der Benutzer und Gruppen aus Verzeichnisdiensten	×	√	√
Erstellen der Passwörter			
Anforderung: Vorgabe von Passwort-Richtlinien bei der Erstellung von Passwörtern durch den Benutzer	×	♢	♢
Anforderung: Vorgabe von Passwort-Richtlinien und Generierung geeigneter Passwörter bei Nutzung eines Generators	×	√	√
Anforderung: Kontrolle des Passwort-Zugriffs	√	√	√
Anforderung: Vergabe von Zugriffsberechtigungen auf Passwörtern	√	√	√
Anforderung: Verbergen von Einträgen	♢	√	√
Wünschenswert: Vergabe von Passwörtern durch das System	×	√	√
Autorisierung temporärer Zugriffsberechtigung			
Anforderung: Notfall-Zugriff ist möglich.	♢	√	√
Anforderung: Über eine Freigabe kann von mehreren Personen entschieden werden.	♢	√	♢
Anforderung: Zeitliche Beschränkung der Freigabe	×	♢	√
Anforderung: Automatisches Ändern des Passworts	×	×	√
Anforderung: Loggen des Notfall-Zugriffs	♢	√	√
Freigabe der Passwörter			
Anforderung: Zugriff auf einen Account ist mehreren Personen möglich.	√	♢	♢
Wünschenswert: Einloggen ohne Anzeigen des Passworts ist möglich.	×	√	√
Audit			
Anforderung: Existenz eines Audit-Logs	♢	√	√
Anforderung: Das Log kann nicht von Administratoren geändert werden.	×	√	√
Anforderung: Der Zugriff auf die Logs ist nur eingeschränkt möglich.	√	√	√
Wünschenswert: Automatisches Versenden von Benachrichtigungen	×	♢	√

6 Zusammenfassung und Ausblick

Im Folgenden soll eine kurze Zusammenfassung über die Arbeit gegeben werden, gefolgt von einem Ausblick über die Möglichkeiten des weiteren Umgangs mit der erarbeiteten Grundlagen.

6.1 Zusammenfassung

In dieser Arbeit wurde untersucht, wie organisationsweite Passwort-Verwaltung gestaltet und kryptografisch abgesichert werden kann. Das Management von Kennwörtern in Unternehmen ist vor allem aufgrund der Existenz von privilegierten Accounts problematisch. Diese Konten haben besonders weitreichende Berechtigungen, die zugehörigen Passwörter müssen zwischen mehreren zuständigen Administratoren geteilt werden und ein Notfall-Zugriff auf diese Kennwörter muss unter Beibehaltung der Rechenschaftspflichtigkeit der Administratoren möglich sein.

Insbesondere sollte dieses Problem im Hinblick auf die Situation am LRZ untersucht werden. Dazu wurde zunächst allgemein eine Analyse vom Passwort-Management durch Benutzer und in Unternehmen durchgeführt. In einem weiteren Schritt wurden die Anforderungen ermittelt, die am LRZ an die Passwort-Verwaltung gestellt werden. Der grundlegende Aufbau eines Systems zum Passwort-Management und die Verwaltung der innerhalb dieses Systems vorkommenden Berechtigungen sowie verschiedene Anwendungsfälle, die beim Management von Passwörtern auftreten können, wurden untersucht.

Ausgehend von den Anforderungen wurde anschließend ein Kriterienkatalog entwickelt. Dieser Katalog lässt sich allgemein zur Analyse von Software-Lösungen in dem Bereich organisationsweiter Passwort-Verwaltung verwenden und kann dazu benutzt werden, einen Überblick über die angebotenen Funktionen eines Tools zu gewinnen. Dazu werden neben den Anforderungen auch Ausschlusskriterien aufgelistet, die mögliche Problemstellen bei den untersuchten Lösungen aufzeigen.

Darauf aufbauend wurden Möglichkeiten erläutert, wie sich die Passwort-Verwaltung in Unternehmen kryptografisch absichern lässt. Dazu wurden sowohl die Verwaltung von Daten innerhalb eines Systems zum Management der Kennwörter betrachtet als auch allgemein kryptografische Methoden untersucht, die ein solches Modell absichern können. Mithilfe des kryptografischen Primitivs Attribute-Based Encryption wurde anschließend ein theoretisches Modell zur Passwortverwaltung in Unternehmen dargestellt. Attribute-Based Encryption macht dabei den Verzicht auf einen vertrauenswürdigen Dritten möglich und erlaubt das

sichere Hochladen und Teilen von Passwörtern auch über einen nicht vertrauenswürdigen Server. Das Modell wurde mithilfe des Kriterienkatalogs auf Anwendbarkeit überprüft.

Zudem wurden zwei Programme, die Enterprise Password Safe Edition in Verbindung mit dem Enterprise Server von MATESO und die Password Manager Pro Premium Edition von ManageEngine, anhand des aufgestellten Kriterienkatalogs untersucht und bewertet.

6.2 Ausblick

Es gibt viele professionelle Lösungen von verschiedenen Anbietern, die die Verwaltung von Passwörtern privilegierter Accounts im Unternehmensbereich zum Ziel haben. Ein Überblick über die Einsetzbarkeit zweier Tools wurde bereits gegeben, der erstellte Kriterienkatalog kann nun zur Analyse weiterer Programme für die Verwendung am LRZ oder in Unternehmen allgemein hin genutzt werden. Damit wird ermöglicht, anhand objektiv bewerteter Produktdaten ein geeignetes Programm für das Management der Passwörter der privilegierten Accounts am LRZ auszuwählen.

Des Weiteren ist eine prototypische Implementation des in Kapitel 4.4 vorgeschlagenen Modells denkbar. Dabei sollte das Modell auf Performanz und Einsetzbarkeit getestet werden. Die Performanz des Schemas würde sich aus der für die Schlüsselgenerierung, Verschlüsselung und Entschlüsselung benötigten Zeit ableiten.

Ergebnisse dieser Arbeit können in einem nächsten Schritt ausgeweitet und auf weitere Bereiche angewendet werden. Allgemein können sie für Bereiche genutzt werden, bei denen automatisiert erzeugtes Material zentral und sicher gespeichert werden und mehreren Benutzern der Zugriff entsprechend der Berechtigungen möglich sein muss. In größeren Unternehmen ergibt sich beispielsweise der Bedarf nach einem entsprechenden Management der für die kryptografische Absicherung verwendeten Schlüssel. Oft fallen viele Schlüssel an, von denen auch Backups angelegt werden müssen. Auch in diesem Fall ist eine zentrale und sichere Verwaltung der Schlüssel erforderlich, zudem müssen zur Einhaltung verschiedener Vorschriften Audit-Informationen aufgezeichnet werden.

Auch das Management von Schlüsseln muss also entsprechend abgesichert werden. Ein Modell dazu kann mithilfe der Ergebnisse dieser Arbeit entworfen werden.

Abbildungsverzeichnis

Literaturverzeichnis

[ABS10] Albrecht Beutelspacher, Heike B. Neumann und Thomas Schwarzpaul: *Kryptografie in Theorie und Praxis.* Vieweg+Teubner Verlag, 2010.

[AS99] Adams, Anne und Martina Angela Sasse: *Users are not the enemy.* Comm. ACM, 42:40–46, 1999. http://discovery.ucl.ac.uk/20247/2/CACM%20FINAL.pdf.

[BSW07] Bethencourt, John, Amit Sahai und Brent Waters: *Ciphertext-policy attribute-based encryption.* IEEE Symposium on Security and Privacy, 2007.

[Buna] Bundesministerium für Sicherheit in der Informationstechnik: *Passwörter.* https://www.bsi-fuer-buerger.de/BSIFB/DE/MeinPC/Passwoerter/passwoerter_node.html, zuletzt aufgerufen am 10.09.2012.

[Bunb] Bundesministerium für Sicherheit in der Informationstechnik: *Regelung des Passwortgebrauchs.* https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/m/m02/m02011.html, zuletzt aufgerufen am 10.09.2012.

[Bun08] Bundesministerium für Sicherheit in der Informationstechnik: *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, 2008. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102_V1_0_pdf.pdf?__blob=publicationFile.

[FH07] Florêncio, Dinei und Cormac Herley: *A large-scale study of web password habits.* In: *WWW 07: Proceedings of the 16th international conference on World Wide Web.* ACM Press, Mai 2007. http://research.microsoft.com/pubs/74164/www2007.pdf.

[Fis01] Fisher, Dennis: *Missing the Threats Under Their Noses*, Juni 2001. http://www.eweek.com/c/a/Security/Missing-the-Threats-Under-Their-Noses/, zuletzt aufgerufen am 10.09.2012.

[GPSW06] Goyal, Vipul, Omkant Pandey, Amit Sahai und Brent Waters: *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data.* ACM conference on Computer and Communications Security, 2006.

[Imp10] Imperva Application Defense Center: *Consumer Password Worst Practices*, 2010. http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf.

[LR10] Leibniz-Rechenzentrum: *Das Leibniz-Rechenzentrum IT-Dienstleistungen*, 2010. `http://www.lrz.de/wir/Einfuehrung-LRZ.pdf`.

[LR11] Leibniz-Rechenzentrum: *Leibniz-Rechenzentrum Jahresbericht 2010*. Jahresbericht, Juni 2011. `http://www.lrz.de/wir/berichte/JB/JBer2010.pdf`.

[LYZ+10] Li, Ming, Shucheng Yu, Yao Zheng, Kui Ren und Wenjing Lou: *Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption*. SecureComm, 2010.

[Man] ManageEngine: *Password Manager Pro Premium Edition*. `http://www.manageengine.com/products/passwordmanagerpro/`, zuletzt aufgerufen am 10.09.2012.

[MAT] MATESO GmbH: *Enterprise Password Safe Edition*. `http://www.passwordsafe.de/unternehmen/enterprise-server.html`, zuletzt aufgerufen am 10.09.2012.

[PW12] Potter, Chris und Grant Waterfall: *Information security breaches survey*. Technischer Bericht, infosecurity Europe und PricewaterhouseCoopers LLP, April 2012. `http://www.pwc.co.uk/en_UK/uk/assets/pdf/olpapp/uk-information-security-breaches-survey-technical-report.pdf`.

[RSA07] RSA: *Best Practices für das Log-Management*, 2007. `http://www.rsa.com/products/envision/wp/9536_LMBP_WP_0707_DE.pdf`.

[SC12] Shini, S G und K Chitharanjan: *Secure Cloud based Medical Data exchange using Attribute based Encryption*. ACCTHPCA, Juni 2012.

[SW05] Sahai, Amit und Brent Waters: *Fuzzy Identity Based Encryption*. In: *Advances in Cryptology - Eurocrypt*, Band 3494. Springer, 2005.

[SYL10] Shucheng Yu, Cong Wang, Kui Ren und Wenjing Lou: *Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing*. Proceedings of IEEE INFOCOM 2010, 2010.