INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Evaluation von Systemen zur Mehr-Faktor-Authentifizierung am Beispiel des Leibniz-Rechenzentrums

Maximilian Miran Mizani

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Bachelorarbeit

Evaluation von Systemen zur Mehr-Faktor-Authentifizierung am Beispiel des Leibniz-Rechenzentrums

Maximilian Miran Mizani

Aufgabensteller:	Prof. Dr. Helmut Reiser
Betreuer:	Stefan Metzger Jule Ziegler
Abgabetermin:	15. Januar 2019

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

München, den 15. Januar 2019

. . . . . .
(Unterschrift des Kandidaten)

Abstract

Im Online-Umfeld kommt der Überprüfung der Echtheit von Identitäten eine wichtige Rolle zu. Da einfache sogenannte Authentifizierungsverfahren jedoch einige Unzulänglichkeiten aufweisen, ist eine Erweiterung zur Zwei- oder Mehr-Faktor-Authentifizierung für viele Anwendungen ratsam.

Am Beispiel des Leibniz-Rechenzentrums untersucht diese Arbeit verschiedene Systeme zur Mehr-Faktor-Authentifizierung im Hochschulkontext, um geeignete Produktkandidaten vorzuschlagen.

Dazu werden nach einem Überblick über Methoden und Herausforderungen der Authentifizierung Einsatzszenarien identifiziert und daraus organisatorische sowie technische Anforderungen an ein geeignetes Authentifizierungsverfahren abgeleitet, entsprechend sortiert, gewichtet und zu einem erweiterbaren Katalog zusammengesetzt. Anhand eines hieraus entwickelten technologischen Rahmenkonzepts zur Authentifizierung werden derzeit am Markt verfügbare Produkte gesucht und gegen den Anforderungskatalog geprüft.

Die Produktauswahl dieser Arbeit lieferte zwar kein eindeutiges Ergebnis, konnte die Kandidatenmenge jedoch von 75 auf drei reduzieren. Zwei derer wurden für Livetests am LRZ vorgeschlagen; einer im Rahmen einer prototypischen Implementierung bestätigt.

Inhaltsverzeichnis

1. Einleitung

Im Zeitalter steigender Digitalisierung aller Bereiche des Lebens werden viele Prozesse der Hochschulen in Forschung, Lehre und Verwaltung durch IT-Dienste erbracht.

Da die Nutzung von IT-Diensten i. d. R. ohne persönlichen Kontakt stattfindet, lassen sich falsche Identitäten digital meist recht einfach vorgaukeln. Gerade bei Diensten, die nicht von unautorisierten Personen zugreifbar sein sollen – wie E-Mail, persönliche Studienverwaltung, online-banking oder Zugängen zu kritischen Infrastruktursystemen – kommen Identitätsnachweisen und Methoden zur digitalen Überprüfung der Echtheit von Identitäten nun eine wichtige Rolle zu.

Diese sogenannte Authentifizierung stellt einen essentiellen Teil der Zugangskontrolle dar, deren Umsetzung durch bloßen Einsatz von Nutzername-Passwort-Kombinationen nicht mehr für alle Anwendungen als hinreichend verlässlich angesehen werden kann.(RFC1704, [HA94])

Die mobile Arbeitswelt mit Fernzugriffen, Web-Applikationen und cloudbasierten Diensten erlaubt es nicht mehr, sich in der Authentifizierung allein auf physische Zutrittskontrollen zu verlassen. Werden Passwörter unverschlüsselt übertragen, können diese relativ leicht abgehört werden.[BSI13h] (RFC1704, [HA94]) Aber auch verschlüsselte Passwörter lassen sich in unsicheren Netzen wie öffentlichen Hotspots oder Internetcafés abfangen und später mit Hilfe kostenloser Tools kompromittieren.

Oftmals liegt das Problem freilich auch an sorglosen Nutzern, die unsichere Passwörter wählen, Passwörter mehrfach verwenden oder selten ändern. Selbst komplexeste Passwörter bringen jedoch keinen Sicherheitsvorteil, wenn Anwender sie notieren und auf den Schreibtisch legen oder die Tastatureingaben von sogenannten Keyloggern mitprotokolliert werden.

Werden nur derart einfache Authentifizierungsverfahren eingesetzt, wäre es einem Angreifer nun verhältnismäßig leicht möglich, einem Dienst gegenüber die Identität des Nutzers erfolgreich vorzugaukeln, Zugang zu erhalten und in dessen Namen zu handeln. Bei Diensten wie dem online-banking oder Zugängen zu kritischen Infrastruktursystemen ist das potentielle Schadensausmaß gewaltig.

Abhilfe verspricht hier die Erweiterung des Authentifizierungsvorgangs um einen zweiten Faktor zusätzlich zum Passwort, der – selbst wenn er abgehört werden sollte – einem Angreifer höchstens für einen kurzen Zeitraum nützlich ist.

In der Münchner Hochschulumgebung obliegt die Stärkung der Authentifizierungsverfahren nicht dem Nutzer allein. Auch das 1962 gegründete Leibniz-Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften ist als nationales und europäisches Höchstleistungsrechenzentrum, Betreiber des Münchner Wissenschaftsnetzes (MWN) und v. a. als zentraler IT-Dienstleister der Münchner Universitäten und Hochschulen sowie einer stetig wachsenden Zahl von wissenschaftlichen Einrichtungen im Großraum München bzw. im Freistaat Bayern hier in der Verantwortung.[LRZ]

Aufgabenstellung Im Rahmen dieser Bachelorarbeit soll ein geeignetes System zur Stärkung der aktuell am Leibniz-Rechenzentrum eingesetzten Authentifizierungsverfahren durch einen

zweiten Faktor vorgeschlagen werden. Dabei soll der zu wählende zweite Faktor dem „Stand der Technik“ entsprechen, an die LRZ-Benutzerkennungen gekoppelt und in die bestehenden Authentifizierungssysteme integriert werden können. Seine Validierung soll vollständig inhouse am LRZ betrieben werden und keine biometrischen Verfahren beinhalten.

Hierfür gilt es nach einer Übersicht gängiger Methoden der Authentifizierung (Kapitel 2) für die tägliche Arbeit eines Wissenschaftlers oder Mitarbeiters an Hochschulrechenzentren in Kapitel 3 typische Szenarien bzw. Anwendungsfälle zu definieren und daraus organisatorische sowie technische Anforderungen an ein geeignetes Authentifizierungsverfahren abzuleiten. Hierauf aufbauend wird in Kapitel 4 ein technologisches Rahmenkonzept zur Zwei-Faktor-Authentifizierung am LRZ entwickelt. Anhand des Anforderungskataloges folgt anschließend in Kapitel 5 eine Analyse zur Umsetzung des Konzeptes geeigneter Produkte, die derzeit am Markt erhältlich sind. Auf Basis eines bei dieser Evaluation gewählten Produktes wird in Kapitel 6 ein Prototyp implementiert und getestet.

Diese Arbeit beschränkt sich auf Benutzerauthentifizierungsvorgänge an Mitarbeiterrechnern, Remotezugriffen per VPN oder SSH und LRZ-interne webbasierte Anwendungen. Autorisierung, Zugriffsbeschränkung oder Authentifizierung gegenüber Netzkomponenten werden nicht Teil dieser Arbeit sein.

2. Verfahren und Methoden der Authentifizierung

Wie in der Einleitung erwähnt können aus dem Vorgaukeln falscher Identitäten (Spoofing) verschiedene Risiken entstehen. Der Authentifizierungsvorgang hat zum Ziel die Identitäten der jeweiligen Kommunikationspartner (Subjekte oder Objekte) durch geeignete Nachweise wechselseitig zweifelsfrei festzustellen und so Daten und Aktionen ihren Urhebern verbindlich zuzuordnen. Dies ist u. a. zur Realisierung weiterer Sicherheitsanforderungen wie Integrität oder Vertraulichkeit nötig. Je mehr ein Authentifizierungsverfahren die Erfüllung dieses Ziels gewährleisten kann, als desto sicherer kann es angesehen werden.

Dieses Kapitel wird einen Überblick über verschiedene Varianten und Methoden der Authentifizierung geben und einige dabei bestehende Probleme bzw. Herausforderungen aufzeigen. Zuerst soll dafür aber der Begriff der Authentifizierung definiert und von verwandten Termini in diesem Kontext abgegrenzt werden.

2.1. Definition

Im Sprachgebrauch werden die Begriffe Authentifizierung, Authentisierung, Autorisierung, Authentizität und Identifikation oft synonym gebraucht bzw. verwechselt. Auch wenn die Unterschiede teils gering sind und die entsprechenden Vorgänge fast immer gemeinsam ablaufen, hilft folgende Unterscheidung in Anlehnung an [Shi07, S. 26ff.] und [BSI13a] dem Verständnis dieser Arbeit.

Mit Authentizität (Adjektiv: authentisch) ist die Echtheit, i. S. v. Original, eines Attributs oder einer Entität gemeint, d. h. die Eigenschaft wahr, überprüfbar und vertrauenswürdig zu sein. [Shi07, S. 28]

Authentisierung bezeichnet die Behauptung der Authentizität eines Attributes einer Entität zusammen mit dem Vorlegen zugehöriger Nachweise (authentifizierendes Merkmal bzw. Faktor, englisch „authenticator“ genannt). Entitäten können hier Personen, Gegenstände, Dokumente aber auch der Datenursprung einer Information sein.

Die Authentifizierung ist nun der Vorgang der Verifikation der Authentisierung, d. h. die Beurteilung der Behauptung einer Authentizität, durch Überprüfung der erbrachten Nachweise, dass es sich um das Original handle. Eine solche Verifikation kann auch durch die Bestätigung eines vertrauenswürdigen Dritten („trusted third party“) erfolgen. Ein Attribut einer Entität gilt als authentifiziert und damit (für den authentifizierenden Dienst) als authentisch, wenn die mit der Authentisierung erbrachten Nachweise positiv bestätigt werden konnten. Abbildung 2.1 veranschaulicht diesen Vorgang.

Die Identifikation kann als positive Bestätigung einer behaupteten Identität (i. S. v. „ich bin Benutzer X“) und somit als Spezialfall der Authentifizierung angesehen werden. Authentisierung und Authentifizierung sind wechselseitig abhängig voneinander und gehen in

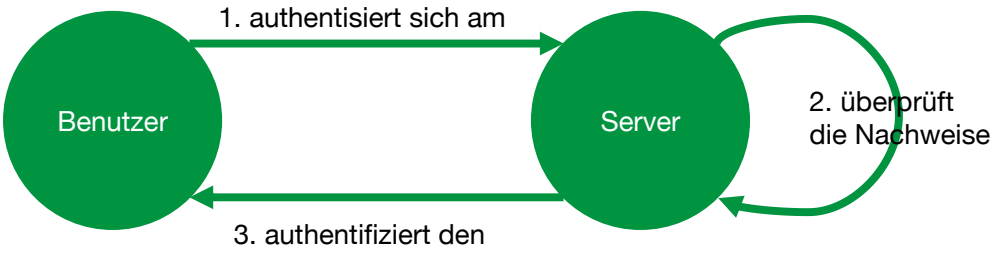

Abbildung 2.1.: Ablaufschema einer erfolgreichen Authentifizierung

der Praxis deshalb einher, sodass im Englischen beide mit „authentication“ begrifflich nicht einmal unterschieden werden.

Autorisierung beschreibt zwei Prozesse: Zum einen die initiale Vergabe von Rechten an eine Entität, also das Erteilen einer Erlaubnis in Zukunft bestimmte Aktionen ausführen zu dürfen. In Computernetzen sind damit i. d. R. Zugriffsrechte gemeint. Zum anderen beschreibt sie die Überprüfung und Entscheidung vor der Gewährung einer Aktion, ob die anfragende Entität derzeit entsprechende Rechte zu deren Ausführung besitzt. Diese zweite Art der Autorisierung sollte sinnvollerweise erst nach positiver Authentifizierung der Entität erfolgen und geht mit ihr daher meist einher.

2.2. Varianten der Authentifizierung

In der Praxis lassen sich drei Varianten von Authentifizierung unterscheiden. Authentifizierung des Datenursprungs zielt darauf ab, den Verfasser bzw. Absender einer Nachricht zweifelsfrei festzustellen, was z. B. im elektronischen Rechtsverkehr[1] eine essentielle Rolle spielt. I. d. R. werden Methoden zur Authentifizierung des Datenursprungs in Zusammenhang mit solchen zur Integritätsprüfung eingesetzt.

Für die Authentifizierung der ein- oder wechselseitigen Maschine-zu-Maschine-Kommunikation („peer entity authentication“) kommen bspw. Nachrichtenverschlüsselung mit zugehörigen Schlüsselaustauschverfahren, digitale Signaturen und (Hashed) Message Authentication Codes zum Einsatz, die zwar ein hohes Maß an Sicherheit bieten, aber wegen ihrer Komplexität für Menschen meist nicht lesbar bzw. anwendbar sind.[2]

In der Benutzerauthentifizierung, mit der sich die vorliegende Arbeit befasst, gilt es einer Maschine die Authentizität bzw. Identität des menschlichen Benutzers nachzuweisen. Im Folgenden werden hierfür aus Basis von [O’G03, AM09, Lam81, Smi01, Eck13] mögliche Nachweise klassifiziert und verglichen.

2.3. Methoden der Benutzerauthentifizierung

Das bekannte [O’G03] Klassifikationsschema teilt Authentikatoren bzw. Authentifizierungsfaktoren in die Kategorien *something you know*, *something you have* und *something you are*

[1] Ein Beispiel wäre hier die stockende Einführung von *beA* (`https://bea.bnotk.de/`)
[2] Siehe hierzu bspw. [Eck13, Kap. 8]

ein. In dieser Arbeit soll die Kategorie *something you are* jedoch leicht angepasst und als *der Person anhängendes Identifikationsmerkmal* aufgefasst werden. Hierdurch ergeben sich etwas homogenere Gruppeneigenschaften, da sich Ausweisdokumente nun in diese Gruppe einordnen lassen anstatt als Besitz-Faktoren gewertet werden zu müssen. (Vgl. [O'G03]) Somit ergeben sich die drei Klassen *Wissens-Faktoren*, *Besitz-Faktoren* und *persönliche Merkmale*.

Darüber hinaus ist die Authentisierung eines Benutzers gegenüber einem Server A auch über einen vertrauenswürdigen Dritten T (thrusted third party, *someone, who knows you*) möglich. In diesem Fall würde T den Benutzer anhand von Faktoren aus den obigen drei Kategorien authentifizieren und die Identität des Benutzers an Server A mitteilen, der auf Aussagen von T vertraut.

Diese Kategorien sind in Abbildung 2.2 aufgeführt.

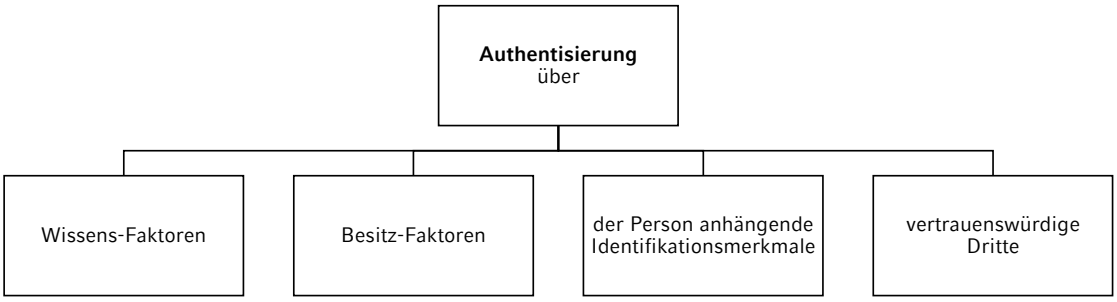

Abbildung 2.2.: Verschiedene Methoden bzw. Techniken der Authentisierung

Wissens-Faktoren stellen ein gemeinsames Geheimnis dar, das nur dem Nutzer und dem authentifizierenden Dienst bekannt ist. Ein solches Geheimnis können eine PIN, ein Passwort i. S. einer beliebigen Zeichenfolge, oder die Antworten auf eine oder mehrere vom System gestellte Fragen an den menschlichen Benutzer, aber auch die Zahlenkombination eines mechanischen Zahlenschlosses sein. Die Sicherheit von Authentifikation über Wissensfaktoren basiert auf der strengen Geheimhaltung des gemeinsamen Geheimnisses und darauf, dass es weder „leicht" zu berechnen bzw. zu erraten, noch zu finden ist.

Besitz-Faktoren, welche diese Arbeit unter *Token* zusammenfasst, sind dadurch charakterisiert, dass sie zur Authentisierung physisch bei sich getragen werden müssen und theoretisch übertragbar sind. Mechanische Schlüssel aus Metall, die sich jahrelang bewährt haben, oder Transponder in digitalen Schließsystemen ermöglichen bei Verlust oder Diebstahl dem neuen Besitzer erfolgreiche Authentifikation und somit Zugang zum abgesicherten System. Digitale Token gibt es in hardware- oder softwarebasierten Versionen. Sie sind oftmals Speicherort für digitale Zertifikate oder auch Generatoren für Einmalpasswörter, auf welche im folgenden Kapitel *2.5.1 Einmalpasswortverfahren* eingegangen werden wird. Der Besitz digitaler Tokens kann über Eingabe des aktuell gültigen vom Token generierten Einmalpasswortes, aber auch durch Beantworten einer vom System gestellten Challenge via USB-, Bluetooth- oder NFC-Schnittstelle nachgewiesen werden.[3] Für die Authentifikation mittels Besitz-Faktoren wird darauf vertraut, dass sich die Token nur im Besitz von Berechtigten befinden.

[3] Da die eingesetzten Geheimnisse bzw. Schlüssel i. d. R. sehr lang und daher meist auf einem Speichermedium abgelegt sind, der Nutzer sie also nicht direkt wissen muss, werden Callenge-Response-Verfahren in dieser Arbeit als Besitzfaktoren klassifiziert.

Geeignete Authentifizierungsfaktoren aus der Kategorie der persönlichen Merkmale kennzeichnen sich einerseits durch ihre Einzigartigkeit i. S. v. einer eineindeutigen Abbildung zwischen Merkmal und Person, Beständigkeit, quantitativer sowie performanter Erfassbarkeit durch Sensoren und Vorhandensein bei allen Menschen. Andererseits erfordern sie keine Geheimhaltung, da sie idealerweise nicht zu ersetzen bzw. neu zu erstellen, zu fälschen, vorzugaukeln oder zu übertragen sind, worauf sich die Authentifizierung verlässt. Damit umfasst die Kategorie neben biometrischen Eigenschaften auch Zertifikate bzw. Ausweisdokumente. Ein Ausweisdokument wird in der Klassifizierung dieser Arbeit nicht als Besitz-Faktoren eingeordnet, da es idealerweise nicht übertragbar sein und ein Authentifizierungsversuch mit fremden Ausweis fehlschlagen sollte. Für die Authentifizierung sollten nur Ausweisdokumente akzeptiert werden, die gewisse Schutzvorrichtungen gegen Fälschungen beinhalten (vgl. Sicherheitsmerkmale auf Geldscheinen, entsprechend lange Schlüssel bei Zertifikaten) und die von einer Institution ausgestellt wurden, der der Authentifizierende Dienst vertraut. Als hinreichend kollisionsresistente und fälschungssichere biometrische Merkmale gelten u. a. Scans von Fingerabdrücken, Iris- bzw. Retina-Struktur, DNS, Stimm- und Gesichtserkennung, Handvenenmuster, aber auch Körpergewicht, Gangart oder Tippverhalten können unterstützend hinzugezogen werden.

2.4. Herausforderungen der Authentifizierung

Jede Kategorie von Authentikatoren muss gewissen Herausforderungen und ihnen typischen Schwachstellen begegnen. Die Sicherheit eines Authentifizierungsverfahrens hängt jedoch nicht allein vom Authentifizierungsfaktor, sondern auch von menschlichen Schwächen und Rahmenbedingungen wie dem Übertragungskanal oder dem Speicherort der Authentisierungsinformationen ab.

Da die Sicherheit von Wissens-Faktoren auf Wahrung des gemeinsamen Geheimnisses beruht und Tastatureingaben bzw. Übertragungskanäle ausgespäht werden können, wird das vereinbarte Passwort bei jeder Benutzung potentiell unsicherer. Eine der größten Schwachstellen bei der Authentifizierung über Wissens-Faktoren besteht wohl darin, dass der Nutzer sich ein Passwort wählen und v. a. merken muss. Ein Passwort, das sich für Nutzer leicht merken lässt, lässt sich meist auch leicht erraten. Ein langes, zufälliges, häufig geändertes Passwort ist jedoch schwer zu merken, wird daher vom Nutzer notiert und im schlimmsten Fall einfach zugänglich verwahrt. Zudem werden Passwörter oftmals an Wörterbucheinträge angelehnt und somit nicht der gesamte Schlüsselraum, d. h. nicht die gesamte Menge der möglichen Passwörter genutzt, was das systematische Erraten des Passworts vereinfacht.

Besitz-Faktoren sind anfällig für Diebstahl und Verlust und können ggf. reproduziert werden. Da die Authentifizierung allein den Besitz des Tokens überprüft, reicht z. B. ein gefundener Schlüssel aus um Zugang zum System zu erhalten, welches mit einem entsprechenden Lesegerät ausgestattet sein muss. Abhängig von der Größe des Tokens ergeben sich darüber hinaus Bequemlichkeitseinbußen beim Mitführen dessen.

Eine Authentifizierung über persönliche Merkmale erfordert spezielle Lesegeräte, die meist nicht überall zur Verfügung stehen. Zudem können solche Merkmale wie bspw. Finderabdrücke oder Reisepass i. d. R. nicht spontan willentlich geändert werden, was bei Diebstahl bzw. erfolgreicher unbemerkter Reproduktion ein Problem für die geforderte Einzigartigkeit darstellt. Biometrische Verfahren stoßen auch oftmals aus Datenschutzgedanken auf Ablehnung, weil aus ihnen womöglich weitere Eigenschaften z. B. über Gesundheit bzw. Genetik

der Person ableitbar sind. Einige biometrische Merkmale unterliegen zudem gewissen langsamen Veränderungen durch Alterung, Krankheit oder äußeren Einflüssen. So verändern sich Stimme, Körpergewicht oder Tippverhalten mit der Zeit. Um u. a. derartige Effekte abzufangen, arbeiten Erkennungsraten biometrischer Lesegeräte schwellwertbasiert, was ein Risiko von falsch-positiv sowie falsch-negativ Entscheidungen beinhaltet.[4]

Auch wenn technische Rahmenbedingungen wie Verschlüsselung von Übertragungskanälen eingehalten sind, scheint es unbefriedigend, dass der Fund eines Tokens oder ein auf der Tastatur hinterlassener Fingerabdruck allein einem Angreifer erfolgreiche Authentifizierung und damit Zugang zum System ermöglichen kann. Zudem ist es wohl wünschenswert, dass das Authentifizierungsverfahren nicht dadurch untergraben werden kann, dass dem Benutzer beim Login vom Mobilgerät oder dem Internet-Café aus bei der Eingabe des Passworts einmal auf die Finger geschaut wurde.

Dies legt eine geeignete Kombination von Authentifizierungsfaktoren verschiedener Kategorien zur Kompensation einzelner Schwachstellen und Stärkung[5] des Verfahrens nahe.

2.5. Zweite Faktoren für die Benutzerauthentifizierung mit statischen Passwörtern

Authentifizierungsverfahren werden häufig als Kombination verschiedener, voneinander unabhängiger Authentifizierungsfaktoren unterschiedlicher Klassen realisiert, um die Vorteile einzelner Faktoren gezielt auszunutzen bzw. Schwachstellen zu kompensieren. Man spricht dann von starker Authentifizierung ([BSI13a]) bzw. von einer Zwei- (2FA) oder Multi-Faktor-Authentifizierung (MFA), die üblicherweise Wissens- mit Besitz-Faktoren kombiniert.[6]

Viele Angriffszenarien wie Phishing, Spyware, Shoulder-surfing (Tastatur und Bildschirm sind für andere sichtbar), Keylogger, Abhören der Netzleitung, Man-in-the-middle, Wörterbuchangriffe gegen schlecht gewählte Passwörter oder das Ausnutzen menschlicher Eigenschaften und Schwächen (Social engeneering) zielen darauf ab, die Zugangsdaten von Nutzern zu kompromittieren.[AM09, Kap. 3] Ein bestehendes Verfahren zur Benutzerauthentifizierung, das auf statischen Passwörtern basiert, lässt sich durch Kombination mit Besitz-Faktoren oder persönlichen Merkmalen stärken, um so Verwundbarkeiten durch oben genannte Angriffsarten zu reduzieren.[BSI13c] Dieses Kapitel stellt gängige Arten an zweiten Faktoren für statische Passwörter vor.

Zuerst wird mit einmalpasswort- und zertifikatsbasierten Systemen sowie den FIDO-Standards die Klasse der Besitz-Faktoren betrachtet. Anschließend werden als zweite Faktoren geeignete persönliche Merkmale wie Ausweisdokumente und Biometrie knapp vorgestellt

[4] Für ausführlichere Informationen hierzu siehe z. B. [O'G03, S. 11f.]

[5] Authentifizierungsverfahren werden oftmals untereinander als „stärker" bzw. „schwächer" verglichen. Ein absolutes Vergleichsmaß zu finden, ist hier sicher schwierig. [O'G03] schlägt als solches jedoch das Verhältnis von Kosten des Angriffs zu potentiellem Gewinn für den Angreifer vor, wobei Kosten seiner Auffassung nach neben Geld auch Zeit und rechtliche Folgen beinhalten. Ein Authentifizierungsverfahren ist ihm nach stark, wenn die Kosten eines Angriffes größer sind als der potentielle Gewinn. In dieser Arbeit soll mit „stark" „stärker als die meisten anderen gängigen Methoden" gemeint sein.

[6] Die Kombination z. B. mehrerer Wissens-Faktoren wird nicht als Mehr-Faktor-Authentifizierung bezeichnet, da zwei Passwörter gegenüber der Konkatenation der beiden als ein Passwort keinen Sicherheitsvorteil bringen. Auch dürfen die kombinierten Faktoren sich nicht auseinander ableiten lassen.

und Möglichkeiten moderner Smartphones zur Unterstützung des Authentifizierungsvorgangs eingeführt.

2.5.1. Besitzfaktoren (Tokens)

Digitale Tokens (Besitz-Faktoren) werden typischer Weise als Einmalpasswortsysteme oder zertifikatbasierte Challenge-Response-Verfahren implementiert. Analoge Tokens wie bspw. mechanische Schlüssel werden in diesem Kapitel nicht betrachtet.

Einmalpasswortverfahren

Werden statische Passwörter durch Sniffing des Netzverkehrs oder Shoulder-surfing ausspioniert, stehen sie dem Angreifer fortan für spätere Verwendung zur Verfügung. Abhilfe schafft hier der Einsatz von Einmalpassworten (One-Time-Passwords, OTP), die – wie der Name sagt – idealer Weise nur höchstens einmal benutzbar sind. Die Kenntnis eines bereits verwendeten OTPs ist somit für den Angreifer (nach sehr kurzer Zeit) wertlos.

OTP-Systeme sind besonders für den Einsatz bei den steigend an Bedeutung gewinnenden Web-Anwendungen im Browser geeignet, da hier im Gegensatz zu SSH- und VPN-Verbindungen clientseitig i. d. R. keine zur Authentisierung des Clients geeigneten Zertifikate zum Einsatz kommen und keine zusätzliche Software benötigt wird. In UNIX-Umgebungen lassen sich OTPs über Pluggable Authentication Module (PAM) auch in SSH und VPN einbinden.[Eck13, BSI13h]

Der Aufbau von OTP-Systemen folgt meist einer Client/Server-Architektur, in der der Client das OTP an den Server übermittelt. Der Server validiert das OTP und gibt das Ergebnis dann über eine Schnittstelle an den authentifizierenden Dienst wie z. B. RADIUS(RFC2865, [RWRS00]) weiter.

Die Sicherheit von Einmalpassworten besteht in der Idee, dass die Kenntnis eines vergangenen bzw. bereits benutzen OTPs dem Angreifer keinen Nutzen bringen solle. Dafür ist es wichtig, dass sich aus dem Wissen über vergangene OTPs nicht das nächste OTP ableiten lassen darf. Erreicht wird diese Eigenschaft durch den Einsatz von kryptographischen Hashfunktionen als Einwegfunktionen zur Generierung der OTPs. Um den Nutzer mit der notwendigen Erzeugung solcher Einmalpassworte möglichst wenig zu belasten, wird entweder zu Beginn eine Liste an OTPs vorberechnet (wie bei den ausgedruckten TAN-Listen aus den Anfangszeiten des Online-bankings) oder kontinuierlich bzw. bei Bedarf ein neues OTP erzeugt. Dazu werden i. d. R. Client und Server vom Systemadministrator out-of-band mit einem Seed initialisiert. Als Seed bezeichnet die vorliegende Arbeit einen geheimen Startwert, mit dem ein (Pseudo-)Zufallszahlengenerator oder ein Verschlüsselungsverfahren initialisiert wird.

Im Folgenden werden nun Techniken zur anfänglichen Vorberechnung und zur kontinuierlichen Neuberechnung von OTPs kurz vorgestellt.

Vorberechnete Listen Bekannter Vertreter vorberechneter Einmalpasswortlisten ist das S/Key-Verfahren aus den 1990er Jahren. (RFC1760, [Hal95]) Vereinfacht ausgedrückt wird hierbei im Client aus einem Seed s durch n-fache Anwendung einer kryptographischen Hash-Funktion f eine Folge von n Einmalpasswörtern $p_i, i = 1, \ldots, n$ generiert. Der Server wird mit $p_n = f^n(s)$ initialisiert und speichert dann nur das jeweils zuletzt verwendete p_i. Zur Authentifizierung wird diese Folge nun rückwärts abgearbeitet, d. h. $p_i = f^{n-i}(s)$. Der Client

übermittelt p_{i-1}. Da f eine Einwegfunktion ist, kann der Server (oder ein Angreifer) p_{i-1} aus p_i zwar nicht berechnen, aber er kann das OTP durch Anwendung von f validieren, wenn $f(p_{i-1}) == p_i$ gilt.[7] In der ursprünglichen Version wurden die 64-Bit Zahlen p_i zur Benutzerfreundlichkeit dann auf sechs kurze englische Wörter abgebildet. [Hal95, HMIS98, Eck13]

Auch wenn gute Hashfunktionen ausreichend Schutz vor Ableitung des nächsten OTPs gewährleisten, gilt das S/Key-Verfahren nicht mehr als sicher, da es u. a. für man-in-the-middle-Angriffe anfällig ist. [Eck13] diskutiert weitere Sicherheitslücken ausführlich.

Ein noch heute unter UNIX stellenweise verbreitetes Verfahren vorberechneter OTP-Listen ist OPIE (One-time Passwords In Everything), das auf S/Key basiert und sich über PAM auch in SSH integrieren lässt.[MAM95, Eck13]

Nach n Authentifizierungsvorgängen sind derartig vorberechnete OTP-Listen jedoch aufgebraucht und müssen vom Benutzer neu initialisiert werden.

Kontinuierliche Neuberechnung Eine Fortentwicklung dieser OTP-Verfahren stellen Techniken zur kontinuierlichen Neuberechnung von Einmalpasswörtern dar. Diese Neuberechnung kann dabei in Client und Server oder nur im Server erfolgen. In letzterem Fall fordert der Client bei einem Authentifizierungsversuch ein OTP vom Server an, welches ihm dann Out-of-Band z. B. via SMS übermittelt wird.[8] Das OTP wird dabei jedoch insgesamt zweimal übertragen, wobei es beide Übertragungskanäle zu sichern gilt. *2.5.3 Smartphones im Authentifizierungsvorgang* beschäftigt sich ausführlicher mit dem Einsatz von SMS bzw. Mobilgeräten zur Authentifizierung.

Verfahren, die das aktuell gültige OTP im Server und Client parallel berechnen, kommen ohne zweiten Übertragungskanal aus.

Hierzu werden in der Praxis häufig *RSASecurID*[9] Tokens der Firma *RSA Security* eingesetzt. Diese schlüsselanhängergroßen Hardwaretoken verfügen über einen kleinen Display, der alle 30 oder 60 Sekunden ein neues OTP anzeigt, welches der Nutzer neben seiner Benutzerkennung und dem eigentlichen Passwort bei Authentifizierungsvorgängen mit angeben muss. Nach einem erfolgreichen Hacker-Angriff auf die Firma *RSA Security* im Jahr 2011 mussten rund 40 Millionen Tokens ausgetauscht werden, da hierbei neben den geheimen Seeds der einzelnen Tokens auch die nicht offen gelegten Algorithmen zur Berechnung der OTPs gestohlen wurden. (Vgl. z. B. [Eik11])

Die *Initiative for Open Authentication*[10] (OATH[11]) formuliert in ihrem Whitepaper mitunter das Ziel einen OTP-Standard auf Basis von open-source Algorithmen zu setzen. [oat15, S. 10] Hieraus entstanden u. a. die drei als RFC veröffentlichten Verfahren HOTP, TOTP und OCRA, die heutzutage breite Anwendung finden und nun kurz vorgestellt werden.

Der HOTP (HMAC-based One-Time Password (RFC4226, [MBH+05]) Algorithmus erzeugt Einmalpasswörter zählerbasiert. Dabei wird jeweils der HMAC-SHA-1 (RFC2104,

[7] Derartige Verfahren werden in Anlehnung an den Erstautor auch *Lamport Schema* genannt.

[8] Hier sind bspw. die derzeit noch eingesetzten smsTAN-Verfahren im Online-banking zu nennen. Vgl. z. B. `https://www.sparkasse.de/service/sicherheit-im-internet/tan-verfahren.html`

[9] `https://www.rsa.com/en-us/products/rsa-securid-suite`

[10] `https://openauthentication.org/`

[11] Ausgesprochen wie das englische Wort „oath" für Schwur/Eid. OATH darf nicht mit OAuth (Open Authorization) (RFC6749, [Har12]), einem offenen Standard für API-Autorisierung und Delegation verwechselt werden.

[KBC97]) [12] aus einem nur Server und Client bekannten Seed sowie dem aktuellen Zählerwert berechnet und als Dezimalzahl auf sechs oder acht Ziffern reduziert. Das so erhaltene OTP ist dann solange gültig, bis es in einem Authentifizierungsvorgang eingesetzt wurde. Danach wird der Zähler in Server und Client inkrementiert.

Die HOTP-Zähler in Server und Client können auseinander laufen. Falls ein Angreifer einen HOTP-Token klonen und benutzen konnte, würde der reguläre Nutzer sich nun nicht mehr erfolgreich authentifizieren können und so den Angriff womöglich bemerken. Andererseits verhindert die nötige Synchronisation der Zähler den Einsatz mehrerer Instanzen eines HOTP-Tokens durch den eigentlichen Nutzer. Den gleichen HOTP-Generator auf mehreren Clients (z. B. als hardwarebasierten Schlüsselanhänger und parallel als App auf dem Smartphone) oder gegenüber mehrerer Server zu nutzen ist mittels HOTP nicht ohne Weiteres möglich. Daher ist pro Server-Client-Paar ein eigener Token nötig. Auch könnte ein Nutzer sich durch mehrfaches Inkrementieren des Zählers eine Art TAN-Liste erzeugen, ausdrucken und womöglich unsicher verwahren. Eine Resynchronisation eines HOTP-Client-Server-Paares ist durch Rücksetzen der Zähler auf einen gemeinsamen Wert relativ einfach umsetzbar. Da ein HOTP wegen der Zählerinkrementierung nur genau einmal nutzbar ist, sind Replay-Angriffe nicht umsetzbar. Man-in-the-middle-Attacken sind jedoch möglich.

Der TOTP (Time-based One-time Password (RFC6238, [MMPR11])) Algorithmus erweitert HOTP. Anstatt des zählerbasierten Ansatzes geht hier jedoch das aktuelle 30- oder 60-Sekunden-Zeitintervall neben dem Seed in die HMAC-Berechnung ein. Da Server und Client in diesem Verfahren keine Zähler in Synchronisation halten müssen, kann ein TOTP-Token sowohl auf mehreren Clients als auch gegenüber mehreren Authentifizierungsservern parallel eingesetzt werden. Bei hardwarebasierten Generatoren wird so der Nutzungskomfort durch Reduktion der Anzahl mitzuführender Tokens sicherlich erhöht. In der mobilen Arbeitswelt scheint die Möglichkeit TOTP-Softwaretokens auf Smartphone, Tablet und PC parallel vorhalten zu können von Vorteil. Aus der Nutzbarkeit paralleler Instanzen ergibt sich jedoch auch für einen Angreifer, der in Besitz des Seed gelangen konnte, die Möglichkeit der unbemerkten Generierung gültiger TOTPs.

Die Zeitabhängigkeit der TOTPs bedeutet gleichzeitig auch eine Zeitzonenabhängigkeit. Dem kann aber durch einheitliche Wahl z. B. der UTC-Zeitzone oder der UNIX-Zeit begegnet werden. Unabhängig von der Zeitzone kann die interne Uhr von TOTP-Hardwaretokens mit schwächer werdender Batterie jedoch auch langsamer laufen und außer Synchronisation geraten.

Zur Kompensation bzw. Toleranz eventueller kleiner Uhrendifferenzen und aus Bequemlichkeitsgründen um Überschreitung der Zeitintervallgrenzen während der manuellen TOTP-Eingabe abzufangen, sind in manchen Implementierungen die TOTPs benachbarter Zeitblöcke ebenfalls gültig. D. h. zu einem Zeitpunkt t können neben $TOTP(Seed, t)$ auch $TOTP(Seed, t-1)$ und $TOTP(Seed, t+1)$ zur Authentifikation genutzt werden.

Die Verwendung von Toleranzintervallen begünstigt jedoch Replay-Angriffe mit abgefangenen TOTPs, was v. a. bei Verwendung eines einzelnen TOTP-Generators für mehrere Dienste ein gewisses Sicherheitsrisiko darstellt. In manchen Implementierungen ist es innerhalb des Toleranzintervalls möglich gegenüber einem Server ein gültiges TOTP mehrfach zu nutzen. Zwar konterkariert dies die Grundidee von Einmalpasswörtern und macht Replay-Angriffe

[12] HOTP der OATH setzt auf SHA-1 als Hashfunktion zur Berechnung der HMACs. Die 2015 bekanntgewordenen Angriffe auf SHA-1 haben aber keine negativen Auswirkungen auf die Sicherheit von HOTP, da zur Berechnung von HMACs neben der verwendeten Hashfunktion auch ein geheimes Seed zum Einsatz kommt.[oat05]

erst möglich. Würde ein TOTP im Toleranzintervall allerdings nur einmal akzeptiert werden, könnte auch nur maximal ein legitimer Authentifizierungsvorgang pro Toleranzintervall stattfinden. Bei einem TOTP-Intervall von 60 Sekunden und einer Toleranz von ± 1 Zeitblock ergäbe sich im schlimmsten Fall eine Sperrzeit von knapp drei Minuten zwischen zwei Authentifizierungen. Hier gilt es zwischen Nutzungskomfort und Sicherheitsanforderungen abzuwägen.

Fazit zu OATH HOTP/TOTP Die Synchronisation von Uhrzeiten ist zwar komplexer als die einfacher Zähler, aber es handelt sich dabei um ein durch das Network Time Protokoll (NTP) und globale Zeitdienste im Internet ohne weiteres Zutun gelöstes Problem. Durch Toleranzintervalle und der Möglichkeit der Uhrensynchronisation an unabhängigen Dritten im Internet sind bei TOTP-Softwaretokens auf UNIX-Zeit Resynchronisationen durch Systemadministratoren nicht zu erwarten. Nach einem HOTP-Zählerdrift müssen die Zähler in Client und Server vom Administrator auf einen gemeinsamen neuen Wert größer als der bisherige gesetzt werden, weshalb sich für Mitarbeiter an entfernten Standorten TOTP-Tokens wohl besser eignen. Da ein vom Angreifer z. B. durch Blick auf das Display des Hardwaretokens erspähtes HOTP nicht nach kurzer Zeit sondern erst nach Benutzung verfällt, ist es für Phishing anfälliger als TOTP.

Allgemein lassen sich Einmalpasswortsysteme mit wenig Aufwand oder Änderungen an der IT-Infrastruktur in bestehende Authentifizierungssysteme integrieren, da entsprechende Schnittstellen (für bspw. RADIUS) mittlerweile i. d. R. direkt verfügbar sind. Vergabe, Sperrung und Entzug von OTP-Tokens sind mit vergleichsweise wenig Aufwand verbunden, da hier z. B. initial keine biometrische Informationen eingelesen oder Zertifikatswiderrufslisten geführt werden müssen.

Die Einrichtung softwarebasierter OTP-Tokens, d. h. die Übertragung des Seeds und die Erstsynchronisation der Zähler bzw. der Zeitzone sowie die Wahl des Algorithmus geschieht oft auf unsicherem Kanal. Für Softwaretokens in Form von Smartphoneapps haben sich QR-Codes etabliert.

Da die OTPs der Nutzerfreundlichkeit halber meist auf nur sechs oder acht Dezimalziffern gekürzt werden, ist eine spätere Wiederholung eines OTPs möglich. Bei sechsstelligen TOTPs ergibt sich eine Schlüsselraumgröße von 10^6. Mit 30-Sekunden-Erneuerungsintervall sollte sich die TOTPs also nach spätestens $5 * 10^5$ Minuten bzw. ca. 347 Tagen wiederholen. Da jedes TOTP aber nur in einem kurzen Zeitraum gültig ist, und dieser ohne Kenntnis des Seeds kaum vorherzusagen ist, scheint dieses Risiko vernachlässigbar gering. Sechsstellige HOTPs wiederholen sich spätestens nach 10^6 erfolgreichen Authentifizierungsvorgängen, was abhängig von der Häufigkeit an Logins pro Tag recht lange dauern dürfte.

Ihrer Einfachheit in Wartung und Implementierung, der recht intuitiven Bedienbarkeit und der offenen OATH-Standards wegen sind TOTP-Softwaretokens auf Smartphones, Desktops und Webanwendungen weit verbreitet. Der Google Authenticator ist eine bekannte auf HMAC-SHA-1 basierende HOTP und TOTP Implementierung.[13] Aber auch Dienste wie *Dropbox*, *GitHub* oder *OwnCloud* bieten 2FA via TOTP standardmäßig an.[14]

[13] Projektseite: `https://github.com/google/google-authenticator`

[14] Vgl. `https://help.github.com/articles/configuring-two-factor-authentication-via-a-totp-mobile-app/`,
`https://www.dropbox.com/de/help/security/enable-two-step-verification`,
`https://owncloud.com/de/owncloud-9-1-kommt-mit-integrierter-zwei-faktor-authentifizierung/`

Challenge-Response-Verfahren und PKI

Challenge-Response-Verfahren (CR) zählen zu den komplexeren Methoden der Authentifikation. Um nachzuweisen, dass der Client im Besitz eines gemeinsamen Geheimnisses bzw. Schlüssels ist, muss er hierbei eine vom Server gestellte Aufgabe lösen. Der große Vorteil besteht darin, dass das Geheimnis selbst oder Teile dessen nicht übertragen werden. CR kann zur einseitigen oder durch zweifache Anwendung zur wechselseitigen Authentifizierung beitragen.

Im Grundprinzip der verschiedenen Challenge-Response-Verfahren behauptet der Client dem Server gegenüber eine Identität. Zur Authentifizierung derer sendet ihm der Server daraufhin eine Zufallszahl (Challenge), die der Client dann mit dem gemeinsamen Geheimnis (symmetrische Variante) verschlüsselt bzw. hasht oder mit seinem privaten Schlüssel (asymmetrische Variante) signiert zurücksendet.[15] Die asymmetrische Variante erfordert jedoch eine Public Key Infrastruktur. Einsatz finden CR-Verfahren u. a. in Authentifizierungsvorgängen wie CHAP für PPP-Verbindungen (RFC1994, [Sim96]) oder an WLAN-Accesspoints.[Eck13]

Auch wenn das Geheimnis hierbei nicht direkt übertragen wird, ist es einem Angreifer als Man-in-the-middle oder durch Lauschen möglich, gestellte Challenges und zugehörige Antworten abzugreifen. Sollte sich eine Challenge später irgendwann wiederholen, kann er durch Replay der abgehörten Antwort einen erfolgreichen Spoofing-Angriff durchführen. Andererseits sind auch Known- oder Chosen-plaintext-Angriffe auf die gesammelten Challenge-Response-Paare möglich. Abhilfe versprechen verschlüsselte Challenges oder Zero-Knowledge-Verfahren.[Eck13]

OATH ORCA Die *Initiative for Open Authentication* hat mit OCRA (OATH Challenge Response Algorithm) ein flexibles CR-Framework erarbeitet, das den HOTP-Algorithmus erweitert und so eine Vielzahl an Challengearten und Parametern ermöglicht. Im Grunde berechnet OCRA dabei den HMAC-SHA{1, 256, 512} über das gemeinsame Geheimnis konkateniert mit einer Liste an Eingabeparametern, die u. a. die Challenge enthalten. Für bequemeren Einsatz bei manueller Eingabe der Response, kann das Ergebnis anschließend auf 4-10 Ziffern verkürzt werden.(RFC6287, [MRB+11]) [oat07]

Smartcards und Trusted Plattform Module Das Lösen der Challenges kann auf sogenannte Smartcards bzw. Chipkarten ausgelagert werden. Dabei handelt es sich um Plastikkarten im EC-Kartenformat mit integrierter Schaltung zur Speicherung oder Berechnung von Zugangsdaten. Hiermit ergibt sich der Vorteil, dass das zur Verschlüsselung notwendige Zertifikat oder Seed die Karte nicht verlassen muss bzw. kann und so auch von ggf. auf dem Clientterminal vorhandener Malware nicht eingesehen werden kann. Auch Trusted Plattform Module oder Physical Unclonable Functions können hier zum Einsatz kommen. (Siehe auch [SD07])

Chipkarten lassen sich in Speicher- und Prozessorchipkarten unterscheiden. Erstere bieten lediglich einige Bytes an Speicher und eigenen sich für Informationen mit eher geringerem Schutzbedarf, da sie sich relativ einfach auslesen lassen. Prozessorchipkarten verfügen darüber hinaus über einen Mikrocontroller mit schlankem Betriebssystem, der den Datenzu-

[15] OTPs sind insofern ein Spezialfall der Challenge-Response, als dass die aktuelle Challenge (Uhrzeit, Counter) dem Client stets bekannt ist und nicht übertragen werden muss.

griff regelt und Verschlüsselungsvorgänge ausführen kann. Zahlungsmittel wie GeldKarte[16] oder die elektronische Gesundheitskarte[17] beruhen auf derartiger Hardware.

Chipkarten können über kontaktbehaftete oder kontaktlose Schnittstellen wie *Radio Frequency Identification* (RFID) oder *Near Field Communication* (NFC) angesprochen werden, was im Allgemeinen besondere Lesegeräte voraussetzt.

Der Authentifizierungsvorgang mittels Smartcard ist mehrstufig. Im ersten Schritt findet i. d. R. eine Benutzerauthentifikation gegenüber der Karte statt. Diese geschieht z. B. durch PIN oder Fingerabdruck am Lesegerät oder der Karte selbst. Daran schließt eine Authentifizierung zwischen Lesegerät und Karte an.

Im zweiten Schritt finden nun Challenge-Response-Verfahren zur Authentifizierung zwischen der Karte und dem Zielsystem statt. Hierbei kommen häufig DES und 3DES zum Einsatz, da symmetrische Verschlüsselung auf den ressourcenbeschränkten Smartcards effizienter umzusetzen sind. Mittlerweile verbreiten sich jedoch auch RSA-Challenges mit „kurzen“ Schlüssellängen von bis zu 1024 Bit, wobei das Schlüsselmaterial vom Kartenhersteller oft außerhalb der Karte generiert wird.

Um bspw. an dieses Schlüsselmaterial zu gelangen, zielen Angriffe auf Smartcardsysteme entweder auf Kompromittierung des Lesegerätes oder des Chips ab. [Eck13, S. 536f.] stellt einige Angriffsszenarien und wirksame Security-by-Design-Merkmale von Smartcards vor.

Mit Smartcards realisierbare Sicherheitsmechanismen können erheblich zur Stärkung des Authentifizierungsvorgangs am Zielsystem beitragen. Ihr Einsatz setzt jedoch spezielle und v. a. vertrauenswürdige Lesegeräte voraus, die i. d. R. nicht von Haus aus an jedem Clientterminal verfügbar sind. [Eck13, S. 525ff.]

FIDO UAF und U2F Smartcard-Konzepte sind auf USB-Tokens übertragbar, die die Vorteile einer Smartcard aufweisen, ohne dabei aber ein spezielles Lesegerät vorauszusetzen.

Die nicht-kommerzielle FIDO-Allianz (Fast IDentity Online)[18] entstand 2012 als Zusammenschluss vieler marktführender Unternehmen[19] mit dem Ziel offene, herstellerunabhängige Standards zur starken Authentifikation im Internet zu entwickeln. Seitdem arbeitet die FIDO-Allianz daran, rein passwortbasierte Verfahren abzulösen bzw. durch möglichst einfach zu handhabende zweite Faktoren zu stärken. Mit UAF und U2F hat sie mittlerweile zwei Standards herausgegeben.[fid17a]

Das *Universal Authentication Framework* (UAF) Protokoll ermöglicht Webservices kryptographisch gesicherte (Multifaktor-)Authentifikation ohne Passwörter. Es definiert ein plattformunabhängiges Netzwerkprotokoll, das Zugriffberechtigungen auf Basis von PIN, Fingerabdruck, Gesichts- bzw. Stimmerkennung oder Kombination derer zuverlässig nachweisen kann.

Mit dem *Universal 2nd Factor* (U2F) Protokoll lassen sich bestehende Authentifizierungsverfahren durch einen zweiten Faktor stärken. Dabei kommt ein Hardwaretoken zum Erzeugen bzw. Speichern von Schlüsselmaterial sowie für Verschlüsselung und Signierung zum Einsatz, der über USB-, Bluetooth- oder NFC-Schnittstelle verfügt. Rein softwarebasierte

[16] `https://www.geldkarte.de/`

[17] `https://www.gematik.de/`

[18] `https://fidoalliance.org`
Im Gegensatz zu OATH beschränkt sich FIDO auf Standards zur Authentifikation im Web-Umfeld und setzt durchweg auf asymmetrische Verschlüsselungstechniken.

[19] unter anderem Infineon, Lenovo, Google, PayPal, Microsoft, ... Vgl. `https://fidoalliance.org/participate/members-bringing-together-ecosystem/`

Tokens sind zwar möglich, werden jedoch aufgrund des geringeren Sicherheitsniveaus nicht empfohlen.[fid17b] Entsprechende Hardwaretokens sind z. B. als YubiKey von Hersteller Yubico[20] bekannt.

Der Einsatz von U2F-Tokens setzt immer einen U2F-fähigen Webservice sowie Client (i. d. R. Webbrowser) voraus. *Google Chrome* und *Mozilla Firefox* bspw. in der jeweils aktuellen Version unterstützen das U2F-Protokoll nativ.[21] Webservices können das U2F-Protokoll bzw. den UAF-Standard durch Einbinden einer JavaScript API i .d. R. recht einfach implementieren.

Vor der ersten Verwendung muss der U2F-Token beim Webservice für ein bestehendes Nutzerkonto registriert werden. Im Folgenden werden Registrierung und Anmeldung vereinfacht dargestellt. [fid17b, AR16] und [Eck13, S. 584ff.] beschreiben den U2F-Authentifizierungsvorgang ausführlich.

Während der Registrierung wird in einem Challenge-Response-Handshake pro Dienst bzw. Account ein eigenes Schlüsselpaar vom U2F-Token erzeugt. Dies gewährt dem Nutzer Anonymität, da ein anderer FIDO-Server über den Schlüssel nicht auf den U2F-Token und damit den Nutzer rückschließen kann.[22] Der öffentliche Schlüssel wird dann auf dem Server, der private Schlüssel auf dem U2F-Token mitsamt einiger Informationen zur Identifizierung der Serveranwendung (AppID, Domainname, Transportprotokoll) abgelegt und verlässt diesen nie.

Bei einem späteren Anmeldevorgang fordert der Server nach Authentifizierung via Nutzername und Passwort (erster Faktor) den Besitz des zum hinterlegten öffentlichen Schlüssels gehörenden privaten Schlüssels durch Lösen einer Challenge nachzuweisen (zweiter Faktor). Der U2F-Token gleicht die Domänendaten des Servers mit seinen für dieses Konto gespeicherten ab und signiert die Challenge. Optional kann hier auch ein Präsenznachweis des Nutzers (z. B. durch Drücken eines Knopfes am Token) gefordert werden.

Während des Registrierungs- und Anmeldevorgangs vermittelt der U2F-fähige Client zwischen U2F-Token und Server. Die Verbindung zwischen Client und Server ist SSL/TLS-gesichert.

Ziel der FIDO-Allianz ist es starke Authentifikation im Web benutzerfreundlich umzusetzen. Mit U2F lässt sich stärkerer Schutz gegen Man-in-the-middle- und Phishing-Attacken erzielen als mit OTP-Tokens. In den Betrachtungen zur Sicherheit von U2F kommt [Eck13, S. 591ff.] in vielen Fällen zum Schluss, dass denkbare Angriffsszenarien außerhalb der Reichweite von U2F, sondern stattdessen z. B. in der Public-Key-Infrastructure (PKI) von SSL-Zertifikaten für Webserver gründen.

Die Einführung von FIDO erfordert clientseitig keine Änderungen an Software oder Treibern. Während der Anmeldevorgänge beschränkt sich die Nutzerinteraktion auf das Verbinden des U2F-Tokens (via USB, Bluetooth, NFC) und ggf. Drücken eines Knopfes. Die eingesetzten kryptographischen Verfahren laufen transparent für den Nutzer ab, weshalb hier wohl kaum Akzeptanzhürden zu erwarten sind.

Hardwaretokens sind jedoch immer mit Anschaffungskosten verbunden. Auch beschränken sich UAF bzw. U2F auf Authentifikation gegenüber Webservices und sind derzeit noch nicht

[20] `https://www.yubico.com/`

[21] Siehe `https://support.google.com/accounts/answer/6103523?co=GENIE.Platform%3DAndroid&hl=de` bzw. [Ber18], `https://www.yubico.com/2017/11/how-to-navigate-fido-u2f-in-firefox-quantum/`

[22] Bei SSH wird hingegen nur ein öffentlicher Schlüssel erzeugt, der dann auf mehreren Servern abgelegt wird. Mit Kenntnis dieses öffentlichen Schlüssels ist anhand der *.authorized_keys*-Dateien erkennbar, auf welche Server der Nutzer Zugriff hat.

ohne weiteren Aufwand für SSH-Verbindungen oder Desktoplogin einsetzbar.[23]

Mit FIDO2[24] unternimmt die FIDO-Allianz eine Weiterentwicklung von U2F. FIDO2 besteht aus der W3C Web Authentication API (WebAuthn)[w3c18] und dem Client to Authentication Protocol (CTAP)[25] zur Kommunikation zwischen WebAuthn-Client und externer Geräte wie Authenticator-Tokens oder Headsets. Es bietet das gleiche Sicherheitsniveau wie U2F und ermöglicht passwortlose Ein-, Zwei- oder Mehrfaktorauthentifizierung für Web- und Desktopanwendungen.

Der Verlust eines U2F-Tokens stellt nicht notwendigerweise ein Sicherheitsrisiko dar, da anhand des Tokens keine Rückschlüsse auf die mit ihm registrierten Webservices möglich sind. Der Finder könnte den U2F-Token auch problemlos zum Schützen seiner eigenen Accounts nutzen. Darüber hinaus sind zum erfolgreichen Login noch Kenntnis von Nutzername und Passwort (bzw. allgemein der erste Faktor) erforderlich.

Der Revoke eines U2F-Tokens ist natürlich nicht zentral möglich, sondern muss für jeden Webservice einzeln ausgeführt werden. Für die Recovery stellen sich dieselben Probleme wie bei jeder 2FA-Strategie, welche anwendungsspezifisch durch Backup-Login-Codes, Sicherheitsfragen oder Wiederherstellungs-Emails mit entsprechender Senkung des Sicherheitsniveaus gelöst werden. Die Registrierung eines zweiten U2F-Tokens als Backup ist eine lohnende Alternative. Zuletzt bleibt die Kontaktierung des Administrators. Anschließend kann ein neuer U2F-Token registriert werden.

Authentifizierung mit Hilfe von Bildern

Die bisher beschriebenen Verfahren setzen clientseitig eine Recheneinheit voraus, um das aktuelle OTP zu bestimmen oder die Challenge zu lösen. Diese Aufgaben können auch dem Nutzer übertragen werden, wenn sie zuvor in ein für Menschen les- und effizient lösbares Format codiert wurden. Einige Verfahrenstypen, die hierfür Bilder nutzen, werden im Folgenden kurz vorgestellt. Sie machen sich zu Nutze, dass sich Menschen Bilder und Kategorien i. d. R. besser einprägen können als lange, wirre Zeichenketten.

Ein Ansatz besteht darin, den Nutzer vorab aus einem Set an Bildern ein eigenes Portfolio auswählen zu lassen. Beim Login werden ihm dann n Bilder angezeigt, von denen m aus seinem Portfolio stammen. Die restlichen $n - m$ Bilder sind zufällig gewählt. Anstatt konkreter Bilder können auch nur Kategorien festgelegt werden. Werden jeweils alle Bilder mit Buchstaben versehen, so lässt sich hieraus leicht das aktuelle OTP bestimmen.

Um Resistenz gegen häufiges Shoulder-surfing zu erhöhen, können auch mehrere Slideshows an Bildern gezeigt und anschließend gefragt werden, ob bzw. wie viele Bilder des Portfolios enthalten waren.[YKN09]

[WWSB06] entwirft ein System, bei dem in einer großen Menge an Icons die eigenen zu finden sind. Identifizierung der Portfolio-Icons durch Shoulder-surfing wird erschwert, indem der Nutzer in deren konvexe Hülle klicken und den Vorgang mehrfach wiederholen muss.

Derartige Bild-basierte Verfahren erschweren v. a. automatisierte Angriffe. [DP00] beschreibt grundlegende Arten solcher mitsamt entsprechender Abwehrmaßnahmen wie bspw. der Verwendung von Filtern. Inwieweit die beschriebenen Methoden moderner Bilderkennung widerstehen können muss allerdings noch untersucht werden.

[23] Für Windows 10 wurde jedoch FIDO2-Unterstützung angekündigt. Siehe z. B. [mic18]
[24] `https://fidoalliance.org/fido2/`
[25] `https://www.yubico.com/solutions/fido2/`

Allgemein scheinen derartige Methoden im Vergleich zu anderen 2FA-Lösungen wie z. B. FIDO U2F-Tokens unkomfortabler und für den Nutzer deutlich aufwändiger. Des Weiteren sind sie lediglich für Loginsysteme mit graphischer Oberfläche anwendbar und (im Vergleich zu anderen OTP-Lösungen) wohl auch nur schwer „on-the-top“ in bestehende Systeme zu integrieren.

Auch wenn manche Firmen wie bspw. *Confident Technologies*[26] Bild-basierte Authentifizierungslösungen anbieten, haben sie sich wohl nur als unterstützende CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) zur Verhinderung maschineller Eingaben bzw. Logins im Webkontext durchsetzen können.[27] Eine Ausnahme bildet hier wohl SUBROSA (Simple User Based Resource Oriented Segmentation Architecture) [28], die aus der Wahl von Ausschnitten verschiedener Bilder mehrere KiloByte lange Schlüssel generieren und (i. d. R. kombiniert mit biometrischen Verfahren) zum Login nutzen. Des geringen Verbreitungsgrades wegen wird im Folgenden jedoch nicht tiefer auf sie eingegangen.

Vergleich möglicher Besitzfaktoren

Im praktischen Einsatz unterscheiden sich OTPs, Smartcards und U2F-Tokens stellenweise sehr.

Während OTP-Tokens i. d. R. keine Lesegeräte erfordern und U2F-Tokens mit standardmäßig an jedem Client vorhandenen Schnittstellen wie USB, Bluetooth oder NFC auskommen, benötigen Smartcards spezielle Lesegeräte und setzen die Installation von entsprechenden Treibern (durch die Endanwender selbst) voraus. Alle Clients mit Smartcard-Lesegeräten auszustatten stellt einen erheblichen Kostenfaktor dar und ist bei einigen Mobilgeräten wohl auch kaum möglich. Mitunter deshalb verbreiteten sich OTP-Verfahren in den letzten Jahren wieder.

Im Vergleich bieten U2F-Tokens den größten Bedienkomfort für den Nutzer. Zur Authentifizierung müssen sie lediglich über bspw. USB verbunden und deren Knopf betätigt werden. OTPs hingegen sind vom Nutzer manuell abzutippen, was den Anmeldevorgang verlangsamt.

Ein großer Unterschied zwischen OTP und U2F besteht in der zugrunde liegenden Architektur und der sich daraus ergebenden Administration.

Hardwarebasierte OTP-Token sind i. d. R. vom Hersteller mit einem Seed initialisiert[29], das vom Administrator dann auf dem gewünschten Server hinterlegt werden muss. Sollen OTPs pro Nutzer für verschiedene Dienste zum Einsatz kommen, muss entweder jeder Dienst das Seed des Nutzers kennen (was ohne Synchronisation zwischen den womöglich fremden Diensten nur mit TOTP funktionieren würde), oder es muss für jeden Dienst ein eigenes Seed erzeugt werden. Letzterer Ansatz wird von OTP-Anwendungen auf Smartphones implementiert und führt zu einer schnell wachsenden Liste von OTP-Konten, in welcher der Nutzer beim Anmeldevorgang das entsprechende Token finden muss. Wird alternativ ein zentraler Authentifikationsdienst wie RADIUS oder das OAuth-Protokoll in Kombination mit OTP eingesetzt, käme man mit nur einem Token aus.

[26] `http://confidenttechnologies.com`
[27] Gegen den bekannten reCAPTCHA-Dienst von Google (`https://www.google.com/recaptcha`) sind jedoch teils recht erfolgreiche Angriffe bekannt.[SPK16]
[28] `http://securechannels.com/products/authentication/`
[29] von welchem der Hersteller womöglich auch eine Kopie behält

Mit FIDO U2F kann der Nutzer einen einzigen Token für beinahe beliebig viele Dienste registrieren. Die Auswahl des Schlüssels während des Anmeldevorgangs übernimmt das U2F-Protokoll, weshalb die Bedienbarkeit nicht von der Anzahl der registrierten Dienste beeinträchtigt wird. Da U2F auf asymmetrische Verschlüsselung setzt, sind nur die privaten Schlüssel geheim zu halten, was der U2F-Token übernimmt. Wegen der bei OTP eingesetzten symmetrischen Verschlüsselung, sind hier Vorkehrungen zur Geheimhaltung dieser Seeds auf jedem Server (und Software-Token) zu treffen.

OTP eignet sich für den Einsatz in größeren Organisationen, da hier i. d. R. eine zentrale Authentifizierungsinstanz wie bspw. RADIUS vorhanden ist und sich OTP als zweiter Faktor somit ohne großen Aufwand auf alle mit ihr verbundenen Instanzen wie Desktops, VPNs und Webservices erstreckt.

FIDO U2F hingegen ist dezentral und daher wohl eher für den Endanwenderbereich entworfen worden, denn es trennt die einzelnen Webdienste von einander[30] und der Nutzer muss seinen Token pro Dienst selbst registrieren. Auch beschränkt sich das U2F-Protokoll auf Webservices, was sich mit FIDO2 jedoch ändern dürfte.

Das U2F-Protokoll generiert für jeden Dienst ein eigenes Schlüsselpaar, wobei Dienste dabei im Wesentlichen anhand der Domainnamen identifiziert werden. Organisationsintern sind Anwendungen jedoch oftmals über Subdomains wie `idportal.lrz.de` oder `servicedesk.lrz.de` erreichbar. Mittels *Trusted Factes* ([fid17b]) lässt sich im U2F-Protokoll eine Registrierung inklusive bestimmter Subdomains erzielen und so der initiale Aufwand für den Nutzer senken. Einige Organisationen könnten aber wohl ein Interesse daran haben, die Kontrolle über die eingesetzten 2FA-Tokens nicht dezentral den Nutzern zu überlassen, wie es sich mit U2F ergeben würde. Authentifizierungsmanagementsysteme wie bspw. *privacyIDEA*[31] können in solchen Fällen gewisse Abhilfe schaffen, indem sie u. a. ein zentrales Rollout von U2F-Tokens ermöglichen.[32]

2FA mittels OTP lässt sich verhältnismäßig schnell und günstig „on-the-top“ zu bestehenden passwortbasierten Verfahren für fast alle Anwendungsarten implementieren. Kann hierbei noch auf die Mobilgeräte der Nutzer als Tokengenerator zurückgegriffen werden, entfallen jegliche Hardwarekosten. Kapitel *2.5.3 Smartphones im Authentifizierungsvorgang* stellt weitere Möglichkeiten des Einsatzes von Smartphones zur 2FA vor.

Werden jedoch noch höhere Anforderungen an Benutzerfreundlichkeit und Sicherheit gestellt und hauptsächlich Webservices eingesetzt, sollte die Implementierung von FIDO-Standards in Betracht gezogen werden.

Allgemein lässt sich mittels Tokens nur der Besitz eines (übertragbaren) Gegenstandes nachweisen und nicht die Identität dessen, der den Token nutzt. Daher basiert die Authentifizierung über Besitzfaktoren auf der Grundannahme, dass sich der Token im Besitz der „richtigen“ Person befindet. Stellt das Einsatzszenario höchste Ansprüche an Authentizität, sollten (mitunter) der Person anhängende Identifikationsmerkmale wie Biometrie oder hoheitliche Ausweisdokumente zur Authentifizierung herangezogen werden.

[30] Diese durch eigene Schlüsselpaare pro Dienst erzielte Anonymität ist organisationsintern ggf. nicht nötig oder gewünscht.
[31] `https://www.privacyidea.org/`
[32] Vgl. `https://netknights.it/u2f-im-unternehmenseinsatz/`, `https://www.yubico.com/2016/02/otp-vs-u2f-strong-to-stronger/`

2.5.2. Der Person anhängende Merkmale

Da der Person anhängende Merkmale i. d. R. nicht übertragbar sind, lässt sich mit ihnen als Faktoren zur Authentifizierung Authentizität in hohem Maße erzielen. Identifikationsmerkmale dieser Kategorie lassen sich in Biometrie und hoheitliche Ausweisdokumente unterteilen.

Obwohl Biometrie oftmals mit hoher Authentizität in Verbindung gebracht wird, darf sie allein nicht als Kryptographieersatz angesehen werden. Herausforderungen und Möglichkeiten des sinnvollen Einsatzes biometrischer Techniken diskutieren z. B. [Eck13, S. 495ff.] oder [O'G03]. Da das LRZ den Einsatz biometrischer Verfahren zur Authentifizierung im Vorfeld dieser Arbeit ausschloss (siehe Kapitel *1 Einleitung*), wird in diesem Kapitel nicht tiefer auf sie eingegangen.

Als hoheitliche Ausweisdokumente gelten in Deutschland Reisepass und Personalausweis.

Der elektronische Reisepass (ePass) wird seit 2005 von der deutschen Bundesdruckerei herausgegeben. Der integrierte passive RFID-Chip enthält neben personen- und dokumentenbezogenen Daten auch entsprechende digitale Signaturen des Ausstellers. Über Zugriffkontrollprotokolle (BAC, PACE, EAC) lassen sich die verschiedenen Datengruppen auslesen, nachdem sich das Lesegerät dem Chip des Passes gegenüber authentisiert hat.[Eck13, S. 540ff.] Da der ePass allerdings über keine Funktion zur digitalen Authentifizierung des Inhabers verfügt (und er im Alltag auch selten bei sich getragen wird), ist er im Rahmen einer sicheren 2FA für die tägliche Arbeit am PC nicht sinnvoll einsetzbar.

Der (neue[33]) Personalausweis (nPA) in Deutschland bietet mit der sogenannten eID eine Authentisierungsfunktion gegenüber Onlinediensten sowie eine Funktion zur digitalen Signatur. Da der nPA von einer hoheitlichen Stelle ausgestellt wird, sollten sich Personen damit zweifelsfrei identifizieren lassen – also ein Höchstmaß an Authentizität der (integren) Daten gegeben sein. Eine Untersuchung des nPA als Authentifizierungsmerkmal für Dienste des LRZs wurde bereits in einer früheren Arbeit unternommen. Demnach würde „die Einbindung des nPA durchaus einen erheblichen Aufwand"([Web12]) für den Service Provider (LRZ) darstellen und auch die Benutzerfreundlichkeit stark beeinträchtigen.[Web12]

Das Auslesen von persönlichen Merkmalen erfordert geeignete Lesegeräte sowie neue Middleware auf dem Nutzerterminal, womit sich die gleichen Hürden wie bei Smartcards ergeben (siehe Kapitel *2.5.1 Smartcards und Trusted Plattform Module* oben). Für das Auslesen des nPA ist durch den Service Provider darüber hinaus noch ein kostenpflichtiges Berechtigungszertifikat beim Bundesverwaltungsamt zu beantragen.[34]

In ihrem Vergleich zwischen eID und FIDO U2F stellt [Eck13, S. 591ff.] heraus, dass beide ein durchaus ähnliches Sicherheitsniveau erzielen, sie sich jedoch im Einsatzszenario unterscheiden. Während U2F einen einfach zu handhabenden, sicheren zweiten Faktor für bestehende Konten bei Webdiensten darstellt, eignet sich der im Umgang und Implementierung umständlichere (Lesegerät, PIN) nPA durch stärkere Personenbindung des Tokens (Personalausweis) für Anwendungszwecke, die gesicherte Angaben zu personenbezogenen Daten erfordern. Ein Altersnachweis bspw. ließe sich mittels U2F nicht implementieren.

[33] seit dem 01.11.2010

[34] `https://www.personalausweisportal.de/DE/Verwaltung/Diensteanbieter_werden/diensteanbieter_node.html`

2.5.3. Smartphones im Authentifizierungsvorgang

Moderne Smartphones bzw. Mobilgeräte enthalten neben leistungsstarken Prozessoren eine Fülle an internen Sensoren, die in Authentifizierungsverfahren einbezogen werden können. Hierbei ist zu unterscheiden zwischen Verfahren der Authentifizierung des Nutzers gegenüber des Smartphones und Möglichkeiten des Einsatzes eines Smartphones als Faktor zur Authentifizierung an einem weiteren System.

Für erstere setzen viele Autoren wie[BCZ17, ZWWZ13] auf Analyse von Handbewegungsmuster mit Hilfe der eingebauten 3D-Beschleunigungssensoren, Gyroskop und Kompass zur Benutzererkennung beim Entsperren des Smartphones. Will man diese einmalige bzw. statische Authentifizierung am Beginn jeder Session zu einer während der Verwendung des Smartphones kontinuierlichen ausweiten, können zusätzlich Daten aus Touchscreen- und Tastaturinteraktion in der Erstellung des Verhaltensmusters berücksichtigt werden. [SYJ+11, DZZ13] beschreiben Modelle einer solchen für den Nutzer transparenten kontinuierlichen Authentifizierung im Hintergrund. Kamera, Mikrophon und GPS-Position ließen sich ebenfalls mit einbeziehen. Neue Smartphones der Firma Apple bspw. lassen sich mit der Funktion *Face ID*[35] durch Gesichtserkennung entsperren.

Für das Thema dieser Arbeit interessanter ist der Einsatz von Smartphones als (zweiter) Authentifizierungsfaktor, welchem natürlich eine Authentisierung des Nutzers gegenüber des Smartphones vorausgeht.

Einige Ansätze hierfür basieren auf dem Empfangen, Berechnen oder Senden von Einmalkennwörtern.(z. B. [EAK11]) Ältere Verfahren, bei denen OTPs per SMS empfangen werden, sind u. a. wegen des unsicheren Übertragungskanals für Phishing anfällig.[MBSS13] Wird das Smartphone hingegen als OTP-Generator (Softwaretoken) eingesetzt, lassen sich alle ansonsten separat mitzuführenden Token in einem Gerät vereinigen und so der Nutzerkomfort steigern. Neben dem bei hardwarebasierten OTP-Token üblichen Abtippen (verbindungsloser Token) kann die Übertragung des OTPs zum authentifizierenden Dienst auch ohne Aufwand für den Nutzer per Funkverbindung (SMS, Internet) erfolgen, was den Einsatz von langen OTPs erleichtert.

Anstelle von Einmalkennworten lassen sich mit Smartphones auch Challenge-Response-Verfahren implementieren. Neben einfachen Anrufen, die der Nutzer entgegenzunehmen hat, sind auch als QR-Codes repräsentierte Challenges möglich, welche das Gerät löst und das Ergebnis verschlüsselt an den authentifizierenden Dienst mitteilt.

Hersteller wie bspw. *DUO Security*[36] bieten mit Mobile-Push[37] eine sehr benutzerfreundliche Möglichkeit den Besitz des zweiten Faktors – ausgelöst durch Klick auf eine Push-Benachrichtigung im Smartphone – auf sicherem Kanal durch Lösen einer gestellten kryptographischen Challenge nachzuweisen. Eine Erweiterung dieser um die Analyse des Verhaltensprofils über eingebaute Sensoren oder Gesichtserkennung (siehe oben) ist denkbar.

Smartphones bieten mit ihren eingebauten Sensoren und der im Vergleich zu Smartcards deutlich höheren Rechenleistung als Authentifizierungsfaktor einige interessante Möglichkeiten und v. a. große Ersparnis an Anschaffungskosten für z. B. OTP-Token oder biometrische Lesegeräte, sofern die Nutzer ihre eigenen Smartphones für diese Zwecke bereitstellen. Die parallele private Nutzung bedingt dann jedoch hohe Anforderungen an Isolierung der an der

[35] `https://support.apple.com/de-de/HT208108`
[36] `https://duo.com`
[37] Produktname: „DUO Push“(`https://duo.com/product/trusted-users/two-factor-authentication/authentication-methods/duo-push`)

Authentifizierung beteiligten Apps von restlichen Anwendungen auf dem Gerät und gewisse Richtlinien zum Umgang mit dem Gerät. Der häufigen Nutzung wegen wird der Verlust des Smartphones i. d. R. auch schneller bemerkt als der Verlust von Schlüsselanhänger-Tokens oder Smartcards.

Funktionen wie Mobilfunk (GSM/UMTS/LTE) gewähren zwar einen out-of-band-Kanal (zum organisationsinternen Netz), können aber mit geringen Zusatzkosten an Tarifgebühren und Mobilfunkanbindung der Authentifizierungsserver verbunden sein.

Durch seine Vielseitigkeit bietet ein Smartphone die Möglichkeit gleichzeitig unterschiedliche Arten an Faktoren in sich zu vereinen und ist so für Personen interessant, die in ihrer Arbeit mit verschiedenen, nicht gemeinsam verwalteten Authentifizierungsverfahren zu tun haben. Ob eine solche Konzentration vieler Authentifizierungsfaktoren in einem Gerät gewünscht ist, muss im Anwendungsfall entschieden werden.

2.5.4. Multi-Faktor-Authentifizierung

Über eine Zwei-Faktor-Authentifizierung hinaus lassen sich auch weitere Informationen in den Authentifizierungsvorgang mit einbeziehen. Dabei muss es sich nicht zwingend um weitere Besitzfaktoren oder persönliche Merkmale handeln, sondern auch der Kontext des Authentifizierungsvorgangs kann analysiert werden.

Eine kontextbasierte Authentifizierung könnte u. a. Wochentag, Uhrzeit, IP-Adresse, Session-ID, Gerätezertifikate oder z. B. aus vergangenen Authentifizierungen abgeleitete Gerätereputation betrachten. Kommt ein Smartphone als zweiter Faktor zum Einsatz, stehen neben derartigen Eigenschaften des Terminals, von dem die Authentisierung ausgeht, auch solche des Smartphones als weitere Datenquelle zur Verfügung, sowie Analysen von Verhaltensmustern des Nutzers (siehe oben).

Daneben ließen sich ferner Eigenschaften der Ressource, auf die zugegriffen werden soll, und potentielle Auswirkungen der angefragten Operation zur Bestimmung eines Risikoindizes heranziehen. Abhängig davon könnten dann weitere Aktionen wie Reporting oder Alarmierung ausgelöst werden.

Eine Kombination von risiko- und kontextbasierten Verfahren verspricht ein deutlich erhöhtes Maß an Authentizität, da sie u. a. Phishing und Spoofing Angriffe erschwert. Auf dieser Grundlage ließe sich auch eine Erhöhung der Benutzerfreundlichkeit realisieren: Ein adaptives Authentifizierungsverfahren würde nach dem ersten Faktor nur dann weitere Besitzfaktoren oder persönliche Merkmale vom Anwender fordern, wenn es den Kontext als ungewöhnlich oder das Risiko als hoch einstufen würde.

Der Einbezug von Kontextinformationen und Risikoindizes dürfte also eine lohnende Erweiterung des Authentifizierungsvorganges darstellen.

Kapitelfazit

Zwei-Faktor-Authentifizierung ist zwar immer mit Aufwand und Kosten für Anschaffung und Verwaltung des zweiten Faktors (Tokens, Lesegeräte, ...) sowie Schulung bzw. Support der Mitarbeiter im Umgang damit verbunden. Hinzu kommt abhängig vom 2FA-Verfahren i. d. R. eine gewisse Einschränkung der Benutzerfreundlichkeit bzw. eine Verlängerung des

Anmeldevorgangs.[38]

Wegen der eingangs dargestellten Unzulänglichkeiten der Ein-Faktor-Authentifizierung und v. a. der Schwächen statischer Passwörter, werden diese Nachteile i. d. R. jedoch von dem durch 2FA gewonnenen deutlich höheren Maß an Authentizität überwogen.

In diesem Kapitel wurden verschiedene gängige Verfahren der Authentifikation über Besitzfaktoren (OTPs, Smartcards, FIDO U2F, ...) oder der Person anhängende Merkmale (Biometrie, nPA) als zweite Faktoren für statische Passwörter betrachtet sowie Konzepte zum unterstützenden Einsatz von Smartphones eingeführt.

Die vorgestellten Verfahren unterscheiden sich in den Kosten für Anschaffung bzw. Betrieb und dem Aufwand für Verwaltung sowie Implementierung bzw. Nachrüstung, da sich nicht alle Technologien ohne größere Änderungen „on-the-top“ in bestehende Systeme integrieren lassen. Sie schützen gegen verschiedene Angriffsvektoren unterschiedlich gut, erreichen verschieden hohe Maße an Authentizität und sind für den Nutzer nicht alle gleich komfortabel. Einige Verfahren erfordern einen Kommunikationskanal zwischen Server und Client oder sogar einen externen Dienstleister. Auch sind nicht alle 2FA-Verfahren für jede Zugangsart wie bspw. SSH-, VPN-, Web- oder Desktoplogin gleichermaßen anwendbar.

Die Wahl eines konkreten Verfahrens zur Zwei-Faktor-Authentifizierung hängt also vom Einsatzszenario ab.

[38] Die Anzahl der Anmeldevorgänge pro Nutzer kann in verteilten Systemen jedoch durch Single-Sign-On-Verfahren ([RR12] [Eck13, S. 515]) reduziert werden.

3. Anforderungen an ein Authentifizierungssystem am LRZ

Für einen formalen und nachvollziehbaren Produktauswahlprozess ist die Erstellung eines Katalogs an Anforderungen nötig, gegen den alle Kandidaten zu prüfen und zu bewerten sind. Im Folgenden werden nun anhand von Authentifizierungsszenarien Anforderungen an ein 2FA-System für Dienste des LRZs abgeleitet und zu einem Anforderungskatalog strukturiert.

Hierzu werden zuerst am LRZ übliche Szenarien identifiziert, deren Authentifizierungsvorgänge durch einen zweiten Faktor gestärkt werden sollen. Anschließend werden Strukturierungs- sowie Bewertungsschema des Kataloges eingeführt. Danach werden Anforderungen aus den Szenarien abgeleitet, entsprechend sortiert, gewichtet und zu einem Katalog zusammengesetzt. Um Produkte später durch eine Vorselektion effizienter gegen den Anforderungskatalog testen zu können, werden zuletzt Ausschlusskriterien festgelegt.

3.1. Szenarien der Authentifizierung am LRZ

Die folgenden am LRZ üblichen Abläufe weisen derzeit nur einfache Authentifizierungsverfahren auf. Durch einen zweiten Faktor sollen diese gestärkt und das Risiko erfolgreicher Spoofing-Angriffe gegen sie gemindert werden.

Szenario 1: Fernwartung per SSH

Das LRZ betreibt eine Vielzahl von Linux-Servern bzw. -VMs, auf die von Mitarbeitern und Forschenden häufig per Secure-Shell-Verbindung (SSH) zugegriffen wird. Obwohl die Authentifizierung hier meistens über ein Private-Public-Key-Paar erfolgt, kommen stellenweise noch Nutzername-/Passwort-Kombinationen zum Einsatz. Bei einigen Servern ist eine stärkere Authentifizierung wünschenswert.

Szenario 2: Telearbeit und VPN

Das LRZ bietet seinen Angestellten die Möglichkeit zeitweise von zu Hause aus arbeiten zu können. Der Remotelogin per VPN steht aber auch Dienstreisenden mit eigenem Laptop, Smartphone oder von einem Fremdrechner aus zur Verfügung.

Auch die meisten MWN-internen Dienste wie bspw. das Online-Zeitschriftenangebot der Universitätsbibliotheken sind von außerhalb nur via VPN zugreifbar. Hierzu kommt anwenderseitig meist der SSL-VPN-Client *Cisco Anyconnect*[1] zum Einsatz. Gleichermaßen können Mitarbeiter der einzelnen Forschungsinstitute so aus der Ferne auf ihr Institutsnetz zugreifen.

[1] `https://www.cisco.com/c/de_de/products/security/anyconnect-secure-mobility-client/index.html`

Bislang erfolgt die Authentisierung zu Beginn der SSL-VPN-Verbindung ins MWN über eine Nutzername/Passwort-Kombination, der LRZ- oder Institutskennung. Da gerade bei einem Remotelogin von einem potentiell unsicheren Ort (Mobilgerät, Fremdrechner, shoulder-surfing-gefährdete Umgebung) auf interne Ressourcen zugegriffen werden soll, scheint diese Art der Authentifizierung in einigen Fällen unzureichend. Für Authentifizierung, Autorisierung und Accounting kommt das RADIUS-Protokoll zum Einsatz.[LRZ16, S. 128]

Szenario 3: Login an Arbeitsplatzrechnern

Das LRZ stellt seinen Nutzern Desktop-Arbeitsplätze zur Verfügung. An den Windows-, Linux-, oder MacOS-Rechnern können sich Anwender an ihrem Schreibtisch oder im Rechnerpool mit ihrer LRZ- bzw. Institutionskennung einloggen und von dort aus auf weitere, ihnen freigegebene Dienste zugreifen.

Der Großteil der Arbeitsplätze sind Windows-Maschinen, von denen die meisten als Light-Desktop im Rahmen des Programms TUM-PC an der TU München betrieben werden. Auf nur einem kleinen Teil (ca. 1%) der Rechner läuft das Betriebssystem MacOS.[LRZ16, S. 49ff.]

Die Benutzerverwaltung an den Arbeitsplätzen übernehmen die MWN Active Directory Services (MWN-ADS). Die MWN-ADS sind an die zentralen Benutzerverwaltungen von TUM (TUMonline), LMU (CampusLMU) und LRZBVW (SIM) angebunden. Kennungsinformationen und Gruppenzugehörigkeiten bspw. werden so synchron gehalten.

Da die MWN-ADS einen Single Sign On für die Benutzung weiterer Dienste aus den MWN-ADS wie z. B. dem MWN-Storage bietet, scheint eine stärkere Authentifikation für manche Nutzergruppen ratsam.[LRZ15a]

Szenario 4: Login auf Webapplikationen des LRZs

Viele Dienste des LRZs lassen sich über ein Web-Interface nutzen. So stellt bspw. das *id-portal* (`https://idportal.lrz.de/`) als Web-Frontend für die LRZ-Benutzerverwaltung seinen Nutzern eine Möglichkeit zur Datenselbstauskunft oder zur Änderung ihrer hinterlegten persönlichen Daten bereit. Über ein Self-service portal (*NeSSI* – Network Self Service Interface, `https://nessi.lrz.de/NeSSI/`) können Netzverantwortliche u. a. IP-MAC-Adresszuordnungen ihrer Domänen einsehen oder gesperrte Rechner selbst entsperren. Webredakteure können über das Content-Management-System *TYPO3*[2] ihre am LRZ gehosteten Webauftritte verwalten.

Darüber hinaus betreibt das LRZ ein Ticket-System, einen Roundcube-Webmail-Client[3] und mit *Sync+Share*[4] einen auf *PowerFolder*[5] basierenden Filesharing-Dienst. Daneben agiert das LRZ für die beiden Münchner Universitäten als Shibboleth-Identity-Provider im DFN. Aber auch interne Applikationen wie die Firewall-Lösung *pfsense*[6] und ein *Confluence*-Wiki[7] zur Dokumentation im LRZ-Service-Management-System werden über ein Web-Frontend bedient.

Die meisten der genannten Dienste sind über das Internet auch von außerhalb des MWNs erreichbar. Als Logindaten dienen hier jeweils die LRZ-Benutzerkennung samt zugehörigem

[2] `https://typo3.org/`
[3] `https://roundcube.net/`
[4] `https://syncandshare.lrz.de/`
[5] `https://www.powerfolder.com/de/`
[6] `https://www.pfsense.org/`
[7] `https://de.atlassian.com/software/confluence`

Passwort. Gerade bei Diensten mit personenbezogenen bzw. sensiblen Daten oder kritischen Anwendungen wie der Firewall ist eine stärkere Authentifizierung wünschenswert. Eine entsprechende 2FA-Lösung soll jedoch nicht nur auf die genannten, sondern möglichst auf beliebige Web-Applikationen anwendbar sein.[LRZ16]

Szenario 5: Verlust von Authentisierungsinformationen und temporäre Gastzugänge

Hin und wieder kann es vorkommen, dass ein Mitarbeiter seine Zugangsdaten oder seinen zweiten Faktor vergisst bzw. verliert. In diesem Fall benötigt er einen temporären bzw. neuen Zugang oder eine Rücksetzung seiner bisherigen Zugangsdaten. Der betroffene Nutzer hat hierfür bislang seinen LRZ-Masteruser bzw. Administrator zu kontaktieren. Ebenso wird Gästen, die nur kurze Zeit einer Institution im MWN angehören, ein zeitweiser Gastzugang gewährt.

Derartige Vorgänge sollen möglichst wenig manuellen Aufwand für die Administratoren mit sich bringen. Helpdesk- und Self-Service-Funktionen sind daher wünschenswert.

Aus diesen Einsatzszenarien[8] ergeben sich Anforderungen an ein geeignetes Zwei-Faktor-Authentifizierungssystem. Diese werden in den nächsten Teilkapiteln abgeleitet, priorisiert und in einem Anforderungskatalog zusammengestellt.

3.2. Aufbau des Anforderungskataloges

Die abgeleiteten Anforderungen werden im Katalog hierarchisch gruppiert. Einerseits entstehen Gruppen, um Übersichtlichkeit zu wahren und thematischer Nähe sowie intuitiver semantischer Gewichtung bzw. Interpretation Rechnung tragen zu können. Andererseits können auf diese Weise komplexere, nicht direkt messbare Anforderungen wie bspw. komfortable Bedienbarkeit durch ein Divide-and-Conquer-Verfahren in leichter handhabbare Teilanforderungen zerlegt werden.

Dadurch ergibt sich eine Baumstruktur (zyklenfrei, gerichtet) wie in Abbildung 3.1 mit der Ausgangsfrage als Wurzelknoten und theoretisch beliebig vielen Ebenen, deren Blätter nicht alle auf demselben Level liegen müssen. Direkt messbar sind nur die Anforderungen in den Blattknoten. Im Falle dieser Arbeit ergab sich ein Baum der Höhe 5.

Durch die u. a. thematische Gruppierung werden einzelne Anforderungen aus den Szenarien des vorherigen Kapitels teilweise in andere Äste des Baumes eingeordnet und Redundanzen so vermieden.

Mit Speicherung der Information, aus welchem Szenario die Anforderung stammte, lässt sich in der späteren Bewertung der gesamte Baum oder ein gefilterter Teilbaum evaluieren.[9]

[8] Da die ersten vier Szenarien 2FA für eine konkrete Art von Anwendung thematisieren und Szenario fünf eher Verwaltungsaufgaben behandelt, wird im weiteren Verlauf der Arbeit stellenweise von vier Anwendungsszenarien und einem Verwaltungsszenario gesprochen.

[9] Die gewählte Baumstruktur ließe sich zu einem zyklenfreien, gerichteten Graphen erweitern, womit eine Anforderung dann gleichzeitig mehreren Überanforderungen bzw. Gruppen untergeordnet werden könnte. Da sich durch die in dieser Arbeit gewählte thematische Gruppierung hierfür jedoch kein Bedarf ergab, der den zusätzlichen Aufwand (z. B. Speicherung verschiedener Gewichte derselben Anforderung bzgl. verschiedener Überanforderungen etc.) gerechtfertigt hätte, wurde diese Erweiterung hier nicht unternommen.

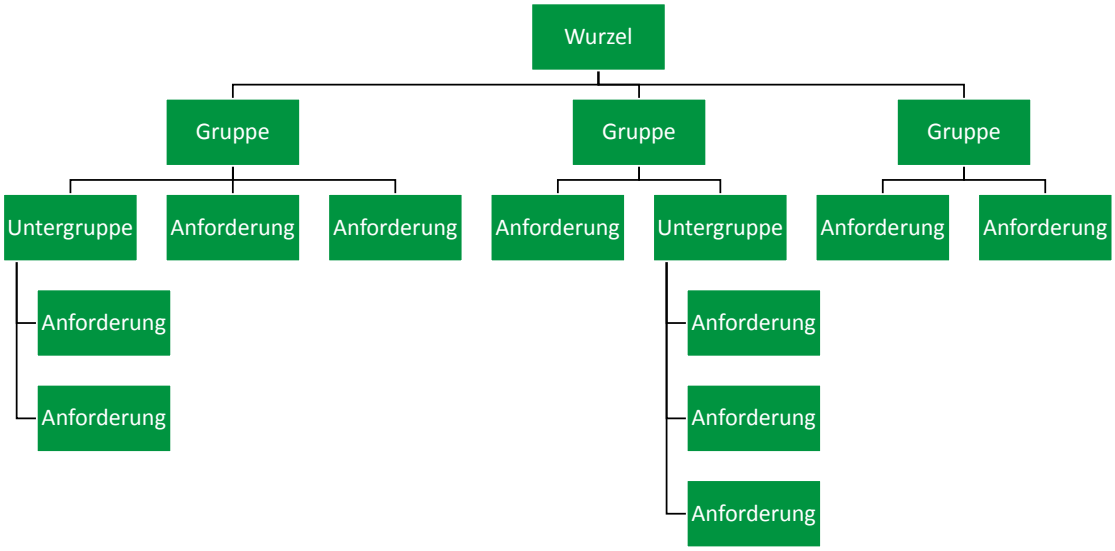

Abbildung 3.1.: Baumstruktur des Anforderungskataloges

3.3. Bewertungsschema

Um verschiedene Produkte bzw. Technologien gegen den Anforderungskatalog prüfen zu können, wird ein Bewertungsschema benötigt, das die hierarchische Struktur des Kataloges berücksichtigt. Bei der Entwicklung eines Schemas wurde Wert v. a. auf Einfachheit und intuitive semantische Interpretierbarkeit gelegt.

Jeder Anforderung wird ein Gewicht in Hinblick auf deren Relevanz zur Erfüllung der übergeordneten Anforderung zugeordnet. Der Wurzelknoten hat kein Gewicht. Auch wenn Prozentpunkte mathematisch äquivalent handhabbar wären, wurden als mögliche Gewichte folgende vier Stufen (Tabelle 3.1) gewählt, da diese intuitiv greifbarer sind.

Gewichtung	**Zahlenwert**
wünschenswert	1
relevant	2
wichtig	3
sehr wichtig	4

Tabelle 3.1.: Ausprägungen der Anforderungsgewichtung

Wird nun ein Produkt oder eine Technologie gegen den Anforderungskatalog geprüft, erfüllt es jede Anforderung zu gewissen Teilen. Zu jeder Anforderung wird daher ein Maßstab benötigt, anhand dessen die Eigenschaften des zu bewertenden Produkts auf einen Erfüllungsgrad der Anforderung abgebildet werden können. Dabei ist jeweils genau einer der vier gewählten Erfüllungsgrade aus Tabelle 3.2 möglich.

Um nicht jeden Maßstab explizit ausformulieren zu müssen, wird zur Abbildung von Produkteigenschaften auf Erfüllungsgrade das folgende implizite Schema verwendet:

Bei der Prüfung eines Produkts gegen eine Anforderung wird vorläufig von vollständiger

Erfüllungsgrad	**Zahlenwert**
Nicht erfüllt	0
Stellenweise	1
Großteils	2
Voll	3

Tabelle 3.2.: Ausprägungen der Erfüllungsgrade

Erfüllung ausgegangen. Bei Fehlen von zur Erfüllung der Anforderung notwendigen Eigenschaften des Produkts wird der vorläufige Erfüllungsgrad dann immer weiter herabgestuft bis sich schließlich ein endgültiger Erfüllungsgrad ergibt.[10]

Diese Vorgehensweise ist zwar formal nicht ganz korrekt, erspart bei dem vorliegenden recht umfangreichen Anforderungskatalog jedoch einen unverhältnismäßig hohen Arbeitsaufwand, der sich in Hinblick auf den Nutzen für die spätere Produktbewertung wohl nicht lohnen würde.

Um Produkte effizient bzgl. des Anforderungskataloges miteinander zu vergleichen, wird eine Bewertungsfunktion benötigt, die einen Teilbaum des Kataloges auf eine Bewertungszahl abbildet. Zusammengesetzten Anforderungen bzw. Gruppen wird eine Bewertungszahl zugeschrieben, die sich aus Linearkombination der Gewichte und Erfüllungsgrade ihrer untergeordneten Teilanforderungen berechnet. Wird nur ein einzelnes Blatt betrachtet, so ist dessen Bewertungszahl gleich seinem Erfüllungsgrad.

Anforderungen mit der Gewichtung 4 sind als essentiell zur Erfüllung ihrer direkt übergeordneten Anforderung zu betrachten. Werden sie durch ein Produkt nicht voll erfüllt, wird deren direkt übergeordnete Anforderung als nicht erfüllt (0 Punkte) gewertet. Mit anderen Worten: Eine Anforderung gilt als nicht erfüllt (0 Punkte), falls mindestens eine ihrer essentiellen Teilanforderungen (Gewichtung 4) nicht voll (<3 bzw. 2.5 Punkte) erfüllt ist. Dies kann nach oben kaskadieren. Anforderungen mit der Gewichtung 4 werden später (vgl. Kapitel *3.5 Ausschlusskriterien*) als Ausschlusskriterien für eine Vorselektion herangezogen.

Etwas mathematischer formuliert ergibt sich folgende Formel:

Sei a ein Teilbaum des Anforderungskataloges, p das zu bewertende Produkt, $Gewicht(a)$ die Gewichtung von a, $Erfgrad(a, p)$ der Erfüllungsgrad von Produkt p für a, $Teilanf(a)$ die Menge aller direkten Kinder von a.

Diese rekursive Bewertungsformel bietet zwei große Vorteile: 1) lässt sie sich gleichermaßen auf jeden Knoten bzw. jede Anforderung des Anforderungskataloges anwenden. Damit ergibt sich die Möglichkeit auch nur Teilbäume auszuwerten und Produkte nur hinsichtlich derer zu vergleichen. 2) liefert Formel als Ergebnis eine Zahl mit Semantik und Wertebereich der Erfüllungsgrade zurück, was die intuitive Interpretation der Bewertungszahl ermöglicht. Bei bzgl. der Ausgangsfrage sinnvoll vergebenen Gewichten und Maßstäben zeigt eine höhere Bewertungszahl daher eine höhere Erfüllung der Anforderungen und somit eine bessere Eignung des bewerteten Produkts an.[11]

[10] Da alle Kandidaten der späteren Hauptauswahl gleichzeitig gegen eine Anforderung geprüft werden, scheint man durch dieses Zugeständnis an die Reduktion des Arbeitsumfangs einen hinreichend einheitlichen impliziten Maßstab erreichen zu können. Im Gegenzug werden die Beschreibungen der Anforderungen etwas ausführlicher formuliert.

[11] Bei redundanzfreier thematischer Gruppierung und nur einem Gewicht pro Anforderung ist es nicht möglich unterschiedliche Relevanz einer Anforderung für verschiedene Szenarien abzubilden. Das Führen von Ge-

$$Bewertung(a,\ p) = \begin{cases} Erfgrad(a,p) & \text{falls } Teilanf(a) = \{\} \\ \begin{cases} 0 & \text{falls } Teilanf(a) \neq \{\} \wedge\ \exists i \in Teilanf(a): \\ & \quad Gewicht(i) = 4 \wedge Bewertung(i,p) < 2.5 \\ \dfrac{\sum_{i \in Teilanf(a)} Bewertung(i,p)}{\sum_{i \in Teilanf(a)} Gewicht(i)} & \text{falls } Teilanf(a) \neq \{\} \\ Gewicht(a) * Erfgrad(a,p) & \text{sonst} \end{cases} & \text{sonst} \end{cases}$$

Abbildung 3.2.: Rekursive Berechnungsformel des Bewertungsschemas

Jede Anforderung des Katalogs besitzt also neben Titel und Beschreibung auch die Attribute Gewichtung, $\text{Erfüllungsgrad}_{\text{ProduktX}}$, $\text{Bewertung}_{\text{ProduktX}}$. Der Übersicht halber ist daher eine tabellarische Darstellungsweise des Anforderungskataloges ratsam, welche hier auch Verwendung findet.

3.4. Erstellung des Anforderungskataloges

Im Folgenden werden nun aus den Szenarien des vorherigen Kapitels Anforderungen an ein geeignetes Zwei-Faktor-Authentifikationssystem für das LRZ abgleitet. Diese wurden um nicht-szenariospezifische Anforderungen an Bedienkomfort, Einrichtungsaufwand, Verwaltung, Sicherheitsniveau und Kosten ergänzt, die den Großteil des Kataloges ausmachen und der Erfüllung der in Kapitel *2 Verfahren und Methoden der Authentifizierung* beschriebenen Authentifizierungsziele dienen, aber auch zum Teil in subjektiven Erfahrungen gründen. Im Anschluss daran wurden diese in die in Tabelle 3.3 dargestellten Kategorien unterteilt, hierarchisch gruppiert und samt Gewichtung entsprechend als tabellarischer Anforderungskatalog dargestellt. Innerhalb einer Gruppe wurden Anforderungen thematisch sortiert.

Da der Einsatz biometrischer Verfahren im Vorfeld dieser Arbeit und nPA durch eine vorangegangene Untersuchung [Web12] ausgeschlossen worden waren, fanden konkrete Anforderungen hieran keinen Eingang in den Katalog.[12]

Unter „Mitarbeiter“ werden im Folgenden (Personen in ihrer Rolle als) Angestellte des LRZs verstanden. „Nutzer“ bezeichnet einen Nicht-Mitarbeiter, der Dienste des LRZs in Anspruch nimmt. „Anwender“ fasst beide Gruppen zusammen. Das „2FA-Backend“ beschreibt die zur Authentifizierung des zweiten Faktors notwendigen Dienste. Im „2FA-System“ sind 2FA-Backend, der zweite Faktor selbst und entsprechende Eingabeschnittstellen vereint. Das 2FA-System ist Teil des „AAA-Systems“ zur Authentifizierung, Autorisierung und Accounting.

wichten bzgl. jedes Szenarios scheint hier jedoch nur zu selten lohnenswert. In den wenigen Einzelfällen wurde daher das Maximum der szenarienspezifischen Gewichtungen gewählt.

[12] Auch ist die Granularität der Anforderungen unterschiedlich. *2-1 Integration in bestehende AAA-Systeme* bspw. könnte deutlich feingranularer formuliert werden. Für die Produktbewertung wird sich jedoch nur auf Herstellerdokumentation und Support-Angaben verlassen und von „Standard“ bzw. üblichen Konfigurationen ausgegangen.

Zur besseren Unterscheidung von Kapitel- oder Abbildungsnummern wird für Anforderungsnummern ein '-' als Trennzeichen verwendet. In einem Breitendurchlauf wird nun zu jeder Anforderung(sgruppe) deren Gewichtung kurz begründet.

Zwei-Faktor-Authentifizierung am LRZ

Anforderung	**Gewichtung**	**Szenario**
1 Rahmenanforderungen *Rahmenbedingungen für das gesamte 2FA-System des LRZs*	2	
2 Einrichtungsaufwand *Zeit- und Ressourcenaufwand der Implementierung und Einführung der 2FA-Lösung*	3	
3 Verwaltung, Wartung, Erweiterbarkeit *Administrative Tätigkeiten wie bspw. Hinzufügen neuer Nutzer oder Vergabe von temporären Zugängen (Szenario 5)*	4	
4 Sicherheitsniveau *Anforderungen aus Perspektive der IT-Sicherheit*	3	
5 Kosten(-effizienz) *Finanzielle Aspekte für das LRZ sowie einzelne Anwender*	2	
6 Bedienkomfort *Der Authentifizierungsvorgang aus Anwendersicht*	3	
7 Anwendungsbereiche *Anforderungen aus den Einsatzszenarien 1 bis 4*	4	

Tabelle 3.3.: Anforderungskatalog: Kategorie 0 (Zwei-Faktor-Authentifizierung am LRZ)

Da in dieser Arbeit ein 2FA-System für die geschilderten Einsatzszenarien gesucht wird, ist die Erfüllung eben dieser *7 Anwendungsbereiche* mit höchster Gewichtung versehen. Ebenso hoch gewichtet werden administrative Tätigkeiten für *3 Verwaltung, Wartung, Erweiterbarkeit*, da sie den kontinuierlichen Aufwand nach der Implementierung bestimmen. Hierauf folgen *2 Einrichtungsaufwand*, *4 Sicherheitsniveau* und *6 Bedienkomfort*. Allgemeine *1 Rahmenanforderungen* werden durch andere Kategorien konkretisiert. *5 Kosten(-effizienz)* tritt bei einer großen Organisation wie dem LRZ eher in den Hintergrund.

Rahmenanforderungen

Anforderung	Gewichtung	Szenario
1 Rahmenanforderungen *Rahmenbedingungen für das gesamte 2FA-System des LRZs*	2	
1-1 Universelle Lösung *Die gefundene 2FA-Lösung ist für alle Szenarien einsetzbar und erfordert nicht mehrere Teilbereichslösungen.*	3	
1-2 Einbezug bestehender Komponenten als zweiten Faktor *Bereits vorhandene Komponenten können zur Authentifizierung oder als Träger des zweiten Faktors einbezogen werden und so Kostenersparnis oder höheren Nutzungskomfort erzielen. (z.B. Smartphones, Email, RFID-Chips, ...)*	1	
1-3 Verbreitungsgrad und Akzeptanz *Die 2FA-Lösung ist auf dem Markt akzeptiert und weit verbreitet. Das System gilt als ausgereift und hat sich im Unternehmenseinsatz bewährt.*	3	
1-4 inhouse hosting *Das 2FA-System kann vollständig inhouse/on-premise gehostet werden ohne auf einen externen Dienstleister angewiesen zu sein. Am Authentifizierungsvorgang sind nur Anwenderterminal und Dienste des LRZs beteiligt.*	4	

Tabelle 3.4.: Anforderungskatalog: Kategorie 1 (Rahmenanforderungen)

In die Authentifizierung gegenüber LRZ-Diensten sollen keine externen Authentifizierungsdienstleister eingebunden sein. Das *1-4 inhouse hosting*, welches später unter *3.5 Ausschlusskriterien* auch als Ausschlusskriterium gesetzt wird, erhält daher höchste Gewichtung. Zur Reduzierung der Komplexität der Infrastruktur und des administrativen Aufwands ist eine *1-1 Universelle Lösung* wichtig, die möglichst auf alle Einsatzszenarien anwendbar ist. Hierbei soll es sich nicht um ein Nischenprodukt, sondern um eine praxiserprobte, auf dem Markt verbreitete Lösung handeln, was meist ein Reifegradindiz darstellt. Der Einbezug bereits am LRZ eingesetzter Komponenten wie RFID-Chips oder Smartphones der Anwender als zweiten Faktor ist für Bedienkomfort und Kostenersparnis wünschenswert, aber für die Wahl der 2FA-Technologie nicht ausschlaggebend.

Einrichtungsaufwand

Anforderung	Gewichtung	Szenario
2 Einrichtungsaufwand *Zeit- und Ressourcenaufwand der Implementierung und Einführung der 2FA-Lösung*	3	
2-1 Integration in bestehende AAA-Systeme *Die Authentifizierung des zweiten Faktors lässt sich in bestehende AAA-Systeme integrieren.*	4	
2-2 Zusätzliche IT-Infrastruktur *Es ist möglichst wenig zusätzliche IT-Infrastruktur (Hard- und Software) im AAA-Backend nötig.*	2	
2-3 Aufwand der Einführung *Der Zeit- und Ressourcenaufwand der Inbetriebnahme des 2FA-Systems (Rollout) ist gering.*	3	
2-3-1 Import bestehender Anwenderkennungen *Alle bestehenden Anwenderkennungen lassen sich initial mit geringem Aufwand in der 2FA-Verwaltungssoftware registrieren.*	4	
2-3-2 Hardwareanpassung der Anwenderterminals *An den Terminals der mit 2FA auszustattenden Anwendungen sind möglichst keine Hardwareanpassungen nötig (neue Schnittstellen, Lesegeräte etc.).*	3	3
2-3-3 Anpassung der Anwendungen *Es sind möglichst wenig bzw. unkomplizierte Anpassungen der mit 2FA auszurüstenden Anwendungen nötig (neue Eingabefelder, APIs etc.).*	2	
2-3-4 Verteilen der zweiten Faktoren an die Anwender *Das Verteilen der zweiten Faktoren an die Anwender ist mit möglichst geringem Aufwand verbunden.*	3	
2-4 Einrichtungsaufwand für den Anwender *Der Nutzer kann seinen zweiten Faktor mit möglichst geringem Initialaufwand in Betrieb nehmen.*	2	2
2-4-1 Clientseitige Einrichtung auch durch technisch nicht-versierte *Die clientseitige Einrichtung der 2FA erfordert keine gesteigerten technischen Fachkenntnisse.*	3	2
2-4-2 Fremdrechner *Bei der Nutzung eines nicht vorkonfigurierten Fremdrechners ist kein großer Einrichtungsaufwand für die Nutzung von 2FA gegenüber LRZ-Diensten und insb. kein Administratorzugang erforderlich. (Ausnahme: Desktoplogin am Fremdrechner)*	3	
2-5 Technische Dokumentation *Alle Komponenten des 2FA-Systems sind ausführlich dokumentiert.*	3	

	2-6 Anleitungsmaterial für Endanwender *Der Hersteller liefert (Vorlagen für) an Endanwender gerichtete Anleitungen zur Interaktion mit dem 2FA-System (u.a. 2F-Login, Self-Service-Funktionen, Vorgehensweisen bei Diebstahl/Verlust). Für das LRZ entsteht kaum redaktioneller Aufwand zur Erstellung solcher.*	1	

Tabelle 3.5.: Anforderungskatalog: Kategorie 2 (Einrichtungsaufwand)

Um zur Einführung einer 2FA kein gänzlich neues System für Authentifizierung, Autorisierung und Accounting aufbauen zu müssen, ist es sehr wichtig, dass sich die Abfrage des zweiten Faktors in das bestehende AAA-System integrieren lässt. Zusätzliche IT-Infrastruktur im AAA-Backend kann dafür in Kauf genommen werden. Um den Aufwand der Einführung aus Sicht des LRZs darüber hinaus möglichst gering zu halten, ist der Import bestehender Anwenderkennungen in die 2FA-Verwaltungssoftware gefolgt vom Aufwand des Rollouts der zweiten Faktoren an die Anwender wichtig. Da Anpassungen der mit 2FA auszurüstenden Anwendungssoftware als studentische Praktika ausgegeben werden könnten, wird ihr Aufwand geringer gewichtet als nötige Änderungen der Terminalhardware, welche i. d. R. Handarbeit erfordern.

Der Einrichtungsaufwand für den einzelnen Anwender tritt etwas in den Hintergrund. Da das LRZ nicht nur Institute der Informatik unterstützt, ist eine clientseitige Einrichtung der 2FA auch durch technisch nicht-versierte Anwender wichtig. Dabei sollen keine Administratorrechte erforderlich sein, da sonst von einem nicht vorkonfiguriertem Fremdrechner aus ggf. nicht auf Dienste des LRZs zugegriffen werden könnte.

Eine solide technische Dokumentation des 2FA-Systems ist während der Einrichtung wichtig. Bereitgestelltes Anleitungsmaterial für Endanwender würde die Arbeit des Helpdesks erleichtern.

Verwaltung, Wartung, Erweiterbarkeit

Anforderung	Gewichtung	Szenario
3 Verwaltung, Wartung, Erweiterbarkeit *Administrative Tätigkeiten wie bspw. Hinzufügen neuer Nutzer oder Vergabe von temporären Zugängen (Szenario 5)*	4	
3-1 Erweiterbarkeit *Neue Dienste und Nutzeraccounts können aufwandsarm mit 2FA ausgestattet werden.*	3	
3-1-1 Neue Dienste *Das 2FA-System lässt sich nachträglich mit geringem Aufwand auf weitere Anwendungen ausweiten. (Der Aufwand ist deutlich geringer als für die Einrichtung des 2FA-Systems.)*	2	
3-1-2 Neue Anwender *Die Anzahl der 2FA-Zugänge ist praktisch nur durch das Lizenzpaket limitiert.*	4	
3-1-3 Weitere 2FA-Technologien *Neben der hauptsächlich gewählten, lassen sich nach der Einführung auch weitere Arten von zweiten Faktoren in das 2FA-System integrieren.*	1	
3-1-4 Adaption an den „Stand der Technik“ *Das 2FA-System ist offen für verbesserte Versionen eingesetzter 2FA-Technologien (z.B. höhere Schlüssellängen o.ä.).*	3	
3-2 Wartung *Der Aufwand zur Instandhaltung des 2FA-Systems ist gering.*	3	
3-2-1 Supportzusage *Technologie und Support der Anbieterfirma werden nicht in absehbarer Zeit eingestellt. Die Software wird aktiv gewartet und weiterentwickelt.*	3	
3-2-2 Kontinuierlicher Aufwand des zweiten Faktors *Die kontinuierliche Pflege jedes zweiten Faktors erfordert wenig Aufwand (Batteriewechsel, Aktualisierung von Keys etc.).*	3	
3-2-3 Kontinuierlicher Aufwand der 2FA-Dienste *Der kontinuierliche Aufwand für Schutz und Pflege des 2FA-Servers sowie der ausgestatteten Anwendungen ist gering.*	2	
3-2-4 Backup *Das 2FA-System lässt sich mit allen relevanten Komponenten sichern und wiederherstellen.*	4	
3-3 Verwaltung *Typische Verwaltungsaufgaben erfordern wenig Aufwand (Arbeitszeit).*	4	
3-3-1 Funktionsumfang der Verwaltungssoftware *Die Software unterstützt die Verwaltung des 2FA-Systems maßgeblich.*	3	

	3-3-1-1 Automatisierbarkeit von Routinearbeiten *Die Software unterstützt u.a. automatische Vergabe und Rollout von zweiten Faktoren.*	4	
	3-3-1-2 Unterstützung verschiedener 2FA-Technologien *Die Software kann verschiedene Arten von zweiten Faktoren über ein einheitliches Interface verwalten.*	1	
	3-3-1-3 Self-Service *Die Software bietet umfangreiche Funktionen zur Verwaltung der eigenen 2FA durch den Anwender.*	3	5
	3-3-1-4 Helpdesk-Rolle *Neben Nutzer- und Administrator-Rolle verfügt die Verwaltungssoftware auch über eine Helpdesk-Rolle.*	1	5
	3-3-1-5 Verwaltung von Hardwaretoken *Die Software unterstützt bei der Verwaltung von Hardwaretoken (Inventarisierung, Verteilung).*	1	
	3-3-2 Auswirkung auf bisherigen Prozess zur Kennungsverwaltung *Im bisherigen Prozess zur Vergabe, Verwaltung und Überprüfung von Berechtigungen und Zugangsdaten sind keine großen Änderungen nötig.*	3	
	3-3-3 Änderung des zweiten Faktors *Art oder Key des zweiten Faktors eines Accounts ist nachträglich änderbar oder neu ausstellbar (durch Nutzer oder Admin).*	3	5
	3-3-4 Wiederherstellung nach Diebstahl/Verlust *Für Ersatz nach Diebstahl oder dauerhaftem Verlust des zweiten Faktors ergibt sich für den Admin oder Helpdesk höchstens geringer Aufwand.*	2	5
	3-3-5 Sperren von zweiten Faktoren *Nach Diebstahl oder dauerhaftem Verlust des zweiten Faktors kann dieser mit geringem Aufwand gesperrt werden.*	3	
	3-3-6 Anwender-Self-Service nach Verlust des zweiten Faktors (Fallback) *Ein Anwender kann seinen zweiten Faktor nach Vergessen oder dauerhaftem Verlust selbst sperren oder wiederherstellen (Recovery-Codes o.ä.).*	2	5
	3-3-7 Mehrere zweite Faktoren pro Anwender *Einem Anwenderaccount können mehrere zweite Faktoren zugeordnet werden. (z.B. als Backup bei Verlust)*	1	
	3-3-8 Geltungsbereich der 2FA *Die 2FA ist pro Dienst und Anwender(-gruppe) durch den Admin (und ggf. den Nutzer) einzeln aktivierbar.*	4	3
	3-3-9 Ausweitung auf synchronisierte Benutzerverwaltungen *Die Einführung der 2FA darf sich nicht zwangsweise auch auf die zentralen Benutzerverwaltungen der Universitäten und Institute erstrecken.*	4	3
	3-3-10 Portierungsmöglichkeit *Ein bestehender zweiter Faktor kann einem anderen Nutzer zugeordnet werden. (teure Hardwaretoken)*	1	

	3-3-11 Temporäre Zugänge *Zweite Faktoren lassen sich auch temporär (tagesweise) oder ersatzweise ausstellen, falls der eigene Token bspw. zu Hause vergessen wurde.*	3	5
	3-3-11-1 Gastzugänge *Zeitlich beschränkte neue Zugänge mit 2FA sind ausstellbar.*	3	5
	3-3-11-2 Tagesersatz für zweiten Faktor *Ein temporärer Ersatz für den zweiten Faktor, falls dieser z.B. von einem Dienstreisenden zu Hause vergessen wurde, lässt sich unter Wahrung der Authentizität zuteilen.*	2	5
	3-3-11-3 Einrichtungsaufwand für temporäre Zugänge *Der Einrichtungsaufwand für temporäre bzw. 2FA-Gastzugänge ist gering.*	3	5
	3-3-12 Einfluss auf Verwaltung des ersten Faktors *Der zweite Faktor bringt kaum zusätzlichen Aufwand zur Verwaltung des ersten Faktors mit sich.*	2	
	3-3-13 Rücksetzung des ersten Faktors *Eine Rücksetzung des ersten Faktors zieht nicht notwendigerweise eine Rücksetzung des zweiten Faktors mit sich. (z.B. teure Hardwaretoken)*	2	5

Tabelle 3.6.: Anforderungskatalog: Kategorie 3 (Verwaltung, Wartung, Erweiterbarkeit)

In einer großen Organisation mit derart vielen Anwendern wie dem LRZ kommt *3 Verwaltung, Wartung, Erweiterbarkeit* des 2FA-Systems hoher Stellenwert zu. Da gerade durch die beiden Münchner Universitäten semesterweise mehrere Tausend neue Accounts hinzukommen, ist eine angemessene Verwaltung bzw. operativer Betrieb des 2FA-Systems essentiell.

Die häufigste Form der Erweiterung des 2FA-Systems des LRZs dürfte wohl im Hinzufügen neuer Nutzeraccounts bestehen. Weitere Dienste an die 2FA anzubinden wird nach der Einführung dagegen weit seltener vorkommen. Andere als die hauptsächlich gewählte Art an zweiten Faktoren verwenden zu können ist für Nutzer wünschenswert, die aus anderem Kontext bereits (mehrfach verwendbare) zweite Faktoren besitzen. Für den langfristig sicheren Betrieb wichtiger jedoch ist eine Adaption der eingesetzten zweiten Faktoren an den „Stand der Technik“ z. B. durch Erhöhung der Schlüssellängen o. ä.

Zur Instandhaltung bzw. Wiederherstellung des 2FA-Systems sind Backupmöglichkeiten unerlässlich. Bei der großen Zahl an auszugebenden zweiten Faktoren ist es wichtig den kontinuierlichen Wartungsaufwand jedes einzelnen Faktors (z. B. Batteriewechsel, Aktualisierung von Schlüsselmaterial) gering zu halten. Der Aufwand zur Pflege der 2FA-Server bzw. Anwendungen tritt dagegen etwas zurück. Um notwendige Updates bzw. Patches zu erhalten, ist es wichtig eine Anbieterfirma bzw. -organisation zu wählen, die ihre Arbeit nicht in absehbarer Zeit einstellen wird.

Eine qualitative Verwaltungssoftware, die v. a. Automatisierung von Routinearbeiten, aber auch gewisse Nutzer-Self-Service-Funktionen und ggf. Unterstützung für verschiedene Arten von zweiten Faktoren bietet, ist für einen effizienten operativen Betrieb wichtig. Eine Helpdesk-Rolle und spezielle Unterstützung zur Verwaltung von Hardwaretoken könnten von Vorteil sein.

Um Einführungshürden zu senken, soll der zweite Faktor keinen zusätzlichen Aufwand zur Verwaltung des ersten Faktors mit sich bringen. Eine Änderung des ersten Faktors soll auch

nicht notwendigerweise eine Änderung des zweiten Faktors nach sich ziehen, was gerade beim Einsatz von z. B. Hardwaretoken einen deutlichen Mehraufwand bedeuten würde.

Um den Servicedesk zu entlasten ist eine Anwender-Self-Service-Funktion zur Wiederherstellung nach Verlust des zweiten Faktors wichtig. Hierfür wäre es u. a. hilfreich einem Anwender mehrere (Instanzen an) zweiten Faktoren als Backup zuordnen zu können. Der Aufwand zur Wiederherstellung durch den Admin kann daher geringer gewichtet werden, als das Sperren nach einem Diebstahl des zweiten Faktors. Den zweiten Faktor bei Änderung dessen Trägers (z. B. Smartphone) portieren zu können, ist wünschenswert.

Damit nicht alle Nutzer zum Einsatz von 2FA verpflichtet werden, ist es essentiell diese pro Nutzer(-gruppe) und pro Dienst einzeln aktivieren zu können. Ein Self-Service hierzu wäre hilfreich. Sehr wichtig ist außerdem, dass die Einführung der 2FA sich nicht zwangsweise auf die mit den LRZ-Kennungen synchronisierten Benutzerverwaltungen der Universitäten und Institute erstreckt. Eine Kompatibilität aller von den Universitäten selbst betriebenen Anwendungen mit dem 2FA-System des LRZs ist nicht ohne weiteres anzunehmen.

Für Gäste oder Projektmitarbeiter sind temporäre Zugänge mit möglichst geringem Einrichtungsaufwand für den Administrator wichtig. Sollte ein Dienstreisender seinen zweiten Faktor zu Hause vergessen, ist temporärer Ersatz relevant.

Sicherheitsniveau

Anforderung	Gewichtung	Szenario
4 Sicherheitsniveau *Anforderungen aus Perspektive der IT-Sicherheit*	3	
4-1 Nichtabstreitbarkeit *Die Kombination aus erstem und zweitem Faktor lässt zweifelsfrei auf Authentizität des Anwenders schließen.*	2	
4-2 Angemessenes Sicherheitsniveau *Das 2FA-System verfügt über ein ausreichend hohes Sicherheitsniveau für die kommenden fünf Jahre.*	3	
4-2-1 Stand der Technik *Der eingesetzte zweite Faktor wird voraussichtlich für die nächsten fünf Jahre noch als „Stand der Technik“ gelten.*	3	
4-2-2 Zweiter Faktor nicht ableitbar *Der zweite Faktor ist weder aus Kenntnis des ersten, noch durch Wissen über vorangegangene Verwendungen ableitbar.*	3	
4-2-3 Angemessene Verschlüsselungsverfahren *Im Zuge der 2FA kommen jederzeit angemessene, dem „Stand der Technik“ entsprechende Verschlüsselungsverfahren für Übertragung und Speicherung zum Einsatz.*	3	
4-2-4 Kontextbasierte MFA *Das System verfügt über die Möglichkeit Kontextinformationen als weitere Faktoren in den Authentifizierungsvorgang mit einzubeziehen.*	1	
4-3 Schutz gegen Spoofing *Der Mindestaufwand für erfolgreiches Spoofing ist hoch.*	3	
4-3-1 Fund des zweiten Faktors nicht ausreichend *Fund oder Diebstahl eines zweiten Faktors allein eröffnet kein wesentliches Risiko für erfolgreiches Spoofing (Smartcard mit Fingerabdrucksensor, OTP-App mit PIN, FIDO etc.).*	3	
4-3-2 Replay *Das 2FA-System verhindert Replay-Angriffe nach Phishing oder Sniffing auf dem Übertragungskanal.*	3	
4-4 Übertragung des zweiten Faktors *Der zweite Faktor wird während des Authentifizierungsvorgangs in angemessen sicherer Weise übertragen.*	3	
4-4-1 Out-of-Band *Der zweite Faktor wird „out-of-Band“ übertragen.*	1	
4-4-2 Anzahl der Übertragungen *Der zweite Faktor wird höchstens einmal übertragen (Client nach Server)*	1	
4-4-3 Kein Klartext *Der zweite Faktor wird nie im Klartext übertragen.*	4	

	4-5 Injektivität *Unterschiedlichen Benutzeraccounts werden unterschiedliche Ausprägungen des zweiten Faktors zugeordnet (ein Nutzer hat nicht den gleichen 2F für alle seine Accounts)*	2	
	4-6 Verfügbarkeit *Eine Zwei-Faktor-Authentifizierung ist zu jedem Zeitpunkt möglich.*	4	
	4-6-1 Ausfallsicherheit des 2FA-Systems *Die Dienste zur Authentifizierung des zweiten Faktors lassen sich redundant auslegen und Ausfallsicherheit so gewährleisten.*	4	
	4-6-2 Intervall der Authentifizierungsvorgänge *Die nötige Wartezeit zwischen zwei Authentifizierungsvorgängen ist akzeptabel gering.*	1	
	4-6-3 Physische Robustheit *Der zweite Faktor ist robust gegen physische Beschädigungen.*	2	
	4-7 Manipulation des zweiten Faktors *Der zweite Faktor bzw. dessen Träger sowie entsprechende Lesegeräte sind gegen unbemerkte Manipulation geschützt.*	3	
	4-8 Keine Speicherung des zweiten Faktors *Der zweite Faktor kann von Anwendung oder Terminal nicht in einer zur erfolgreichen späteren Analyse geeigneten Art gespeichert werden.*	2	

Tabelle 3.7.: Anforderungskatalog: Kategorie 4 (Sicherheitsniveau)

Die Einführung eines zweiten Faktors folgt dem Ziel, einen Mehrwert zur Sicherheit des Authentifizierungssystems zu leisten. Da die Verfügbarkeit der LRZ-Dienste nicht durch die Verfügbarkeit des 2FA-Systems reduziert werden soll, kommt letzterer hoher Stellenwert zu. Eine redundante Auslegung scheint hierfür unerlässlich. Die Verfügbarkeit der einzelnen zweiten Faktoren selbst in Form von Robustheit gegen physische Schäden und geringer Wartezeit zwischen zwei Authentifizierungsvorgängen wird geringer gewichtet, da diese Fälle akut jeweils nur einen Anwender betreffen.

Ein angemessenes Sicherheitsniveau, das für die nächsten fünf Jahre durch entsprechende Verschlüsselungsverfahren dem Stand der Technik entspricht, ist von Bedeutung. Hierzu müssen *brute force* Angriffe oder das Ableiten des zweiten Faktors als aussichtslos gelten.

Im Sinne der Authentizität und Nichtabstreitbarkeit ist der Schutz gegen Spoofing-Versuche durch Replay oder Fund eines zweiten Faktors ebenso wichtig wie der Schutz des zweiten Faktors gegen unbemerkte Manipulation oder Speicherung durch Anwendung oder Terminal. Um Phishing zu erschweren, ist eine sichere Übertragung des zweiten Faktors wichtig, in der er möglichst Out-of-Band, möglichst selten und keinesfalls im Klartext versandt wird.

Kosten(-effizienz)

Anforderung	Gewichtung	Szenario
5 Kosten(-effizienz) *Finanzielle Aspekte für das LRZ sowie einzelne Anwender*	2	
5-1 Für das LRZ *Kosten für das LRZ für Einführung und Betreib einer 2FA*	3	
5-1-1 Pro Mitarbeiter *Kosten des zweiten Faktors pro Mitarbeiter.*	3	
5-1-1-1 Anschaffung pro Mitarbeiter *Anschaffungskosten des zweiten Faktors pro Mitarbeiter*	1	
5-1-1-2 Kontinuierlich pro Mitarbeiter *Kontinuierliche Kosten des zweiten Faktors pro Mitarbeiter*	3	
5-1-1-3 Wiederherstellung nach Verlust pro Mitarbeiter *Kosten für die Wiederherstellung nach Diebstahl/Verlust des zweiten Faktors pro Mitarbeiter*	1	
5-1-2 2FA-Infrastuktur *Kosten für Anschaffung und Unterhalt der nötigen Hard- und Software des 2FA-Systems (inkl. Lizenzen)*	2	
5-1-3 Verwaltung von 2FA-Authentifizierungsinformationen *Kosten für Erstellen, Verknüpfen oder Zurücksetzen des zweiten Faktors*	3	
5-1-4 Späteres Hinzufügen neuer 2FA-Nutzer *Kosten für das Ausstatten später hinzukommender weiterer Anwenderaccounts mit 2FA*	1	
5-1-5 Pro auszustattender Anwendung *Kosten pro Anwendung(sinstanz), deren Authentifizierung mittels 2FA gestärkt werden soll*	2	
5-2 Für Nutzer von LRZ-Diensten *Kosten für Nicht-Mitarbeiter, die 2FA für LRZ-Dienste nutzen wollen*	2	
5-2-1 Anschaffung für den Nutzer *Anschaffungskosten des zweiten Faktors pro Nutzer*	2	
5-2-2 Kontinuierlich für den Nutzer *Kontinuierliche Kosten des zweiten Faktors pro Nutzer*	3	
5-2-3 Wiederherstellung nach Verlust für den Nutzer *Kosten für die Wiederherstellung nach Diebstahl/Verlust des zweiten Faktors pro Nutzer*	2	

Tabelle 3.8.: Anforderungskatalog: Kategorie 5 (Kosten(-effizienz))

Bei einer großen Organisation wie dem LRZ rücken die Kosten für ein 2FA-System etwas in den Hintergrund.

Die für das LRZ entstehenden Kosten pro Mitarbeiter, wobei einmalige Kosten (Anschaffung bzw. Wiederherstellung) geringer gewichtet werden als kontinuierliche, pro mit 2FA aus-

zustattender Anwendung und solche für Anschaffung und Unterhalt der 2FA-Infrastruktur treten hinter diejenigen für Verwaltung von Authentifizierungsinformationen zurück, da letztere durch die große Zahl semesterweise neu hinzukommender Accounts stärker ins Gewicht fallen.

Bedienkomfort

Anforderung	Gewichtung	Szenario
6 Bedienkomfort *Der Authentifizierungsvorgang aus Anwendersicht*	3	
6-1 Einarbeitungszeit für Anwender *Einarbeitungszeit und Schulungsbedarf für neue 2FA-Nutzer bzw. Gäste sind durch intuitive Bedienbarkeit gering.*	3	3, 5
6-2 Verlängerung des Authentifizierungsvorgangs *Der zusätzliche Zeitaufwand bei der Authentifizierung (für Anwender und System) ist gering.*	4	3
6-3 Unkomplizierte Handhabung *Die Eingabe des zweiten Faktors im Rahmen des Authentifizierungsvorgangs ist für den Anwender wenig aufwändig.*	3	
6-4 Verschiedene Authentisierungsmethoden *Dem Nutzer stehen verschiedene Authentisierungsmethoden zur Auswahl.*	1	
6-5 Branding *Die 2FA-Lösung lässt sich in das LRZ-Design überführen.*	1	
6-6 Mehrbelastung der Anwender *Der zweite Faktor ist leicht mitzuführen und verursacht nur eine geringe Mehrbelastung der Mitarbeiter.*	4	2, 3
6-6-1 Transportierbarkeit *Der zweite Faktor ist leicht physisch mitzuführen.*	4	2, 3
6-6-2 Einheitlichkeit *Jeder Anwender soll für LRZ-Dienste möglichst gleichartige und wenige zweite Faktoren benötigen.*	3	
6-7 Private Nutzung *Der zweite Faktor lässt sich ohne Verringerung des Sicherheitsniveaus auch für private Zwecke nutzen.*	1	
6-8 Redundante zweite Faktoren *Pro Anwender lassen sich redundante zweite Faktoren anlegen (OTP-App auf mehreren Geräten, Backup-FIDO-Stick).*	1	
6-9 Design der User-Interfaces *Das Design der User-Interfaces ist übersichtlich und ansprechend gestaltet. Es entspricht modernen Gestaltungsprinzipien. (Loginmasken, Self-Service-Portal etc.)*	2	

Tabelle 3.9.: Anforderungskatalog: Kategorie 6 (Bedienkomfort)

Auch wenn das durch 2FA erreichte, höhere Maß an Authentizität ein wichtiges Ziel ist, darf dadurch der tägliche operative Betrieb nicht gravierend eingeschränkt werden. Daher ist es sehr wichtig, dass der Authentifizierungsvorgang durch den zweiten Faktor zeitlich nicht drastisch verlängert wird. Zur komfortablen Bedienung des 2FA-Systems durch (gelegentliche) Anwender sind weiterhin geringe Einarbeitungszeit bzw. geringer Schulungsbedarf sowie eine unkomplizierte Handhabung des zweiten Faktors während des Authentifizierungsvorgangs wichtig. Gerade bei Mitarbeitern des LRZs, die sich täglich mehrmals authentifizieren müssen, kann ein aufwändiger 2FA-Vorgang viel Arbeitszeit kosten und für Missmut sorgen. Um die Akzeptanz des neu einzuführenden 2FA-Systems zu erhöhen, ist auf eine geringe (physische) Mehrbelastung der Anwender sehr zu achten. Durch leichte Transportierbarkeit des zweiten Faktors sowie möglichst gleichartige und wenige zweite Faktoren pro Anwender ist dies erreichbar. Die Möglichkeit einer privaten Nutzung des zweiten Faktors, ein Erscheinungsbild im LRZ-Design sowie redundante Instanzen auf Zweitgeräten o. ä. sind wünschenswerte Aspekte eines komfortablen zweiten Faktors.

Anwendungsbereiche

Anforderung	Gewichtung	Szenario
7 Anwendungsbereiche *Anforderungen aus den Einsatzszenarien 1 bis 4*	4	
7-1 An LRZ-Kennung koppelbar *Der zweite Faktor ist an die LRZ-Kennung des Anwenders koppelbar.*	4	2, 3, 4
7-2 RADIUS-Integration *Die Authentifizierung des zweiten Faktors ist in den zentralen RADIUS-Dienst integrierbar.*	4	2, 3, 4
7-3 LDAP-Integration *Die Authentifizierung des zweiten Faktors ist in den zentralen OpenLDAP-Dienst integrierbar.*	4	2, 4
7-4 Mobilgeräte *Der zweite Faktor ist von Mobilgeräten wie Smartphone oder Tablet aus einsetzbar.*	3	2, 4
7-4-1 Android *Die Eingabe des zweiten Faktors ist auf Android-Geräten möglich.*	4	2, 4
7-4-2 Blackberry *Die Eingabe des zweiten Faktors ist auf Blackberry-Geräten möglich.*	2	2, 4
7-4-3 iOS *Die Eingabe des zweiten Faktors ist auf iOS-Geräten möglich.*	4	2, 4
7-4-4 USB-Schnittstelle nicht erforderlich *Es gibt eine Möglichkeit den zweiten Faktor auch ohne USB-Schnittstelle eingeben zu können.*	4	2, 4
7-5 Fernwartung per SSH (Szenario 1) *Das 2FA-System erfüllt die Anforderungen aus Szenario 1: Fernwartung per SSH*	4	1
7-5-1 Eingabe über Kommandozeile *Der zweite Faktor lässt sich während einer Konsolensitzung über die Kommandozeile eingeben.*	3	1
7-5-2 SSH-Integration *Die Abfrage des zweiten Faktors lässt sich in den Authentifizierungsvorgang (via Username/Password) der SSH-Verbindung integrieren.*	4	1
7-5-3 UNIX-Kompatibilität *Benötigte Software zur Authentifizierung des zweiten Faktors lässt sich auf UNIX-Systemen (Debian, Ubuntu, Suse) installieren.*	4	1
7-5-4 Dezentrale Authentifizierung *Eine 2FA ist auch gegenüber UNIX-Servern möglich, die nicht an einen zentralen Authentifizierungsdienst wie RADIUS angeschlossen sind.*	1	1
7-5-5 Windows-SSH-Clients *Der zweite Faktor lässt sich mit Windows-SSH-Clients wie bspw. PuTTY nutzen.*	3	1

7-6 Telearbeit und VPN (Szenario 2) *Das 2FA-System erfüllt die Anforderungen aus Szenario 2: Telearbeit und VPN*		4	2
	7-6-1 VPN-Server Integration *Die Authentifizierung des zweiten Faktors lässt sich in den authentifizierenden Dienst des VPN-Servers integrieren.*	4	2
	7-6-2 Kompatibel mit Cisco Anyconnect Desktop-VPN-Client *Der zweite Faktor lässt sich zusammen mit der Desktop-Version des Cisco Anyconnect VPN-Clients nutzen.*	4	2
	7-6-3 Kompatibel mit weiteren gängigen Desktop-VPN-Clients *Der zweite Faktor lässt sich mit weiteren gängigen Desktop-VPN-Clients nutzen.*	3	2
	7-6-4 Kompatibel mit Cisco Anyconnect mobile VPN-Client *Der zweite Faktor lässt sich zusammen mit der mobilen Version des Cisco Anyconnect VPN-Clients nutzen.*	3	2
	7-6-5 Kompatibel mit weiteren gängigen mobilen VPN-Clients *Der zweite Faktor lässt sich mit mobilen Versionen weiterer gängiger VPN-Clients nutzen.*	2	2
7-7 Login an Arbeitsplatzrechnern (Szenario 3) *Das 2FA-System erfüllt die Anforderungen aus Szenario 3: Login an Arbeitsplatzrechnern*		4	3
	7-7-1 UNIX-Desktoploginsysteme *Die Abfrage des zweiten Faktors ist in aktuelle UNIX-Desktoploginsysteme integrierbar. (Debian, Ubuntu, Suse)*	4	3
	7-7-2 MacOS-Desktoploginsysteme *Die Abfrage des zweiten Faktors ist in aktuelle MacOS-Desktoploginsysteme integrierbar.*	4	3
	7-7-3 Windows-Desktoploginsysteme *Die Abfrage des zweiten Faktors ist in aktuelle Windows-Desktoploginsysteme integrierbar.*	4	3
	7-7-4 MS Active Directory *Die Authentifizierung des zweiten Faktors lässt sich in das MWN-ADS (Microsoft Active Directory) integrieren.*	4	3
	7-7-5 Sperrbildschirm *Das 2FA-System bietet dem Administrator die Möglichkeit nach einem Sperrbildschirm die Eingabe des zweiten Faktors nicht zu fordern.*	1	3
	7-7-6 Offline Modus *Ein Desktoplogin mittels 2FA ist (für Ausnahmefälle) auch ohne Internet-Verbindung möglich. (z.B. Laptop im Zug)*	2	
7-8 Webapplikationen (Szenario 4) *Das 2FA-System erfüllt die Anforderungen aus Szenario 4: Webapplikationen*		4	4
	7-8-1 Webbrowserkompatibilität *Die Authentifizierung des zweiten Faktors kann über einen gängigen Webbrowser erfolgen.*	4	4
	7-8-1-1 Apple Safari *Die Eingabe des zweiten Faktors ist in mobiler und Desktop-Version von Apple Safari möglich.*	3	4

	7-8-1-2 Google Chrome *Die Eingabe des zweiten Faktors ist in mobiler und Desktop-Version von Google Chrome möglich.*	3	4
	7-8-1-3 Mozilla Firefox *Die Eingabe des zweiten Faktors ist in mobiler und Desktop-Version von Mozilla Firefox möglich.*	3	4
7-8-2 Integration in Webanwendungen *Das 2FA-System stellt Schnittstellen zur Authentifizierung des zweiten Faktors in verschiedenen Programmiersprachen des Web-Kontextes und Beispielcode oder Plugins für gängige Webanwendungen bereit.*		3	4
	7-8-2-1 PHP *Der zweite Faktor lässt sich mittels PHP-Modul abfragen und authentifizieren.*	3	4
	7-8-2-2 JavaScript *Der zweite Faktor lässt sich mittels JavaScript-Modul abfragen und authentifizieren.*	3	4
	7-8-2-3 Python *Der zweite Faktor lässt sich mittels Python abfragen und authentifizieren.*	2	4
7-8-3 Föderiertes Identitätsmanagement *Die Integration des zweiten Faktors in ein föderiertes Identitätsmanagementsystem (Shibboleth) ist möglich.*		1	4
7-8-4 Lokale Authentifizierungsdienste *Auch gegenüber Webanwendungen, die nicht an den eigenständigen zentralen Authentifizierungsdienst angebunden sind, kann sich der Nutzer mit seinem zweiten Faktor authentifizieren.*		2	
7-9 Plugins für gängige Anwendungen *Plugins zur Integration der 2FA in gängige Web- und Desktopanwendungen werden bereit gestellt.*		1	

Tabelle 3.10.: Anforderungskatalog: Kategorie 7 (Anwendungsbereiche)

Gesucht wird ein zweiter Faktor zum bestehenden Authentifizierungssystem des LRZs. Dementsprechend muss sich dieser in die bestehende Nutzerverwaltung eingliedern lassen, also an die LRZ-Kennung der Anwender koppelbar sein. Da RADIUS- und LDAP-Protokoll zur Authentifizierung am LRZ zum Einsatz kommen, ist eine Kompatibilität der Abfrage des zweiten Faktors mit beiden Protokollen essentiell.

Darüber hinaus ist es für die Szenarien *7-6 Telearbeit und VPN (Szenario 2)* und *7-8 Webapplikationen (Szenario 4)* wichtig, dass der zweite Faktor auch von Mobilgeräten wie Smartphones oder Tablets, die i. d. R. über keine USB-Schnittstelle verfügen, abgefragt werden kann. Die mobilen Betriebssysteme Android und iOS stehen ihrem Verbreitungsgrad wegen hier gegenüber Blackberry im Vordergrund.

Alle Einsatzszenarien werden maximal gewichtet.

Fernwartung per SSH (Szenario 1)

Um die Authentifizierung beim SSH-Zugriff auf Linux-Server bzw. -VMs per Nutzername-/Passwort-Kombination durch einen zweiten Faktor zu stärken, ist es notwendig, dass jener sich in den SSH-Authentifizierungsvorgang integrieren lässt. Und da SSH-Zugriffe i. d. R. über die Kommandozeile durchgeführt werden, muss der zweite Faktor sich auch darüber eingeben lassen. Ebenso muss das 2FA-System hierfür kompatibel mit den auf den meisten LRZ-Servern eingesetzten UNIX-Betriebssystemen sein. Auch eine Möglichkeit zur SSH-Authentifizierung über Windows-SSH-Clients wie PuTTY[13] ist wichtig. Obwohl die meisten Server des LRZs an zentrale Authentifizierungsdienste angeschlossen sind, ist eine Unterstützung zur 2FA an den übrigen Servern wünschenswert.

Telearbeit und VPN (Szenario 2)

Da viele Dienste des LRZs nicht weltweit, sondern nur aus dem MWN heraus erreichbar sind, ist für den Zugriff von außerhalb eine VPN-Verbindung erforderlich. Bei Einsatz von 2FA muss sich der zweite Faktor demnach auch in den Aufbau einer VPN-Verbindung integrieren lassen. Ebenso ist die Kompatibilität mit dem vom LRZ unterstützten und verbreiteten SSL-VPN-Client *Cisco Anyconnect* sehr wichtig. Andere VPN-Clients werden etwas geringer gewertet. Insgesamt werden Desktopversionen ihrer Einsatzhäufigkeit wegen dabei jeweils etwas höher gewichtet als ihre mobilen Pendants.

Login an Arbeitsplatzrechnern (Szenario 3)

Mitarbeitern des LRZs werden je nach Wunsch Windows-, UNIX- oder MacOS-Desktoprechner zur Verfügung gestellt. Eine 2FA muss dementsprechend in die jeweiligen Desktoploginsysteme integrierbar sein. Zudem kommen Active Directory Services zum Einsatz, mit denen der zweite Faktor daher kompatibel sein muss.

Abhängig von Dauer bzw. Aufwand des 2FA-Vorgangs für den Mitarbeiter könnte die Forderung nach Eingabe des zweiten Faktors auch beim Sperrbildschirm (und nicht nur beim Neustart) dazu führen, dass der Desktop für kurze Arbeitsunterbrechungen wie einem Gang zur Kaffeemaschine oder Toilette nicht mehr gesperrt wird. Da dies dem durch 2FA angestrebten Ziel einer höheren Authentizität zuwider läuft, ist Wahlmöglichkeit hierin relevant und die Optionen sind vom LRZ oder den Abteilungen abzuwägen.

Webapplikationen (Szenario 4)

Um 2FA gegenüber Diensten mit Webinterface durchführen zu können, ist es nötig den zweiten Faktor über einen Webbrowser eingeben zu können. Die unter Windows, UNIX und MacOS wohl gängigsten Webbrowser werden hier gleichrangig betrachtet.

Auch wenn in dieser Arbeit hauptsächlich Anwendungen des LRZs mit zentraler Nutzerverwaltung betrachtet werden, ist es für z. B. Kleinprojekte relevant, wenn die gewählte 2FA-Technologie zum Einsatz bei lokalen Authentifizierungsdiensten geeignet ist.

Eine prinzipielle Unterstützung von Systemen zum föderierten Identitätsmanagement ist für das LRZ als Shibboleth-Identity-Provider im DFN für zukünftige Projekte wünschenswert.

[13] `https://www.putty.org/`

Nachdem nun aus den 2FA-Szenarien des LRZs Anforderungen abgeleitet, hierarchisch gruppiert in einem Katalog zusammengestellt und dem Bewertungsschema entsprechend mit Gewichten versehen wurden, kann zur Produktauswahl übergegangen werden. Der vorliegende Anforderungskatalog umfasst 109 Einzelanforderungen (Blätter des Anforderungsbaumes) in 30 Gruppen (innere Knoten).[14] Der hiermit zum Abgleich von Produkten gegen ihn erforderliche Aufwand legt eine Vorselektion nahe.

3.5. Ausschlusskriterien

Um in der folgenden Marktanalyse (siehe Kapitel *5.1 Marktanalyse*) die Anzahl der gegen den gesamten Anforderungskatalog zu prüfenden Produkte reduzieren zu können, werden hier einige Ausschlusskriterien definiert, die für ein 2FA-System am LRZ unerlässlich scheinen und sich hauptsächlich aus der Aufgabenstellung dieser Arbeit sowie den Einsatzszenarien ergeben.[15] Anhand derer werden sich derzeit gängige Technologien (Kapitel *4.1 Technologische Vorauswahl*) und am Markt verfügbare Produkte (Kapitel *5.1.1 Vorselektion*) mit geringerem Aufwand vorselektieren lassen.

Als Ausschluss- oder K.O.-Kriterien sollen gelten:

- Verbreitungsgrad
  Technologie und Produkt sind praxiserprobt und auf dem Markt verbreitet. (*1-3*)

- Vollständiges inhouse hosting möglich
  Der Aufgabenstellung dieser Arbeit entsprechend sollen keine externen Authentifizierungsdienstleister in die Authentifizierung gegenüber LRZ-Diensten eingebunden sein. Alle relevanten Komponenten müssen auf Servern des LRZs implementiert und betrieben werden können. (*1-4*)

- In bestehende AAA-Systeme integrierbar: RADIUS, LDAP, MS Active Directory
  Das 2FA-System muss mit den am LRZ eingesetzten Authentifizierungsprotokollen RADIUS, LDAP und MS Active Directory kompatibel sein. (*2-1*, *7-2*, *7-3*, *7-7-4*)

- An bestehende LRZ-Kennungen koppelbar
  Das 2FA-System muss sich in die bestehende Nutzerverwaltung eingliedern lassen und LRZ-Kennungen zweite Faktoren zuordnen können. (*2-3-1*, *3-3-2*, *7-1*)

- Aktive Wartung
  Das Produkt und insbesondere die zugehörige Verwaltungssoftware werden aktiv gewartet und weiterentwickelt. (*3-2-1*)

- Geeignete Verwaltungssoftware
  Das 2FA-System hat eine ausgereifte Verwaltungssoftware zu enthalten, die mindestens die Routineaufgaben übernehmen und so manuelle Arbeit abnehmen kann. (*3-3-1*)

[14] Eine zusammenhängende Gesamtfassung des Anforderungskataloges ist in Anhang C zu finden.

[15] Im Anforderungskatalog sind sie als diejenigen Knoten mit Gewichtung 4 der obersten zwei Ebenen (sowie deren direkte Kinder mit Gewichtung 4) zu finden, die sich mit geringem Aufwand prüfen und eine gewisse Streuung unter den Produkten erwarten lassen.

- Geltungsbereich der 2FA
  Die 2FA lässt sich pro Nutzer(-gruppe) und Dienst einzeln aktivieren. Sie erstreckt sich nicht zwangsweise auf alle mit den LRZ-Kennungen synchronisierte Benutzerverwaltungen wie z. B. die der Universitäten. (*3-3-8*)

- Für alle Anwendungsszenarien vollständig geeignet
  Technologie bzw. Produkt sind für alle Einsatzszenarien vollständig geeignet. (*7-5*, *7-6*, *7-7*, *7-8*)

Diese Ausschlusskriterien dienen als Grundlage für die Vorstufen der in Kapitel *5 Produktauswahl* folgenden Produktauswahl. Im ersten Schritt orientiert sich eine technologische Vorauswahl in Kapitel *4.1 Technologische Vorauswahl* an diesen Kriterien, um dann in einem zweiten Schritt einen Konzeptvorschlag zur 2FA am LRZ zu erarbeiten. Diesem folgend wird die Marktanalyse in Kapitel *5.1.1 Vorselektion* mit einer Vorselektion anhand der eben festgelegten Ausschlusskriterien begonnen. Der vierte Schritt der Produktauswahl in Kapitel *5.1.2 Hauptauswahl*, die Hauptauswahl bewertet anschließend alle dann noch übrigen Kandidaten gegen den gesamten Anforderungskatalog. Zuletzt werden die Bewertungen der Hauptkandidaten verglichen und in Kapitel *5.2 Produktvergleich und -auswahl* ein Produkt auf Basis des Anforderungskataloges ausgewählt.

4. Rahmenkonzept zur Mehr-Faktor-Authentifizierung am LRZ

Um die Marktanalyse im nächsten Kapitel effizienter zu gestalten und gezielter nach potentiell geeigneten Produkten suchen zu können, unternimmt dieses Kapitel eine Vorauswahl möglicher 2FA-Technologien, die sich an den Ausschlusskriterien in Kapitel *3.5 Ausschlusskriterien* und einigen weiteren Aspekten orientiert. Hieraus wird dann ein Vorschlag zum konzeptuellen Rahmen eines 2FA-Systems am LRZ abgeleitet, der der anschließenden Produktsuche zu Grunde gelegt werden wird.

4.1. Technologische Vorauswahl

Die in Kapitel *2.5 Zweite Faktoren für die Benutzerauthentifizierung mit statischen Passwörtern* vorgestellten 2FA-Technologien werden nun ihrer dortigen Reihenfolge nach durchgegangen und hinsichtlich der Ausschlusskriterien sowie weiterer Aspekte bzgl. ihrer Eignung zur Erfüllung des Anforderungskatalogs untersucht.

In der Klasse der Besitzfaktoren wurden Einmalpasswortverfahren, Challenge-Response-Verfahren und Authentifizierungsmöglichkeiten über Bilder vorgestellt.

Vorberechnete Listen an Einmalpassworten nach dem S/Key-Verfahren gelten nicht mehr als sicher. Auch wenn OPIE noch stellenweise in UNIX-Umgebungen eingesetzt wird, kann man bei vorberechneten OTP-Listen wohl nicht mehr von einer gängigen, auf dem Markt verbreiteten Technologie sprechen. Deutlich weiter verbreitet hingegen sind Verfahren zur kontinuierlichen Neuberechnung von Einmalpassworten. Der Versand von OTPs per SMS gilt seiner Anfälligkeit für Phishing wegen als nicht mehr sicher. Da hierzu zudem Handynummern aller Anwender gespeichert werden müssten, scheinen Verfahren mit SMS als Übertragungskanal ungeeignet. Zeit- oder zählerbasierte Generierung von Einmalpassworten in Hard- oder Softwaretokens könnte den Anforderungskatalog jedoch erfüllen.

Von den betrachteten Challenge-Response-Verfahren werden Smartcard-Technologien keinen Eingang in die Produktauswahl finden. Der Aufwand alle unter Verwaltung des LRZs befindlichen Terminals mit Lesegeräten auszustatten, Nutzer bei der Installation von entsprechenden Treibern auf Privatgeräten zu unterstützen, aber auch die finanzielle und physische Belastung ein entsprechendes Lesegerät anschaffen und mitführen zu müssen scheinen durchaus unverhältnismäßig und hinsichtlich des geringen Nutzungskomforts daher auch nicht lohnend.

Anders verhält es sich mit Implementierungen der Standards der FIDO-Allianz: Sicherheitsniveau, Bedienkomfort und parallele private Nutzbarkeit zeichnen diese Technologie als potentiellen Kandidaten zum Einsatz am LRZ aus.

Eine Authentifizierung mit Hilfe von Bildern wird sich für die SSH-Sitzungen über die Konsole und wegen fehlender graphischer Oberfläche nicht eignen. Eine Integration in proprietäre VPN-Clients oder Desktoplogin von Windows bzw. MacOS scheint ebenfalls schwie-

rig. Der Einarbeitungsaufwand ist verhältnismäßig groß und das erforderliche Merken der initial gewählten Bilder oder Kategorien wirkt für gelegentliche Anwender ungeeignet. Derartige Verfahren werden daher in der Produktauswahl nicht weiter verfolgt.

Aus der Klasse der einer Person anhängenden Merkmale werden keine Technologien in den Konzeptvorschlag eingehen, da Biometrie und hoheitliche Ausweisdokumente (nPA) wie oben bereits erwähnt für den Einsatz am LRZ nicht gewünscht bzw. geeignet sind.

Mobilgeräte als Träger des zweiten Faktors in den Authentifizierungsvorgang einzubeziehen scheint wegen der geringen Mehrbelastung und Kostenersparnis für Nutzer sinnvoll und wird daher weiter verfolgt. Sie könnten als OTP-Generatoren oder zum Lösen von gestellten Challenges dienen. Darüber hinaus sind die sehr benutzerfreundlichen Push-Benachrichtigungen zur Authentifizierung wünschenswert.

Die Authentifizierung am LRZ sollte also durch einen Besitzfaktor gestärkt werden. Hier wirken zeit- oder zählerbasierte Einmalpasswortverfahren in Form von Hard- oder Softwaretokens sowie Implementierungen der FIDO-Standards geeignet. Auch die Nutzung von Mobilgeräten als Träger des zweiten Faktors ist näher zu prüfen.

4.2. Konzeptvorschlag

Wie eben herausgestellt, kommen als 2FA-Technologien für das LRZ Einmalpasswortverfahren mit hard- oder softwarebasierten OTP-Generatoren, FIDO U2F-Tokens oder Mobilgeräte der Anwender zum Berechnen von Challenges oder als Mobile-Push-Faktoren in Frage.

Durch Anbindung an die zentrale LRZ-Benutzerverwaltung und Integration in RADIUS, LDAP bzw. MS Active Directory dürften sich viele Anforderungen aus den Anwendungsszenarien gleichzeitig abdecken lassen. Hierbei ist eine qualitative Software zur Verwaltung der eingesetzten zweiten Faktoren jedoch wichtig. Sollte diese neben den gewünschten auch noch weitere Arten von zweiten Faktoren unterstützen, stellt dies keinen Ausschlussgrund dar. Gesucht wird jedoch kein vollständiges Identity and Access Management (IAM) System, sondern lediglich eine 2FA-/MFA-Komponente für die bestehenden AAA-Systeme des LRZs.

OTPs, die mit dem Passwort konkateniert und in die Passwortzeile eingegeben werden, erfordern keine Anpassung der jeweiligen Anmeldemasken, was gerade bei Desktoplogins oder anderen Anwendungen relevant ist, auf deren Quellcode man i. d. R. keinen Zugriff hat. Softwaretokens wirken aufgrund ihres geringeren Aufwands für Rollout und Wartung sowie der unerheblichen physischen Mehrbelastung der Anwender vorteilhafter als Hardwaretokens. Letztere werden jedoch für Anwender benötigt, die ihre privaten Mobilgeräte nicht als OTP-Generator zur Verfügung stellen wollen. Hard- und softwarebasierte Tokens sollten also parallel einsetzbar sein.

Der Einsatz von FIDO U2F-Tokens erfordert höhere softwareseitige Voraussetzungen als der von OTPs: Browser, (Web-)Anwendung und beim Einsatz zum Desktoplogin auch das verwendete Betriebssystem müssen FIDO-kompatibel sein, was aber zumindest bei den meisten modernen Browsern der Fall ist. SSH- und VPN-Integration könnten sich jedoch schwierig gestalten.

Mobilgeräte mit Internetverbindung ermöglichen sehr nutzerfreundliche und gleichzeitig sichere Loginverfahren. Zur Authentisierung wird eine kryptographische Challenge an das Mobilgerät geschickt. Nach einer simplen Interaktion mit dem Nutzer (Push-Benachrichtigung, PIN oder Auswählen eines angezeigten Icons), wird die Challenge gelöst und das Ergebnis an den authentifizierenden Dienst gesendet.

Alle genannten Technologien kommen ohne zusätzliche Hardwarelesegeräte aus, da die meisten OTP-Tokens manuell abzutippen sind, FIDO U2F-Tokens über USB- oder NFC-Schnittstelle und CR-Apps via Internet kommunizieren. Weitere Unterschiede zwischen OTP und FIDO U2F wurden in *2.5.1 Vergleich möglicher Besitzfaktoren* ausführlich diskutiert. Die Möglichkeit der Erweiterung zur kontext- oder risikobasierten Multi-Faktor-Authentifizierung kann für künftige Anforderungen offengehalten werden.

Prinzipiell scheinen diese Technologien für den Einsatz am LRZ geeignet. Abwägungen zwischen stellenweise konträren Anforderungen wie bspw. Sicherheitsniveau und Bedienkomfort werden durch die gewählten Gewichtungen im Katalog bzw. die entsprechenden Bewertungen des Produkts behandelt. Nun gilt es nach konkreten Implementierungen dieser Technologien zu suchen, die den gesamten Anforderungskatalog bestmöglich erfüllen.

5. Produktauswahl

Um ein den Anforderungen des LRZs entsprechendes 2FA-System zu finden, werden nun auf dem Markt verfügbare Produkte[1] gesucht, bewertet und schließlich miteinander verglichen.

Im ersten Schritt dieser Marktanalyse wurde den Ergebnissen der technologischen Vorselektion im vorangegangenen Kapitel entsprechend eine Liste an Produkten erstellt, die Einmalpasswortverfahren, FIDO-Standards oder CR-Techniken auf Mobilgeräten implementieren.

Quellen dieser Kandidatenliste waren hauptsächlich freie Internetrecherche mit verschiedenen Suchmaschinen (*Google*[2], *DuckDuckGo*[3], *Bing*[4]), Erwähnungen in der verwendeten Sekundärliteratur, Einträge in Foren oder Fachartikeln, aber u. a. auch Suchtreffer, die auf FAQs der Anbieterseiten verwiesen. Ein Großteil der Kandidaten wurde über Vergleichsplattformen für Softwareprodukte wie u. a. *g2crowd.com*[5] oder *Gartner*[6] gefunden. Angaben von 2FA-Hardware-Produzenten wie bspw. *Yubico*[7] zur Kompatibilität mit 2FA-Software anderer Anbieter waren eine weitere wichtige Quelle. Als bei dieser Recherchewiese keine neuen Kandidaten mehr hinzukamen, wurde die Kandidatenliste am 23.09.2018 geschlossen. Insgesamt wurden hierbei 75 Produkte gefunden und betrachtet. Ein Anspruch auf Vollständigkeit dieser Liste kann jedoch nicht erhoben werden.

Unter *5.1.1 Vorselektion* werden dann diejenigen Produkte der Kandidatenliste betrachtet, die während der Vorselektion aufgrund von Ausschlusskriterien verworfen wurden. Im dritten Schritt bewertet *5.1.2 Hauptauswahl* die noch übrigen Kandidaten (Hauptkandidaten) gegen den gesamten Anforderungskatalog. Zuletzt werden die Kandidaten der Hauptauswahl anhand des Bewertungsschemas aus *3.3 Bewertungsschema* miteinander verglichen und das geeignetste Produkt gewählt.

5.1. Marktanalyse

Die Marktanalyse besteht aus einer Vorselektion und einer Hauptauswahl.

5.1.1. Vorselektion

Im Folgenden werden nun diejenigen Kandidaten aufgeführt, die aufgrund eines der in Kapitel *3.5 Ausschlusskriterien* definierten Ausschlusskriterien nicht in die Hauptauswahl aufgenommen wurden. Die Darstellung erfolgt tabellarisch in wertungsfreier alphabetischer

[1] Unter „Produkt“ wird hier eine konkrete Implementierung einer 2FA-Technologie verstanden, die nicht notwendigerweise kommerziellen Interessen folgt.
[2] `https://www.google.com/`
[3] `https://duckduckgo.com/`
[4] `https://www.bing.com/`
[5] `https://www.g2crowd.com/categories/multi-factor-authentication`
[6] `https://www.gartner.com/`
[7] `https://www.yubico.com/`

Reihenfolge unter Angabe von Hersteller, Produktname, Produkt- bzw. Anbieterwebseite, Versionsnummer, Datum der letzten Prüfung, sowie einer knappen Beschreibung der Ausschlussgründe.

Betrachtet wurden jeweils nur die zum Zeitpunkt der letzten Prüfung auf der Homepage als aktuell präsentierten Versionen der Produkte.[8] Informationen über die Produkte entstammen hauptsächlich den angegebenen Herstellerhomepages, den dort (nicht immer leicht) zu findenden technischen Dokumentationen, Whitepapers und in Einzelfällen Telefonaten mit dem technischen Support des Herstellers.

Tabelle 5.1 führt nun die 71 der insgesamt 75 geprüften Produkte auf, die nicht in die Hauptauswahl aufgenommen wurden.

#	Hersteller: **Produktname** (Versionsnummer) *Datum der letzten Prüfung* `Produktlink` Ausschlussgründe
1	Akku: **Control Your Cloud** *27.09.2018* `https://www.akku.work/` Nur für Web-Applikationen
2	Auth0 Inc.: **Auth0** (16999.148) *27.09.2018* `https://auth0.com/` Identity Provider. SDK für Developer
3	AuthLite, LLC: **AuthLite** *27.09.2018* `http://www.authlite.com/` Keine Unterstützung von SSH oder Linux-Desktoploginsystemen. MacOS nur mit Workaround
4	Authlogics: **Authlogics Authentication Server** *27.09.2018* `https://authlogics.com` Keine Unterstützung von SSH, MacOS- oder Linux-Desktoploginsystemen
5	Axiad IDS: **Axiad ID Cloud** *27.09.2018* `https://www.axiadids.com/` Nur SaaS, kein on-premise
6	Bluink Identity: **Bluink Identity** (web 4.1, iOS 8.5, Android 4.11) *27.09.2018* `https://bluink.ca/` Eher Passwortmanager mit Autofill-Funktion. Keine Unterstützung des RADIUS-Protokolls
7	Buckhill Ltd.: **AuthStack + MFAStack** *27.09.2018* `https://www.buckhill.co.uk/products/authstack/` SSO für Web-Applikationen
8	CensorNet: **Adaptive User Authentication** *27.09.2018* `https://www.censornet.com/products/multi-factor-authentication/` Erweitert smspasscode.com um kontextbasierte Authentifizierung. SMS-OTP an SessionID gekoppelt. Keine Unterstützung von Desktoploginsystemen

[8] Nicht zu allen Produkten konnte eine Versionsnummer angegeben werden, da diese teilweise nicht mit vertretbarem Aufwand (<15min für allein die Suche nach der Versionsnummer) auf der zugehörigen Homepage bzw. technischen Dokumentation zu finden war. Dies war häufig bei Produkten der Fall, die auch als SaaS angeboten werden.

9	Centrify: **Adaptive Multi Factor Authentication** *27.09.2018* `https://www.centrify.com/products/application-services/#adaptive_multi_factor_authentication` Cloud-basiert. Nutzerdaten verbleiben aber im lokalen Active Directory und nur Identifier wird synchronisiert. Keine Unterstützung von MacOS-Desktoploginsystemen
10	Cifrasoft Ltd: **SoundLogin** *27.09.2018* `https://www.soundlogin.com/` Anstatt das OTP abzutippen wird es per Sound-Modulation an ein Browseraddon übertragen. Nur Web-Applikationen. Wird nicht mehr aktiv gewartet. Keine Verzeichnisdienste
11	Dominik Reichl: **KeePass** *27.09.2018* `https://keepass.info/` Passwortmanager mit 2FA und Autofill ausschließlich für Windows-Desktoplogin
12	Duo Security: **DUO** *27.09.2018* `https://duo.com/` Trotz großem Funktionsumfang nicht vollständig on-premise möglich
13	ESET, spol. s r.o.: **Secure Authentication Server** (2.7.32.0) *27.09.2018* `https://www.eset.com/us/business/endpoint-security/two-factor-authentication/` Als 2F SMS und Push (mit Zwischenschritt über Herstellerserver) möglich. SSH und MacOS-Desktoplogin via PAM möglich, aber laut Support mit einigem Aufwand verbunden.
14	Early Warning: **Authentify** *27.09.2018* `https://www.earlywarning.com/authentify.html` Out-Of-Band-Authentifizierung in Zusammenarbeit mit dem Mobilfunknetzbetreiber. Übertragungskanal SMS
15	Entrust: **IdentityGuard** *27.09.2018* `https://www.entrustdatacard.com/products/authentication/entrust-identityguard` Keine Unterstützung von RADIUS, SSH, MacOS- oder Linux-Desktoploginsystemen
16	Feitian: **Authentication Server & CloudIDentify** *27.09.2018* `http://www.ftsafe.com` SDK zur Integration von FIDO und OTP in Web und mobile Apps
17	FreeIPA Project: **freeIPA** (4.5.3) *27.09.2018* `https://www.freeipa.org` Eine integrierte Identity- und Authentifizierungslösung für lediglich Linux/UNIX-Umgebungen
18	Gemalto/SafeNet: **SafeNet Authentication Manager (SAM)** (9.0 (SP2)) *27.09.2018* `https://safenet.gemalto.com/` In der technischen Dokumentation keine Angaben bzgl. Desktoplogin auffindbar. Aussage des Telefonsupports: „mit OTP wahrscheinlich möglich“
19	Green Rocket Security Inc.: **GreenRADIUS** (3.0.1.1) *27.09.2018* `http://www.greenrocketsecurity.com/`

	RADIUS Server, mit integrierter 2FA. Würde die LRZ-RADIUS Installation ersetzen.
20	HID Global Corporation/ASSA ABLOY AB: **ActivID Authentication Server** *27.09.2018* `https://www.hidglobal.com/` Desktoplogin via PKI (Smartcard oder USB). Keine Integration in SSH
21	Identity Automation: **RapidIdentity Platform** *27.09.2018* `https://www.identityautomation.com` Große Plattform für Identy Governance und Lifecycle. MFA für Endpoints nur eine kleine Komponente davon. Zu groß und damit zu teuer. Keine Unterstützung von Linux-Desktoploginsystemen
22	Imprivata: **Imprivata** *27.09.2018* `https://www.imprivata.de/` Auf den medizinischen Bereich bzw. Krankenhäuser fokussiert.
23	InAuth, Inc.: **InAuthenticate** *27.09.2018* `https://www.inauth.com` Auf Web- und Mobil-Applikationen beschränkt.
24	Kaseay ltd.: **AuthAnvil** (5.6) *27.09.2018* `https://authanvil.com/` Keine Unterstützung von MacOS-Desktoploginsystemen
25	MI-Token: **MI-Token** *27.09.2018* `https://www.mi-token.com/about/` Keine Unterstützung von SSH, MacOS- oder Linux-Desktoploginsystemen
26	MePIN: **MePIN** *27.09.2018* `https://www.mepin.com` Lediglich SDK zur Integration von FIDO und OTP in Web und mobile Apps
27	MicroStrategy: **Usher Security** (10.11) *27.09.2018* `https://www.usher.com` Anderer Fokus: Ersetzt physische Badges, Passwörter und Tokens durch digitale Badges auf dem Smartphone. LDAP und Active Directory nur als Importquellen für das Usher Netzwerk. Bietet keine MFA für Active Directory.
28	Microfokus / NetIQ: **Advanced Authentication** (6.0) *27.09.2018* `https://www.netiq.com/products/advanced-authentication/` Der Support war nicht bereit technische Auskünfte zu geben oder Angaben zu Kosten zu machen, ohne vorher ein konkretes Projekt gestartet oder die Produktentscheidung getroffen zu haben. Erschwerte die Produktauswahl. Veraltetes App-Design stellt ein schlechtes Indiz dar. Ansonsten sehr umfangreiche Suite
29	Microsoft: **Azure Multi-Factor Authentication** *27.09.2018* `https://docs.microsoft.com/de-de/azure/active-directory/authentication/concept-mfa-howitworks` MFA für Azure-Produkte. Keine Unterstützung von RADIUS
30	Mideye AB: **Mideye** *27.09.2018* `https://www.mideye.com/` nicht vollständig on-premise möglich
31	NetKnights GmbH: **privacyIDEA** (2.23.2) *27.09.2018* `https://www.privacyidea.org/`

	Kostenloses open-source Tool. Ein Fork von LinOTP. Derzeit kein fertiger MacOS-Desktoplogin-Client. Laut Support könnte man aber „prinzipiell das privacyidea_pam Modul verwenden. Allerdings wären hierfür entsprechende Projektzeiten aufzubringen.“
32	Okta: **Okta** *27.09.2018* `https://www.okta.com/` On-premise-hosting nur teilweise möglich. Okta-Server sind in den Authentifizierungsvorgang immer involviert.
33	One Identity (Quest Software Inc.): **Starling Two-Factor Authentication** (September 2018) *27.09.2018* `www.oneidentity.com/products/starling-two-factor-authentication/` Nur als SaaS verfügbar. Nicht on-premise
34	OneLogin: **OneLogin** *27.09.2018* `https://www.onelogin.com/` „Single, Secure Portal to Access All Apps“ : Account-/Passwortmanager zum SSO. Kein SSH oder Linux-Desktoplogin möglich
35	OneSpan (früher vasco): **MYDIGIPASS.COM** *27.09.2018* `http://www.mydigipass.com` Nur SaaS, kein on-premise
36	PasswordWrench: **2-Factor Authentication** *27.09.2018* `www.passwordwrench.com` Passwortmanager. Für 2FA wird jedem Nutzer eine individuelle ausgedruckte 16x16 oder 16x34 Matrix an alphanumerischen Zeichen zugeteilt. Als zweiter Faktor werden dann Einträge dieser Matrix gefordert. Keine Unterstützung des RADIUS-Protokolls
37	Ping Identity: **PingID** *27.09.2018* `https://www.pingidentity.com` Als zentrale Authentifizierungsauthorität konzipiert. Würde das bisherige AAA-System ersetzen
38	PistolStar Inc.: **PortalGuard** (5.6.4.2) *27.09.2018* `https://www.portalguard.com/` Bietet lediglich Authentifizierung gegenüber deren SSO-Portal im Branding des Kunden
39	Privakey: **Privakey** *27.09.2018* `http://www.privakey.com` Lediglich SDK zur passwortfreien Authentifizierung
40	Protectismus: **Protectismus** *27.09.2018* `https://www.protectimus.com` Keine Informationen bzgl. MacOS und Linux Desktoplogin
41	RSA: **RSASecurID Suite (Authentication Manager)** (8.3) *27.09.2018* `https://www.rsa.com/en-us/products/rsa-securid-suite` Keine Information bzgl. MacOS und on-premise hosting auffindbar. Innerhalb einer Woche keine Antwort des Supports erhalten.
42	Redmine: **OTPme (made easy)** *27.09.2018* `https://www.otpme.org`

	Auch wenn einige aktuelle Tickets bearbeitet wurden, scheint die Software nicht aktiv gewartet und noch nicht vollständig produktionsreif. Beta-Status. Die Downloads sind drei Jahre alt.
43	Rublon: **Rublon** (2.1.1) *27.09.2018* `https://rublon.com/` 2FA für Wordpress- oder Atlassian-Produkte
44	SAASPASS Inc.: **SAASPASS** *27.09.2018* `https://saaspass.com/` Sehr umfangreiches Produkt. Zusätzliche Features wie Enterprise SSO, MFA für IoT-Devices, Öffnung physischer Türen etc. aber für die Anwendungsszenarien dieser Arbeit nicht erforderlich. On-premise-Preise entsprechend den Cloud-hosted-Preisen zzgl. Installationskosten. Produkt damit sehr teuer
45	SafeJKA SRL: **ROHOS** (3.5) *27.09.2018* `https://www.rohos.com/` SSH, Linux-Desktoplogin und RADIUS nicht unterstützt
46	SearchGuard: **SearchGuard** (6.x-23) *27.09.2018* `https://search-guard.com` 2FA für Elasticsearch-Cluster (Elastic Stack)
47	SecSign Technologies: **SecSign** *27.09.2018* `https://www.secsign.com/` Sehr umfangreiche Suite. Scheint aber außer Mobile-Push über die Smartphone-App des Herstellers keine anderen Arten von 2FA zu unterstützen. Keine Alternative für Mitarbeiter ohne Smartphone
48	SecurEnvoy Ltd: **SecurEnvoy** (9.3.502) *27.09.2018* `https://www.securenvoy.com/` „Tokenless 2FA“ : OTP nicht aus künstlichem Seed, sondern aus Daten (SIM, IMEI) des Smartphones generiert. RADIUS derzeit nur eingeschränkt. Kein fertiger MacOS-Desktoplogin-Client. Der Hersteller wäre aber bei Auftragsvergabe bereit einen solchen für das LRZ zu entwickeln.
49	SecureMetric Technology: **CENTAGATE (Centralized Authentication Gateway)** *27.09.2018* `https://www.centagate.com/home-centagate/` Viele Links auf der Homepage führen noch ins Leere. Stellenweise „Coming soon“ (z.B. beim pricing). Das Tool scheint noch nicht ganz marktreif und liefert auch keine technische Dokumentation
50	SecurePass: **SecurePass** *27.09.2018* `http://www.secure-pass.net` Der Support wurde am 17. August 2017 eingestellt.
51	Solidpass: **Solidpass** (Stand: 2011) *27.09.2018* `http://www.solidpass.com` Nicht aktiv gewartet. Letzter Blog-Eintrag vom 07.09.2011, Abbildungen auf der Homepage sichtbar veraltet.
52	Sophos Technology GmbH: **Sophos MCS** (4.17) *27.09.2018* `https://www.mcs.sophos.com/produkte/zwei-faktor-authentifizierung/` Nur als Cloud-Service und ausschließlich mit SMS-OTP als zweiten Faktor verfügbar
53	Swiss SafeLab GmbH: **M.ID Server** *27.09.2018*

	https://www.swiss-safelab.com/en-us/products/midserver/productdescription.aspx Keine Unterstützung von SSH, MacOS- oder Linux-Desktoploginsystemen. OTP-App nicht im Playstore verfügbar
54	Swivel Secure: **AuthControl Sentry** *27.09.2018* https://swivelsecure.com/ Keine Unterstützung von SSH oder MacOS-Desktoploginsystemen
55	Symantec: **VIP und VIP Access Manager** (9.8.2) *27.09.2018* https://www.symantec.com/products/validation-id-protection Keine Unterstützung von Desktoploginsystemen
56	ThreatMetrix: **ThreatMetrix** (Release Sommer 2018) *27.09.2018* https://www.threatmetrix.com/ Digital Identity Network. ThreatMetrixID = globaler Identifier pro User, generiert aus verschiedenen Attributen. Globale Datenbank. Ziel: Nutzer eindeutig erkennen. Nicht als zusätzliche 2FA Komponente konzipiert. Nicht on-premise
57	Token2: **Token2** (0.2.3) *27.09.2018* https://www.token2.com/ Auf Web-Applikationen fokussiert
58	TokenOne: **TokenOne** *27.09.2018* https://www.tokenone.com Cloud gehostetes System, das PINs über eine KeyMap auf Chars abbildet und diese als OTP nutzt. Unterstützt nur Web-Applikationen
59	Twizo: **Twizo** (1.0.7) *27.09.2018* https://www.twizo.com/ Scheint nicht on-premise möglich, Produkt besteht nur aus einer API
60	UNLOQ Systems LTD.: **UNLOQ** *27.09.2018* https://unloq.io/ Passwortfreie Authentifizierung über HTTP-API. Auf Web-Kontext fokusiert
61	Univention GmbH: **Univention Corporate Server** (4.3-2) *27.09.2018* https://www.univention.de/ Identity- und Infrastrukturmanagementsystem. Zur 2FA kommt privacyIDEA zum Einsatz.
62	WSO2: **WSO2 Identity Server** (5.3.0) *27.09.2018* https://wso2.com/identity-and-access-management/ Nur für Web-Applikationen und Windows-Desktoplogin (via Kerberos) geeignet; unterstützt das RADIUS-Protokoll nicht.
63	WatchGuard: **AuthPoint** *27.09.2018* https://www.watchguard.com/wgrd-products/multi-factor-authentication Nicht vollständig on-premise möglich. Keine Unterstützgung für SSH oder Linux-Desktoplogin
64	Wheelsystems: **WHEEL CERB AS** *27.09.2018* https://www.wheelsystems.com Keine Informationen bzgl. MacOS- oder Linux-Desktoplogin auffindbar
65	WiKID Systems, Inc: **WiKID** (4.2.0-b2032) *27.09.2018* https://www.wikidsystems.com Keine Unterstützung von Desktoploginsystemen

66	Yubico: **Yubico** `https://www.yubico.com/` Angebotene Lösungen zur Yubico OTP und U2F Validierung stellen keine vollständige Verwaltungssoftware, sondern nur eine API dar.	*27.09.2018*
67	entersekt: **Transakt** `https://www.entersekt.com` Auf Bankwesen beschränkt	*27.09.2018*
68	i-Sprint Innovations: **AccessMatrix UAS** `http://www.i-sprint.com/` Keine Angaben bzgl. Desktoplogin. Scheint auf WebApps und VPN beschränkt.	*27.09.2018*
69	mSIGNIA: **Multi-Factor Authentication Software** `https://msignia.com/` Fokusiert auf online Payment Vorgänge. Keine Unterstützung von Verzeichnisdiensten	*27.09.2018*
70	sms passcode: **Adaptive Multi-factor Authentication** `http://www.smspasscode.com` Als zweiter Faktor ist nur der SMS-Versand verfügbar.	*27.09.2018*
71	wwpass: **wwpass** `http://www.wwpass.com` Nicht vollständig on-premise möglich. PassKey werden über Server von wwpass.com authentifiziert.	*27.09.2018*

Tabelle 5.1.: Kandidaten, die die Vorselektion nicht bestanden

Sehr viele der Produkte bieten eine Zwei-Faktor-Authentifizierung für Webapplikationen und VPN-Zugänge. Ebenso ist auch häufig eine Schnittstelle für Microsoft Active Directory bzw. LDAP und RADIUS vorhanden. 2FA-Desktoplogin für MacOS oder Linux scheint jedoch nicht weit verbreitet. Da OTP-Verfahren älter und somit etablierter als bspw. die FIDO-Standards sind, ließen sich auch mehr OTP-Produkte auf dem Markt finden, die i. d. R. mit den OATH-Standards arbeiten. Kontext- oder risikobasierte MFA scheint noch nicht weit verbreitet. Produkte mit solchen Eigenschaften werben jedoch vordringlich damit. Ähnliches gilt für Mobile-Push als zweiten Faktor. Etwa ein Drittel der betrachteten Produkte bieten SAML und SSO-Funktionen. Womöglich zeichnet sich auch ein Aufkommen von Identity as a Service (IDaaS) ab.

Diese während der Vorselektion gewonnenen Eindrücke können jedoch nicht als vollständig belastbar gelten, da die Recherche eines Produktes i. d. R. aus Effizienzgründen abgebrochen wurde, sobald ein valider Ausschlussgrund vorlag.

4 dieser 71 Kandidaten schieden in der Vorselektion nur knapp aus. Sie sind in Tabelle 5.2 aufgeführt. Eine Folgearbeit zur Validierung der vorliegenden Produktbewertung könnte auch die dann neueren und vielleicht erweiterten Versionen derer betrachten.

#	Hersteller: **Produktname** (Versionsnummer) *Datum der letzten Prüfung* `Produktlink` Ausschlussgründe
1	Microfokus / NetIQ: **Advanced Authentication** (6.0) *27.09.2018* `https://www.netiq.com/products/advanced-authentication/` Der Support war nicht bereit technische Auskünfte zu geben oder Angaben zu Kosten zu machen, ohne vorher ein konkretes Projekt gestartet oder die Produktentscheidung getroffen zu haben. Erschwerte die Produktauswahl. Veraltetes App-Design stellt ein schlechtes Indiz dar. Ansonsten sehr umfangreiche Suite
2	NetKnights GmbH: **privacyIDEA** (2.23.2) *27.09.2018* `https://www.privacyidea.org/` Kostenloses open-source Tool. Ein Fork von LinOTP. Derzeit kein fertiger MacOS-Desktoplogin-Client. Laut Support könnte man aber „prinzipiell das privacyidea_pam Modul verwenden. Allerdings wären hierfür entsprechende Projektzeiten aufzubringen.“
3	SAASPASS Inc.: **SAASPASS** *27.09.2018* `https://saaspass.com/` Sehr umfangreiches Produkt. Zusätzliche Features wie Enterprise SSO, MFA für IoT-Devices, Öffnung physischer Türen etc. aber für die Anwendungsszenarien dieser Arbeit nicht erforderlich. On-premise-Preise entsprechend den Cloud-hosted-Preisen zzgl. Installationskosten. Produkt damit sehr teuer
4	SecurEnvoy Ltd: **SecurEnvoy** (9.3.502) *27.09.2018* `https://www.securenvoy.com/` „Tokenless 2FA“ : OTP nicht aus künstlichem Seed, sondern aus Daten (SIM, IMEI) des Smartphones generiert. RADIUS derzeit nur eingeschränkt. Kein fertiger MacOS-Desktoplogin-Client. Der Hersteller wäre aber bei Auftragsvergabe bereit einen solchen für das LRZ zu entwickeln.

Tabelle 5.2.: Kandidaten, die die Vorselektion nur knapp nicht bestanden

5.1.2. Hauptauswahl

Folgende 4 der betrachteten 75 Produkte haben es durch die Vorselektion geschafft (Tabelle 5.3) und sind daher gegen den gesamten Anforderungskatalog zu testen. Nach einer kurzen Vorstellung jedes Produktes, wird der Anforderungskatalog in gleicher Reihenfolge wie in der *3.4 Erstellung des Anforderungskataloges* durchgegangen und die 4 Produkte bzgl. der jeweiligen Anforderungen dem *3.3 Bewertungsschema* gemäß gewertet. Berechnete Produktbewertungen sind auf zwei Nachkommastellen kaufmännisch gerundet angegeben.

Die hierfür nötige Zuordnung von Erfüllungsgraden wird aus Gründen der Lesbarkeit und des Umfangs dieser Arbeit nicht formal vollständig für jede Anforderung und jedes Produkt begründet. Die Ausführungen konzentrieren sich eher auf Ausreißer und Unterschiede zwischen den Anforderungserfüllungen durch die verschiedenen Produkte.

Die Bewertungszahlen der Gesamtbewertung werden vermutlich nicht stark divergieren, da fast alle maximal gewichteten Anforderungen als Ausschlusskriterien für die Vorselektion herangezogen wurden. Die danach verbliebenen Hauptkandidaten erfüllen diese Anforderungen also voll. Die meisten weniger gewichteten Anforderungen werden von allen Produkten etwa gleich gut erfüllt oder gehen aufgrund ihres niedrigeren Gewichts nicht stark in die Gesamtbewertung ein. Manche der Anforderungen des Kataloges zielen auch eher auf eingesetzte 2FA-Technologien ab. Da alle Produkte jedoch mehrere 2F-Arten anbieten und (bis auf Produkt *Defender*) alle im *4.2 Konzeptvorschlag* erarbeiteten beinhalten, werden die Kandidaten sich in solchen Anforderungen kaum unterschieden.

Wenn nicht anders gekennzeichnet referieren kursiv gedruckte Nummern in diesem Kapitel i. d. R. auf Anforderungsnummern und nicht auf Kapitel. Zur besseren Unterscheidung von Kapitel- oder Abbildungsnummern wird für Anforderungsnummern ein '-' als Trennzeichen verwendet.

#	Hersteller: **Produktname** (Versionsnummer) *Datum der letzten Prüfung* `Produktlink` Ausschlussgründe
72	KeyIdentity GmbH: **LinOTP** (2.10.1.1) *21.10.2018* `https://www.linotp.org`
73	One Identity (Quest Software Inc.): **Defender** (5.9.0) *21.10.2018* `https://www.oneidentity.com/products/defender/`
74	RCDevs SA: **OpenOTP MFA Suite** (1.4.1-1) *21.10.2018* `https://www.rcdevs.com/products/openotp/`
75	SecureAuth Corporation: **SecureAuth IdP + core security** (9.2) *21.10.2018* `http://www.secureauth.com`

Tabelle 5.3.: Kandidaten der Hauptauswahl

Diese 4 Kandidaten der Hauptauswahl werden nun kurz vorgestellt.

KeyIdentity GmbH: LinOTP

Die deutsche *KeyIdentity GmbH* bietet skalierbare IAM-Lösungen mit Fokus Multi-Faktor-Authentifizierung auf open-source-Basis an. Deren Produkt *LinOTP* ist open-source (AGPLv3 und GPLv2) und kostenlos verfügbar. Hiermit lassen sich einige OTP-Szenarien abdecken. Kostenpflichtig sind Enterprise-Features (z. B. Garantie und Support), Identityprovider für den Desktoplogin (KeyIdentity Authentication Provider) sowie die Nutzung der Authentifizierung via Push-Benachrichtigungen (KeyIdentity Authenticator). Als KeyIdentity Smart Virtual Appliance werden auch vorkonfigurierte Instanzen für verschiedene Virtualisierungsplattformen vertrieben. Der Desktoplogin ist nur im (vollumfänglichen) Enterprise-Lizenzpaket enthalten.

LinOTP (bzw. die gesamte *KeyIdentity GmbH* MFA Suite, die hier in *LinOTP*, Version 2.10.1.1 betrachtet werden soll) wird vom Hersteller als gut skalierbar und Dank modularem Aufbau sowie offener API als herstellerunabhängig bezeichnet. Der „API first“-Ansatz erlaubt eine Integration in beinahe jede eigene Anwendung. Die ausführliche Dokumentation beschreibt Automatisierung des Rollouts, User-Self-Service- sowie Helpdesk-Funktionen. Unterstützt werden des Weiteren SAML- und OpenID-Protokoll. Als zweite Faktoren sind OATH HOTP/TOTP (als Hard- oder Software), OTP-YubiKey, U2F, SMS, Email und Voice-Call möglich; mit der KeyIdentity Authenticator-App auch QR-Codes und Mobile-Push.

LinOTP kann auf UNIX-Betriebssystemen gehostet werden. Als Datenspeicher der Token-Seeds sind verschiedene Datenbanktypen möglich. Die Einrichtung erfolgt UNIX-typisch über .config-Files.

One Identity (Quest Software Inc.): Defender

One Identity stellt die Identity-Sparte der *Quest Software Inc.* dar, welche von *Gartner* im Februar 2018 (vor *SecureAuth Corporation*) in den Quadrant der Leader eingeordnet wurde.[one18] In ihrer Produktfamilie von IAM-Lösungen befinden sich u. a. die MFA-Suites *Starling* (IDaaS) und *Defender* (on-premise).

Schwerpunkt der Suite scheint auf der Interaktion mit Microsoftprodukten, v. a. einem Active Directory zu liegen. Neben der web-basierten Administration in modernem Design bietet *Defender* 5.9.0 auch einige Self-Service- und Helpdesk-Funktionen. Eine Automatisierung von Managementtasks ist über die Defender Management Shell (basierend auf WinPowerShell) möglich. Der für die Pflege der Dokumentation und Wissensdatenbank erbrachte Aufwand wird u. a. an den reichlichen aktuellen Anleitungsvideos sichtbar. *Defender* bietet keine Mobile-Push-Authentifizierung, sondern OTP (Hard- und Software), Email, SMS, YubiKey, Symantec VIP als zweiten Faktor. Der Funktionsumfang der zugehörigen Smartphone-App scheint beschränkt.

Die Stärken von *Defender* liegen wohl in einer sehr engen Integration in Active Directory. Da hierin auch die Token-Seeds abgelegt werden, ist letztlich für jeden Anwender ein AD-Konto nötig. Verschiedene User-Importfunktionen aus anderen Quellen werden mitgeliefert. *Defender* ist auf Windows-Maschinen zu installieren und über Klicks in GUIs zu konfigurieren.

RCDevs SA: OpenOTP MFA Suite

Das luxemburgische Unternehmen *RCDevs SA* hat sich auf MFA-Lösungen spezialisiert und setzt in seiner Entwicklung auf offene Standards. Zentrale Managementschnittstelle der *OpenOTP MFA Suite* ist WebADM (vermutlich eine Abkürzung für „WebAdmin“), das LDAP-Verzeichnisse vereint und die einzelnen Container (z. B. OpenOTP Server, SMSHub, ...) sowie Webanwendungen (Self-Service-Funktionen etc.) hostet. Die Komponente TiQR ermöglicht eine Challenge-Response-Authentifizierung mittels PKI anhand von QR-Code-Scans. Die Dokumentation scheint gut und neben SAML und OpenID stehen Plug-Ins für verschiedene Webanwendungen wie bspw. WordPress zur Verfügung.

Als mögliche zweite Faktoren werden OATH OTP, U2F, Mobile-Push, YubiKeys, QR-Scan, PKI, SMS und Email angeboten. *RCDevs SA* verteibt auch eine große Zahl an Hardwaretokens. *OpenOTP MFA Suite* läuft auf Linux-Maschinen (die vorkonfigurierte VM basiert auf CentOS 7) und wird entsprechend über einige .config-Files eingerichtet.

SecureAuth Corporation: SecureAuth IdP + core security

SecureAuth Corporation bezeichnen sich als ein „leader in access management, identity governance, and penetration testing“[9] und werden in den *Gartner peer insights* hoch bewertet. [gar18]

Mit *SecureAuth IdP + core security* bieten sie eine ausführlich dokumentierte Plattform in modernem Design, die Authentifizierung, SSO und Zugangskontrolldienste in sich vereinen kann. Über die 2FA mittels SMS, OATH OTP, YubiKey, Symantec VIP oder Mobile-Push setzen sie auf den Einsatz von risikobasierter MFA. SSO und Self-Service-Funktionen sollen Nutzungskomfort erhöhen und den Helpdesk entlasten.

Die Integration dieses umfangreichen Tools in andere Anwendungen erfolgt hauptsächlich via SAML. *SecureAuth IdP + core security* ist auf Windows-Maschinen zu installieren und über Klicks in GUIs zu konfigurieren.

Herstellervergleich

SecureAuth Corporation und *One Identity (Quest Software Inc.)* sind wohl die beiden größeren der 4 Hersteller und werden in Sachen Support und Zukunftssicherheit punkten können.

Jedes Produkt ermöglicht 2FA für Desktoplogin, VPN-Verbindungen, SSH-Sitzungen sowie Web-Applikationen, ist on-premise hostbar und mit RADIUS, SAML, LDAP und Active Directory kompatibel.

LinOTP bietet 2FA für Web-Applikationen unabhängig von der Anwenderzahl kostenlos und eröffnet somit die spannende Option auch Web-Portal-Zugänge von Studierenden kosteneffizient zu stärken. An zweiten Faktoren unterstützen alle 4 Kandidaten OATH-OTP, SMS, den Einsatz von YubiKeys und (mit Ausnahme von *Defender*) Mobile-Push.

Anhand der Ausschlusskriterien lässt sich hier keinem Kandidaten der Vorzug geben. Durch Bewertung gegen den vollständigen Anforderungskatalog wird nun eine feingranularere Entscheidung getroffen.

Beim folgenden Durchlauf erfolgt die Darstellung des Anforderungskataloges in kompakter Form mit Anforderungsnummer, -titel, Gewichtung und je einer Spalte für die Bewertungs-

[9] `https://www.secureauth.com/company/about-us`

zahl des Produktes. Beschreibung und Begründung der Anforderung kann in *3.4 Erstellung des Anforderungskataloges* nachgelesen werden.[10]

Aufgrund mangelnder Dokumentation der Hersteller, unverhältnismäßigem (Recherche-) Aufwand oder stellenweise fehlendem Detailwissen gelang es nicht, jede Anforderung für jedes Produkt zu bewerten. Solche Fälle wurden an entsprechender Stelle kommentiert. Tabelle 5.4 zeigt die Anzahl nicht bewerteter Einzelanforderungen pro Produkt. Nicht ausgefüllte Einzelanforderungen werden mit Punktzahl 0 gewertet und in den Tabellen mit „???“ markiert.

Produkt	nicht bewertete Einzelanforderungen	
LinOTP	5 / 109 =	4.6%
Defender	5 / 109 =	4.6%
OpenOTP MFA Suite	5 / 109 =	4.6%
SecureAuth IdP + core security	6 / 109 =	5.5%

Tabelle 5.4.: Anzahl nicht bewerteter Einzelanforderungen pro Produkt

Falls nicht anderweitig gekennzeichnet, basieren Aussagen über die Eigenschaften der Produkte auf Angaben der jeweiligen Herstellerhomepages und dort zum Zeitpunkt der letzten Prüfung zu findenden Dokumenten.

[10] Erinnerung: Dem Bewertungsschema gemäß gilt eine Anforderung als nicht erfüllt (0 Punkte), falls eine essentielle Teilanforderung (Gewichtung 4) nicht voll (<3 bzw. 2.5 Punkte) erfüllt ist. Dies kann nach oben kaskadieren. Aufgrund dessen mit 0 bewertete Anforderungen werden hier durch Kursivierung und Fettdruck hervorgehoben.

Rahmenanforderungen

Anforderung	Gewicht	Produktbewertung			
		LinOTP	*Defender*	*OpenOTP*	*SecureAuth*
1 Rahmenanforderungen	2	2.73	2.91	2.73	3.0
1-1 Universelle Lösung	3	3	3	3	3
1-2 Einbezug bestehender Komponenten als zweiten Faktor	1	3	2	3	3
1-3 Verbreitungsgrad und Akzeptanz	3	2	3	2	3
1-4 inhouse hosting	**4**	3	3	3	3

Tabelle 5.5.: Produktbewertung: Kategorie 1 Rahmenanforderungen

Alle 4 Kandidaten sind inhouse hostbar und für alle Anwendungsszenarien geeignet. Sie alle können Smartphones der Anwender als Träger des zweiten Faktors z. B. in Form von OTP-Generatoren miteinbeziehen. Durch den Verzicht auf eine Mobile-Push-Funktion gelingt dies *Defender* jedoch nicht so gut, wie den übrigen Kandidaten.

Da *SecureAuth Corporation* und *One Identity (Quest Software Inc.)* bei Gartner gelistet werden, erhalten sie bzgl. *1-3 Verbreitungsgrad und Akzeptanz* einen Vorsprung. Auf ihren Homepages führen aber auch *KeyIdentity GmbH* und *RCDevs SA* einige Kundenreferenzen.

Einrichtungsaufwand

Anforderung	Gewicht	Produktbewertung *LinOTP*	*Defender*	*OpenOTP*	*SecureAuth*
2 Einrichtungsaufwand	3	2.5	2.32	2.43	2.38
2-1 Integration in bestehende AAA-Systeme	**4**	3	3	3	3
2-2 Zusätzliche IT-Infrastruktur	2	**???**	**???**	**???**	**???**
2-3 Aufwand der Einführung	3	2.83	2.58	2.83	2.58
2-3-1 Import bestehender Anwenderkennungen	**4**	3	3	3	3
2-3-2 Hardwareanpassung der Anwenderterminals	3	3	3	3	3
2-3-3 Anpassung der Anwendungen	2	2	2	2	2
2-3-4 Verteilen der zweiten Faktoren an die Anwender	3	3	2	3	2
2-4 Einrichtungsaufwand für den Anwender	2	3.0	3.0	3.0	3.0
2-4-1 Clientseitige Einrichtung auch durch technisch nicht-versierte	3	3	3	3	3
2-4-2 Fremdrechner	3	3	3	3	3
2-5 Technische Dokumentation	3	3	2	3	3
2-6 Anleitungsmaterial für Endanwender	1	2	3	1	1

Tabelle 5.6.: Produktbewertung: Kategorie 2 Einrichtungsaufwand

Den Hersteller-Dokumentationen nach sind alle Produkte prinzipiell in die auch am LRZ eingesetzten AAA-Systeme wie LDAP, MS AD und RADIUS integrierbar (*2-1*). Diese Aussage basiert jedoch ausschließlich auf Angaben der Hersteller und wurde nicht durch Tests am LRZ untermauert. Auf den ersten Blick tauchten während der Lektüre der Dokumentationen keine offensichtlichen Integrationshindernisse auf. Zwar scheint *Defender* sehr auf MS AD fokussiert, es stehen ihm aber auch andere Importquellen zur Verfügung. Wegen ihrer offenen APIs dürften *LinOTP* und *OpenOTP MFA Suite* am wenigsten Integrationsprobleme bereiten.

Ohne fundiertes Wissen um den Bestand am LRZ, fällt die Bewertung von *2-2* schwer.

Im *2-3 Aufwand der Einführung* unterscheiden sich die Produkte wohl hauptsächlich in *2-3-3 Anpassung der Anwendungen* und dem *2-3-4 Verteilen der zweiten Faktoren an die Anwender*, da dem *4.2 Konzeptvorschlag* nach keine Hardwarelesegeräte benötigt werden. *SecureAuth IdP + core security* und *LinOTP* bieten zur Integration in Webapplikationen Beispielcode. Darüber hinaus stellt *SecureAuth IdP + core security* eine nennenswerte Zahl an Anleitungen zur 2FA-Integration via SAML in verschiedene proprietäre Produkte wie bspw. JIRA bereit. 2FA via Mobile-Push erfordert keine Veränderung der GUI. OTPs können mit dem PW konkateniert oder in einem eigenen Formularfeld eingegeben werden. Für Desktoplogin-, VPN- und SSH/PAM-Integration werden jeweils fertige Module bereitgestellt. Glaubt man den Anleitungen, dürfte der Aufwand durchaus vergleichbar (gering)

sein.

Für das Ausrollen der zweiten Faktoren an die Anwender setzen alle Produkte auf User-Self-Service oder (QR-Codes per) Email. *LinOTP* bietet sogar für Hardwaretoken eine Auto-Enrollment-Funktion: Hardwaretoken werden per Bulk-Import, User aus z. B. LDAP registriert. Die Zuordnung von Hardwaretoken zu Anwender geschieht dann automatisch bei Erstbenutzung eines beliebigen Tokens.

Da zweite Faktoren mit dem Smartphone als Träger einfach durch scannen eines QR-Codes in Betrieb genommen werden und Hardwaretoken i. d. R. selbsterklärend sind, ist der *2-3 Aufwand der Einführung* gering. Lediglich für eine aktuelle Browser- und VPN-Version muss gesorgt werden. Selbiges gilt für Fremdrechner.[11]

Die *2-5 Technische Dokumentation* aller Produkte ist ausführlich, nur die von *Defender* ließ sich etwas schwer durchsuchen und enthält anscheinend keinerlei Informationen über Backup und Wiederherstellung. Dafür gibt *One Identity (Quest Software Inc.)* mit dem *Token User Guide* als einziger *2-6 Anleitungsmaterial für Endanwender* an die Hand.[12]

[11] Die Einrichtung von 2FA für SSH mittels U2F (anstatt von OTP) ist zwar aufwendiger, SSH-Verbindungen aber werden vermutlich selten von technisch nicht-versierten Personen aufgebaut.
Der Desktoplogin an Fremdrechnern ist für diese Anforderung nicht relevant, da auf Fremdrechner für gewöhnlich nicht mit den eigenen Kennungsdaten zugegriffen wird oder eine Einrichtung dessen nicht durch den Anwender zu erfolgen hat.

[12] Technische Dokumentationen:

- *LinOTP* : `https://www.linotp.org/doc/latest/`
- *Defender* : `https://support.oneidentity.com/de-de/defender/5.9.1/technical-documents`
- *OpenOTP MFA Suite* : `https://www.rcdevs.com/docs/`
- *SecureAuth IdP + core security* : `https://docs.secureauth.com/display/SI91/SecureAuth+IdP+version+9.1+-+9.2` bzw. `https://docs.secureauth.com/display/KBA/`

Verwaltung, Wartung, Erweiterbarkeit

Anforderung	Gewicht	Produktbewertung			
		LinOTP	*Defender*	*OpenOTP*	*SecureAuth*
3 Verwaltung, Wartung, Erweiterbarkeit	**4**	2.72	2.54	2.71	2.65
3-1 Erweiterbarkeit	3	3.0	2.5	3.0	3.0
3-1-1 Neue Dienste	2	3	2	3	3
3-1-2 Neue Anwender	**4**	3	3	3	3
3-1-3 Weitere 2FA-Technologien	1	3	3	3	3
3-1-4 Adaption an den „Stand der Technik“	3	3	2	3	3
3-2 Wartung	3	2.5	2.5	2.5	2.5
3-2-1 Supportzusage	3	3	3	3	3
3-2-2 Kontinuierlicher Aufwand des zweiten Faktors	3	3	3	3	3
3-2-3 Kontinuierlicher Aufwand der 2FA-Dienste	2	**???**	**???**	**???**	**???**
3-2-4 Backup	**4**	3	3	3	3
3-3 Verwaltung	**4**	2.67	2.6	2.64	2.51
3-3-1 Funktionsumfang der Verwaltungssoftware	3	3.0	2.9	2.7	2.5
3-3-1-1 Automatisierbarkeit von Routinearbeiten	**4**	3	3	3	3
3-3-1-2 Unterstützung verschiedener 2FA-Technologien	1	3	2	3	3
3-3-1-3 Self-Service	3	3	3	3	2
3-3-1-4 Helpdesk-Rolle	1	3	3	0	3
3-3-1-5 Verwaltung von Hardwaretoken	1	3	3	3	1
3-3-2 Auswirkung auf bisherigen Prozess zur Kennungsverwaltung	3	**???**	**???**	**???**	**???**
3-3-3 Änderung des zweiten Faktors	3	3	3	3	3
3-3-4 Wiederherstellung nach Diebstahl/Verlust	2	3	3	3	3
3-3-5 Sperren von zweiten Faktoren	3	3	3	3	3
3-3-6 Anwender-Self-Service nach Verlust des zweiten Faktors (Fallback)	2	2	1	3	3
3-3-7 Mehrere zweite Faktoren pro Anwender	1	3	3	3	3
3-3-8 Geltungsbereich der 2FA	**4**	3	3	3	3

	3-3-9 Ausweitung auf synchronisierte Benutzerverwaltungen	**4**	3	3	3	3
	3-3-10 Portierungsmöglichkeit	1	3	3	3	**???**
	3-3-11 Temporäre Zugänge	3	3.0	3.0	2.38	2.12
	3-3-11-1 Gastzugänge	3	3	3	3	3
	3-3-11-2 Tagesersatz für zweiten Faktor	2	3	3	2	1
	3-3-11-3 Einrichtungsaufwand für temporäre Zugänge	3	3	3	2	2
	3-3-12 Einfluss auf Verwaltung des ersten Faktors	2	3	3	3	3
	3-3-13 Rücksetzung des ersten Faktors	2	3	3	3	3

Tabelle 5.7.: Produktbewertung: Kategorie 3 Verwaltung, Wartung, Erweiterbarkeit

3-1 Insgesamt scheinen alle Produkte gut erweiterbar. Neue Anwender sind laut Herstellerangaben nur durch das jeweilige Lizenzmodell beschränkt und weitere Dienste lassen sich auch nach der Einführung des Produktes leicht mit 2FA ausstatten. Die Auswahl an integrierbaren Anwendungen für *Defender* scheint jedoch begrenzt (siehe Kategorie *7 Anwendungsbereiche*).

Eine *3-1-4 Adaption an den „Stand der Technik“* ist bei *LinOTP* und *OpenOTP MFA Suite* wegen ihres offenen Designs z. B. durch Auswahl der den OTPs zugrundeliegenden Hash-Algorithmen durch den Administrator möglich. *SecureAuth IdP + core security* und *Defender* sind durch Updates zu versorgen. Obwohl hinter *Defender* ein durchaus großes Unternehmen steht, werden hier Entwicklungen wie kontextbasierte MFA oder Mobile-Push vermisst.

Der Aufwand zur *3-2 Wartung* (z. B. Patches einspielen etc.) der Produkte ist schwer einzuschätzen. Aufwand zur Pflege der zweiten Faktoren selbst entsteht nur bei Wahl von OTP-Hardwaretoken. Als Batterielaufzeiten werden für solche aber bis zu fünf Jahre angeben.[13] Zu *Defender* wurden in der Dokumentation keine Hinweise bzgl. Backup und Wiederherstellung gefunden. Der Support wies darauf hin, dass *Defender* alle relevanten Informationen (Tokens, Lizenzen, Access Node-Konfigurationen) in Active Directory ablegt und sich die Backup-Thematik somit nahezu vollständig dorthin verschiebt. Die übrigen Produkte bieten Anleitungen zur Desaster-Recovery.

Der Umfang der jeweiligen Management-Software (*3-3-1*) ist ähnlich. *OpenOTP MFA Suite* fehlt eine Helpdesk-Rolle. *SecureAuth IdP + core security* bietet weniger Self-Service- und Hardwaretokenverwaltungsfunktionalität als die übrigen. *Defender* unterstützt weniger viele verschiedene 2FA-Technologien wie bspw. Mobile-Push.

Die Einführung einer 2FA am LRZ wird sicherlich *3-3-2 Auswirkung auf bisherigen Prozess zur Kennungsverwaltung* haben. Die Effekte von Self-Service, zusätzlich aufkommenden Helpdesk-Anfragen, Hardware-Token-Vergabe etc. lässt sich ohne Livetest aber kaum beziffern. Da alle Produkte *3-3-1-1 Automatisierbarkeit von Routinearbeiten* unterstützen, wird jedoch eine geringe Beeinträchtigung angenommen.

Wegen verschiedener Gruppen-/Realm-Funktionen ist eine ungewollte *3-3-9 Ausweitung auf synchronisierte Benutzerverwaltungen* wohl nicht zu erwarten. Prinzipiell ließe sich die

[13] `https://www.oneidentity.com/de-de/products/defender/token-flexibility.aspx`

2FA pro LDAP-Nutzer einzeln aktivieren. Zweiter und erster Faktor bleiben unabhängig voneinander (*3-3-12*, *3-3-13*).

Lost-Token-Szenarien lassen sich bei allen Tools über den Helpdesk bzw. Administrator oder durch mehrere 2F pro Anwender abwickeln. *OpenOTP MFA Suite* bietet darüber hinaus auch Emergency-OTPs bzw. Recoverycodes für den Self-Service.

Das Ausstellen temporärer 2FA-(Ersatz-)Zugänge wird durch die Verwaltungssoftware von *LinOTP* sowie *Defender* explizit unterstützt. Auch ist das Definieren eines Ablaufdatums solcher möglich. Mit den beiden anderen Tools bleibt der Umweg einen neuen Token zu erstellen, dem Nutzer zuzuordnen und später wieder manuell zu löschen.

Trotz weniger moderner GUI (siehe *6 Bedienkomfort*), liegen die beiden open-source Produkte bzgl. *3 Verwaltung, Wartung, Erweiterbarkeit* vorn.

Sicherheitsniveau

Anforderung	Gewicht	Produktbewertung			
		LinOTP	*Defender*	*OpenOTP*	*SecureAuth*
4 Sicherheitsniveau	3	2.27	2.18	2.29	2.38
4-1 Nichtabstreitbarkeit	2	2	2	2	3
4-2 Angemessenes Sicherheitsniveau	3	2.4	2.4	2.5	3.0
4-2-1 Stand der Technik	3	2	2	2	3
4-2-2 Zweiter Faktor nicht ableitbar	3	3	3	3	3
4-2-3 Angemessene Verschlüsselungsverfahren	3	3	3	3	3
4-2-4 Kontextbasierte MFA	1	0	0	1	3
4-3 Schutz gegen Spoofing	3	1.5	1.0	1.5	1.0
4-3-1 Fund des zweiten Faktors nicht ausreichend	3	3	2	3	2
4-3-2 Replay	3	**???**	**???**	**???**	**???**
4-4 Übertragung des zweiten Faktors	3	2.67	2.5	2.67	2.67
4-4-1 Out-of-Band	1	2	1	2	2
4-4-2 Anzahl der Übertragungen	1	2	2	2	2
4-4-3 Kein Klartext	**4**	3	3	3	3
4-5 Injektivität	2	3	3	3	3
4-6 Verfügbarkeit	**4**	2.57	2.57	2.57	2.57
4-6-1 Ausfallsicherheit des 2FA-Systems	**4**	3	3	3	3
4-6-2 Intervall der Authentifizierungsvorgänge	1	2	2	2	2
4-6-3 Physische Robustheit	2	2	2	2	2
4-7 Manipulation des zweiten Faktors	3	2	2	2	2
4-8 Keine Speicherung des zweiten Faktors	2	2	2	2	2

Tabelle 5.8.: Produktbewertung: Kategorie 4 Sicherheitsniveau

Es ist wohl anzunehmen, dass Mobile-Push zusammen mit kontext- bzw. risikobasierte MFA bald als Stand der Technik gelten wird. Ein höheres Maß an Authentizität, Out-of-Band-Übertragung und mehr Komfort sind nennenswerte Vorteile. Einige Hersteller führen manuelles Abtippen von OTPs bereits unter „Legacy“. Dies gelingt *SecureAuth IdP + core security* bereits recht gut. OATH-OTP und FIDO/YubiKeys dürften als *4-2-3 Angemessene Verschlüsselungsverfahren* gelten. Welche Verfahren im Rahmen der verschiedenen Mobile-Push-Authentifizierung eingesetzt werden, wurde nicht vollständig untersucht.

I. d. R. ist der *4-3-1 Fund des zweiten Faktors nicht ausreichend* (Recoverycodes ausgenommen). *Defender* und *SecureAuth IdP + core security* lassen sich jedoch auch so konfigurieren, dass beim Login nur der zweite Faktor gefordert wird. *OpenOTP MFA Suite* akzeptiert jedes gültige OTP nur höchstens einmal, was *4-3-2 Replay* verhindert. Die übrigen Tools stellten

hierzu keine Informationen bereit.

Die *4-4 Übertragung des zweiten Faktors* kann bei Mobile-Push oder Email Out-of-Band erfolgen. Eine mehrfache Übertragung dessen (Server→Client & Client→Server) ist nur bei Email oder SMS erforderlich.

LinOTP, *OpenOTP MFA Suite* und *SecureAuth IdP + core security* können eine hohe *4-6-1 Ausfallsicherheit des 2FA-Systems* durch Clustering bzw. parallele Instanzen und replizierte Datenbanken gewährleisten. Die bereitgestellten VM-Images erleichtern eine solche Implementierung. Auch wenn in der *Defender*-Dokumentation keine entsprechenden Informationen hierzu gefunden wurden, ist nach Supportaussage auch hier eine entsprechende Installationsvariante basierend auf einem weitgehend ausfallsicheren Active Directory möglich. Da *OpenOTP MFA Suite* ein TOTP nur höchstens einmal pro Gültigkeitsintervall akzeptiert, ist bei Wahl von TOTP als zweiten Faktor nur höchstens ein Authentifizierungsvorgang pro Gültigkeitsintervall möglich.(Vgl. Kapitel *2.5.1 Kontinuierliche Neuberechnung*) Mobile-Push und FIDO sollten derartige Beschränkungen nicht kennen. Die Verwaltungssoftware aller Produkte bietet Einstellungen für erlaubte Anzahlen von (validen und fehlgeschlagenen) Authentifizierungsvorgängen pro Zeitintervall.

Dem *4.2 Konzeptvorschlag* entsprechend werden keine Hardwarelesegeräte benötigt. Im Gegensatz zu FIDO ist eine *4-7 Manipulation des zweiten Faktors* bei OTP-Softwaretoken oder bei Mobile-Push prinzipiell denkbar.

Das *4 Sicherheitsniveau* aller Produkte scheint also angemessen.

Kosten(-effizienz)

Anforderung	Gewicht	Produktbewertung			
		LinOTP	*Defender*	*OpenOTP*	*SecureAuth*
5 Kosten(-effizienz)	2	2.38	2.54	2.44	1.75
5-1 Für das LRZ	3	2.64	2.62	2.73	2.05
5-1-1 Pro Mitarbeiter	3	2.0	2.6	2.0	1.2
5-1-1-1 Anschaffung pro Mitarbeiter	1	2	2	2	1
5-1-1-2 Kontinuierlich pro Mitarbeiter	3	2	3	2	1
5-1-1-3 Wiederherstellung nach Verlust pro Mitarbeiter	1	2	2	2	2
5-1-2 2FA-Infrastuktur	2	3	2	3	1
5-1-3 Verwaltung von 2FA-Authentifizierungsinformationen	3	3	3	3	3
5-1-4 Späteres Hinzufügen neuer 2FA-Nutzer	1	2	2	3	2
5-1-5 Pro auszustattender Anwendung	2	3	3	3	3
5-2 Für Nutzer von LRZ-Diensten	2	2.0	2.43	2.0	1.29
5-2-1 Anschaffung für den Nutzer	2	2	2	2	1
5-2-2 Kontinuierlich für den Nutzer	3	2	3	2	1
5-2-3 Wiederherstellung nach Verlust für den Nutzer	2	2	2	2	2

Tabelle 5.9.: Produktbewertung: Kategorie 5 Kosten(-effizienz)

Vor der Anforderungsbewertung dieser Kategorie werden die Lizenzmodelle aller Produkte verglichen. Die Darstellungen gründen auf Telefonaten mit den Sales- bzw. Presales-Abteilungen der jeweiligen Hersteller im Oktober 2018. Angefragt wurden jeweils die Preise für das den Einsatzszenarien entsprechende Lizenzpaket bei 250 Mitarbeitern und einer Laufzeit von einem sowie drei Jahren. Nach detaillierterer Darstellung des AAA-Systems des LRZs ausgehandelte Angebote könnten aber besser ausfallen. Insofern dürfen die hier genannten Beträge nur als Orientierung und nicht als verbindliche Angebote der Hersteller angesehen werden. Die meisten Hersteller gewährten einen Rabatt für den universitären Einsatz.

- *KeyIdentity GmbH* lizensiert pro Token und Monat. Bei 250 Mitarbeitern, fallen für ein Jahr 5.500€, für drei Jahre 14.500€ abzüglich eines Rabattes von 3-5% an. Die Software selbst ist nicht zu vergüten. Hardwaretoken können getrennt bezogen werden. Gegen Aufpreis ließe sich der Support von Mo-Fr, 8-18 Uhr auch auf 24/7 ausweiten. Sollen nur Webanwendungen mit 2FA ausgestattet werden, ist *LinOTP* unabhängig von der Nutzerzahl kostenlos.

- *One Identity (Quest Software Inc.)* berechnet pro User und Jahr. Für Softwaretoken werden 37,18€, für Hardwaretoken 52,35€ im ersten Jahr fällig. Ab dem zweiten Jahr werden nur noch 20% des Tokenspreises an Wartung berechnet. Das 2FA-Modul für *Defender* kostet 908,09€ pro Jahr unabhängig von der Nutzerzahl. Für Universitäten gewährt *One Identity (Quest Software Inc.)* einen beachtenswerten Rabatt von 40% (auf den Tokenpreis).

- *RCDevs SA* vertreibt Volumenpakete. Für das Paket mit 250 Mitarbeitern werden 4.500€ mit einem Jahr bzw. 10.800€ mit drei Jahren Laufzeit verlangt. Zusatzkosten entstehen für optionale Hardwaretoken, die über den Hersteller bezogen werden können, nicht aber für die Software. Einige Prozent Rabatt sind zu erwarten. Für bis zu 40 Nutzern ist *OpenOTP MFA Suite* mit allen Funktionen kostenlos.

- *SecureAuth Corporation* berechnet ebenfalls pro User und Monat. Für das Paket *protect* fallen 3$ pro User und Monat an, für das Paket *prevent*, das die risikobasierte Multifaktorauthentifizierung enthält, 5$. Daneben ist aber für die Plattform eine Lizenzgebühr von 3.600$ pro VM und Jahr zu entrichten.

In den Tabellen 5.10 und 5.11 werden die Lizenzkosten der 4 Produkte bei einer Laufzeit von einem bzw. drei Jahren gegenüber gestellt. Hierbei liegt ein Umrechnungskurs von USD in EUR von 0,8718 zu Grunde.

Kosten für ein Jahr Laufzeit	LinOTP	Defender	OpenOTP	SecureAuth IdP	
				protect	*prevent*
Plattform (1 Jahr)	0 €	908,09 €	0 €	3.138,48 €	3.138,48 €
Kosten /User/Jahr	22,00 €	22,31 €	18,00 €	31,38 €	52,31 €
Summe	5.500,00 €	6.485,09 €	4.500,00 €	10.984,68 €	16.215,48 €
	abzüglich 3-5% Rabatt	Softtoken inkl. 40% Rabatt	abzüglich 3-5% Rabatt	nur eine VM	

Tabelle 5.10.: Voraussichtliche Kosten pro Produkt bei einem Jahr Laufzeit und 250 Anwendern

Kosten für drei Jahre Laufzeit	LinOTP	Defender	OpenOTP	SecureAuth IdP	
				protect	*prevent*
Plattform (3 Jahre)	0 €	2.724,27 €	0 €	9.415,44 €	9.415,44 €
Kosten /User/Jahr	19,33 €	22,31 €	14,40 €	31,38 €	52,31 €
ab dem 2. Jahr		4,46 €			
Summe	14.500,00 €	10.532,07 €	10.800,00 €	32.954,04 €	48.646,44 €
	abzüglich 3-5% Rabatt	inkl. 40% Rabatt Softtoken	abzüglich 3-5% Rabatt	nur eine VM	

Tabelle 5.11.: Voraussichtliche Kosten pro Produkt bei drei Jahren Laufzeit und 250 Anwendern

Die Bewertung der Anforderungen erfolgte relativ zueinander. *SecureAuth IdP + core security* verlangt pro User und für die Plattform die höchsten Lizenzgebühren. Ein *5-1-4 Späteres Hinzufügen neuer 2FA-Nutzer* ist im Rahmen der Volumenlizenzen von *OpenOTP MFA Suite* am günstigsten. Für eine Wiederherstellung nach Verlust (*5-1-1-3*, *5-2-3*) werden nur bei Hardwaretoken Kosten fällig. Keines der Produkte verursacht Kosten *5-1-5 Pro auszustattender Anwendung*.

Insgesamt kann das Lizenzmodell von *Defender* durch beachtenswerten Universitätsrabatt und Reduzierung der Kosten pro User ab dem zweiten (User-)Jahr auf lange Sicht punkten.

Bedienkomfort

Anforderung	Gewicht	Produktbewertung			
		LinOTP	*Defender*	*OpenOTP*	*SecureAuth*
6 Bedienkomfort	3	2.35	2.5	2.45	2.75
6-1 Einarbeitungszeit für Anwender	3	2	3	2	3
6-2 Verlängerung des Authentifizierungsvorgangs	**4**	3	3	3	3
6-3 Unkomplizierte Handhabung	3	2	1	2	2
6-4 Verschiedene Authentisierungsmethoden	1	2	3	2	3
6-5 Branding	1	3	1	3	3
6-6 Mehrbelastung der Anwender	**4**	3.0	3.0	3.0	3.0
6-6-1 Transportierbarkeit	**4**	3	3	3	3
6-6-2 Einheitlichkeit	3	3	3	3	3
6-7 Private Nutzung	1	1	1	1	1
6-8 Redundante zweite Faktoren	1	3	3	3	3
6-9 Design der User-Interfaces	2	1	3	2	3

Tabelle 5.12.: Produktbewertung: Kategorie 6 Bedienkomfort

Defender und *SecureAuth IdP + core security* bieten klarere bzw. übersichtlichere Interfaces in modernem Design, was für Nutzer sicherlich von Vorteil ist (*6-1*, *6-9*). Durch eigene CSS-Stylesheets und Logos lässt sich das Design von *SecureAuth IdP + core security* etwas an das *6-5 Branding* des LRZ anpassen. Die open-source Produkte *OpenOTP MFA Suite* und *LinOTP* erlauben größere Änderungen der GUI. Verwendet man die API-Calls von *LinOTP*, ließe sich dessen Funktion auch vollständig in einem bestehenden eigenen Portal implementieren. *Defender* liefert hierzu keine Informationen. Die jeweiligen Smartphone-Apps (zumindest für Android) entsprechen bei allen Produkten modernen Designprinzipien.

Fast alle Produkte bieten durch Mobile-Push oder Hardwaretoken eine weitgehend *6-3 Unkomplizierte Handhabung* des zweiten Faktors, dessen Komfort lediglich bei Auswahl des manuellen Abtippens von OTPs reduziert wird. *Defender* verzichtet mit Mobile-Push auch

auf eine recht komfortable 2FA. Insgesamt werben *SecureAuth IdP + core security* und *Defender* jedoch mit einer größeren Auswahl an konkret möglichen (proprietären) zweiten Faktoren (*6-4*).

Mit jedem Produkt lassen sich einzelnen Anwendern auch mehrere *6-8 Redundante zweite Faktoren* zuordnen, was abhängig vom Lizenzmodell (pro Token oder pro User) zusätzliche Kosten verursachen kann. Eine „wirkliche“ *6-7 Private Nutzung* wäre nur mit FIDO bzw. YubiKeys möglich, die sowieso pro Webseite ein eigenes Schlüsselpaar erzeugen. In einer OTP-Smartphone-App ließen sich für den beruflichen und privaten Gebrauch verschiedene Seeds hinterlegen, was eine komfortable quasi private Nutzung ermöglicht.

Aus Kategorie *6 Bedienkomfort* geht somit *SecureAuth IdP + core security* hauptsächlich wegen modernen Designs und Mobile-Push hervor. Authentifizierung mit Mobile-Push als zweitem Faktor scheint jedoch gegen *1-4 inhouse hosting* zu verstoßen. Dies ist nicht zwingend der Fall. Für das Erzeugen von Push-Benachrichtigungen auf Smartphones sind zwar immer Server von den Betriebssystemanbietern Apple oder Google nötig. Die betrachteten Produkte unterscheiden sich jedoch in den hierfür zu übertragenden Informationen.

Während *OpenOTP MFA Suite* (neben einer ID des Smartphones) einen Identifier des Authentifizierungsvorgangs verschickt, sendet *LinOTP* eine quasi inhaltsleere Nachricht, die lediglich das Öffnen der *KeyIdentity Authenticator*-App auf dem Zielgerät bewirkt. Die *KeyIdentity Authenticator*-App verbindet sich dann mit der on-premise gehosten *LinOTP*-Instanz (oder einem von außen erreichbaren Proxy derer) und wickelt den Authentifizierungsvorgang ohne Beteiligung externer Server ab. Die Push-Benachrichtigung dient also nur als Erinnerung. Es reicht aber auch aus, wenn der Anwender bei Forderung nach seinem zweiten Faktor manuell die App auf seinem Smartphone öffnet, womit *1-4 inhouse hosting* bei nahezu gleichem Bedienkomfort gewährleistet würde.

Der Bedienkomfort für den Administrator ist aufgrund unterschiedlicher Ansätze der Produkte etwas schwerer zu fassen. *LinOTP* und *OpenOTP MFA Suite* lassen sich UNIX-typisch durch verschiedene .config-Files konfigurieren und skripten. *Defender* und *SecureAuth IdP + core security* sind dagegen eher Windows-Tools, die über Mausklicks in vielen Dialogen administriert werden. Eine Bewertung bleibt hier den Vorlieben des Administrators überlassen. Tägliche Verwaltungsaufgaben wie das Hinzufügen eines neuen Users können bei allen Produkten aber über eine webbasierte GUI mit wenigen Klicks erledigt werden.

Anwendungsbereiche

Anforderung	Gewicht	Produktbewertung			
		LinOTP	*Defender*	*OpenOTP*	*SecureAuth*
7 Anwendungsbereiche	**4**	2.84	***0***	2.89	2.89
7-1 An LRZ-Kennung koppelbar	**4**	3	3	3	3
7-2 RADIUS-Integration	**4**	3	3	3	3
7-3 LDAP-Integration	**4**	3	3	3	3
7-4 Mobilgeräte	3	2.71	2.86	2.71	2.71
7-4-1 Android	**4**	3	3	3	3
7-4-2 Blackberry	2	1	2	1	1
7-4-3 iOS	**4**	3	3	3	3
7-4-4 USB-Schnittstelle nicht erforderlich	**4**	3	3	3	3
7-5 Fernwartung per SSH (Szenario 1)	**4**	2.8	2.8	2.8	2.8
7-5-1 Eingabe über Kommandozeile	3	3	3	3	3
7-5-2 SSH-Integration	**4**	3	3	3	3
7-5-3 UNIX-Kompatibilität	**4**	3	3	3	3
7-5-4 Dezentrale Authentifizierung	1	**???**	**???**	**???**	**???**
7-5-5 Windows-SSH-Clients	3	3	3	3	3
7-6 Telearbeit und VPN (Szenario 2)	**4**	3.0	2.69	3.0	3.0
7-6-1 VPN-Server Integration	**4**	3	3	3	3
7-6-2 Kompatibel mit Cisco Anyconnect Desktop-VPN-Client	**4**	3	3	3	3
7-6-3 Kompatibel mit weiteren gängigen Desktop-VPN-Clients	3	3	2	3	3
7-6-4 Kompatibel mit Cisco Anyconnect mobile VPN-Client	3	3	3	3	3
7-6-5 Kompatibel mit weiteren gängigen mobilen VPN-Clients	2	3	2	3	3
7-7 Login an Arbeitsplatzrechnern (Szenario 3)	**4**	2.84	2.84	2.84	2.84
7-7-1 UNIX-Desktoploginsysteme	**4**	3	3	3	3
7-7-2 MacOS-Desktoploginsysteme	**4**	3	3	3	3
7-7-3 Windows-Desktoploginsysteme	**4**	3	3	3	3
7-7-4 MS Active Directory	**4**	3	3	3	3
7-7-5 Sperrbildschirm	1	0	0	0	0
7-7-6 Offline Modus	2	3	3	3	3
7-8 Webapplikationen (Szenario 4)	**4**	2.57	2.07	2.67	2.67
7-8-1 Webbrowserkompatibilität	**4**	2.67	2.67	2.67	2.67
7-8-1-1 Apple Safari	3	2	2	2	2
7-8-1-2 Google Chrome	3	3	3	3	3

	7-8-1-3 Mozilla Firefox	3	3	3	3	3
	7-8-2 Integration in Webanwendungen	3	3.0	1.0	3.0	3.0
	7-8-2-1 PHP	3	3	1	3	3
	7-8-2-2 JavaScript	3	3	1	3	3
	7-8-2-3 Python	2	3	1	3	3
	7-8-3 Föderiertes Identitätsmanagement	1	2	3	3	3
	7-8-4 Lokale Authentifizierungsdienste	2	2	2	2	2
	7-9 Plugins für gängige Anwendungen	1	2	1	3	3

Tabelle 5.13.: Produktbewertung: Kategorie 7 Anwendungsbereiche

Wie schon bei *2-1* erwähnt, sind nach Herstellerangaben alle Produkte mit RADIUS und LDAP kompatibel. Somit sollte eine Kopplung an die LRZ-Kennungen der Anwender möglich sein.

Hardwarebasierte zweite Faktoren wie FIDO/YubiKey (mit NFC-Schnittstelle) lassen sich auch über *7-4 Mobilgeräte* eingeben. Als OTP-Generatoren und für Mobile-Push werden auf Android- und iOS-Geräte entsprechende Apps bereitgestellt. Einzig *Defender* unterstützt Blackberry umfänglicher.

Fernwartung per SSH (Szenario 1) Eine 2FA für *7-5 Fernwartung per SSH (Szenario 1)* realisieren alle Produkte durch UNIX-PAM. Somit ließe sich die 2FA auch auf weitere UNIX-Dienste wie bspw. FTP ausweiten. OTPs lassen sich problemlos über die Konsole eingeben, Mobile-Push wird Out-of-Band authentifiziert und FIDO U2F sowie YubiKey-OTP scheinen mit etwas Konfigurationsaufwand ebenfalls möglich.

Für eine *7-5-4 Dezentrale Authentifizierung* an SSH-Servern stellen sich die unter *7-8-4* diskutierten Probleme. Darüber hinaus ist aber auch der Einsatz von YubiKeys möglich.[yub18]

Telearbeit und VPN (Szenario 2) 2FA für *7-6 Telearbeit und VPN (Szenario 2)* bieten alle Produkte in recht ähnlichem Umfang. Neben dem Cisco VPN-Client werden auch einige weitere gängige Clients in mobiler und Desktopversion unterstützt. *Defender* führt zwar keine weiteren kompatiblen VPN-Clients an, bezeichnet Cisco aber nur als Beispiel.

Login an Arbeitsplatzrechnern (Szenario 3) Einen *7-7 Login an Arbeitsplatzrechnern (Szenario 3)* mittels 2FA ermöglichen alle Produkte ab Windows 10 (*OpenOTP MFA Suite* ab Vista, *SecureAuth IdP + core security* ab Windows 7) und MacOSX. In UNIX-Systeme wird über PAM integriert. Entsprechende Credential Provider bzw. Module werden bereitgestellt. *Defender* erlaubt einen Login an MacOS nur über Active Directory Bridging.

Lokale Benutzerkonten können durch *SecureAuth IdP + core security* und *Defender* nicht mit 2FA ausgestattet werden. *OpenOTP MFA Suite* ermöglicht dies hingegen, indem Nutzername und Passwort lokal und der zweite Faktor remote validiert werden können.

Ein *7-7-6 Offline Modus* für den 2FA-Desktoplogin ist durch Caching nach dem ersten erfolgreichen Online-2FA-Login möglich. *Defender* bietet für dieses Szenario weitere Optionen wie bspw. eine Beschränkung der Zahl an offline-Logins oder die Forderung einer Passwortänderung im Anschluss.

Webapplikationen (Szenario 4) *7-8 Webapplikationen (Szenario 4)* mit 2FA auszustatten setzt einerseits eine *7-8-1 Webbrowserkompatibilität* der Eingabe von (nicht Out-of-Band authentifizierten) zweiten Faktoren voraus. Diese ist für Google Chrome und Mozilla Firefox gegeben. Apple Safari unterstützt das FIDO U2F-Protokoll jedoch bislang nicht. Die andererseits nötige *7-8-2 Integration in Webanwendungen* ermöglichen *LinOTP*, *OpenOTP MFA Suite* und *SecureAuth IdP + core security* über SAML oder (RESTful) APIs bzw. SDKs für gängige Programmiersprachen im Webkontext inkl. Beispielcode.

Defender erlaubt mit seinem *ISAPI Agent* lediglich den Zugriff auf solche Webseiten mit 2FA auszustatten, die auf einem Microsoft Web Server (IIS) gehostet sind. Diese Hosting-Variante kommt am LRZ jedoch selten vor. Eine mögliche Abhilfe stellt der *One Identity (Quest Software Inc.) Cloud Access Manager*[14] dar. Hierbei handelt es sich um einen zentralen Proxy, der mit Webanwendungen und unterschiedlichen Identity Providern über SAML interagiert und so v. a. ein SSO-Portal realisiert. *Defender* kann *Cloud Access Manager* als IdP dienen. Da ein SSO-Portal/-Hub nicht der Aufgabenstellung dieser Arbeit entspricht, wird *One Identity (Quest Software Inc.) Cloud Access Manager* hier nicht weiter untersucht und nicht in die Bewertung von *Defender* miteinbezogen. Sollte eine zukünftige Arbeit jedoch SSO für Dienste des LRZs zum Ziel haben, könnte an dieser Stelle fortgefahren werden.[15]

Unterstützung für *7-8-3 Föderiertes Identitätsmanagement* stellt *LinOTP* durch SAML 1.1 und Shibboleth bereit. Die übrigen Produkte verfügen neben SAML 2.0 auch über MS ADFS-Kompatibilität.

Die Anforderung, auch *7-8-4 Lokale Authentifizierungsdienste* durch einen zweiten Faktor stärken zu können, zielt weniger auf die Produkte, sondern mehr auf die eingesetzte 2FA-Technologie ab. Um solche nicht an die zentrale 2FA-System angeschlossene Webanwendungen mit genau den im zentralen 2FA-System hinterlegten zweiten Faktor lokal authentisieren können, müssten die Seeds der Token lokal hinterlegt (und synchron gehalten) und offene 2FA-Standards zur Validierung verwendet werden. Attraktiver erscheint eine wie unter *6-7 Private Nutzung* geschilderte Variante eines 2F, der für mehrere unabhängige Dienste bzw. Nutzerkonten registriert werden kann. Hierfür scheinen FIDO U2F oder OATH-OTP-Smartphoneapps geeignet, welche von allen 4 Produkten unterstützt werden.

Die meisten *7-9 Plugins für gängige Anwendungen* und Guides werden von *SecureAuth IdP + core security* und *OpenOTP MFA Suite* bereitgestellt. *LinOTP* liefert einige Integrationsbeispiele und *Defender* wenige.

Defender ist wegen mangelnder Integrierbarkeit (*7-8-2*) kaum für Webapplikationen einsetzbar. Die übrigen Produkte scheinen bzgl. aller *7 Anwendungsbereiche* vergleichbar gut abzudecken.

[14] `https://www.oneidentity.com/products/cloud-access-manager/`

[15] In der Kostenschätzung unter Anforderung *6 Bedienkomfort* (Tabellen 5.10 und 5.11) wurde *Cloud Access Manager* nicht beachtet.

5.2. Produktvergleich und -auswahl

Nachdem nun alle 4 Hauptkandidaten gegen den gesamten Anforderungskatalog geprüft und nahezu alle der jeweils 109 Einzelanforderungen bewertet wurden, können die Gesamtbewertungen pro Produkt in Tabelle 5.14 berechnet werden.

Anforderung	Gewicht	Produktbewertung			
		LinOTP	*Defender*	*OpenOTP*	*SecureAuth*
0 Zwei-Faktor-Authentifizierung am LRZ	**4**	2.56	***0***	2.58	2.58
1 Rahmenanforderungen	2	2.73	2.91	2.73	3.0
2 Einrichtungsaufwand	3	2.5	2.32	2.43	2.38
3 Verwaltung, Wartung, Erweiterbarkeit	**4**	2.72	2.54	2.71	2.65
4 Sicherheitsniveau	3	2.27	2.18	2.29	2.38
5 Kosten(-effizienz)	2	2.38	2.54	2.44	1.75
6 Bedienkomfort	3	2.35	2.5	2.45	2.75
7 Anwendungsbereiche	**4**	2.84	***0***	2.89	2.89

Tabelle 5.14.: Ergebnis der Produktbewertung (toplevel)

Da *Defender* von *One Identity (Quest Software Inc.)* keine hinreichende *7-8-2 Integration in Webanwendungen* bietet, kann er das Szenario *7-8 Webapplikationen (Szenario 4)* nicht erfüllen. Er ist daher nicht für alle Anwendungsszenarien geeignet und besteht die Hauptauswahl nicht.

Die Zahlenwerte der Gesamtbewertung der restlichen Produkte sind so ähnlich, dass allein hierauf keinem Produkt der Vorzug gewährt werden kann. Eine Betrachtung der Toplevel-Anforderungen (*1* - *7*) lässt jedoch einige Unterschiede erkennen, die die während der Produktprüfung gewonnenen Eindrücke widerspiegeln.[16]

LinOTP und *OpenOTP MFA Suite* haben in allen Kategorien vergleichbare Bewertungen erzielt, was auch die geringen gewichteten euklidischen Distanzen zwischen den Bewertungen der Toplevel-Anforderungen in Tabelle 5.15 verdeutlichen.[17] Somit lässt sich *SecureAuth IdP + core security* den beiden open-source-Produkten gegenüberstellen, da jene sich auch in Aufbau und Technologie ähneln.

Während der Anforderungsbewertung wurden einige Stärken und Schwächen der einzelnen Produkte dargestellt. In den Toplevel-Bewertungen sichtbar werden v. a. die im Vergleich zu den beiden anderen Tools deutlich höheren Lizenzkosten von *SecureAuth IdP + core security*. Hierfür erhält man jedoch eine moderne graphische Oberfläche und Nutzer-Erfahrung. Was

[16] Eine andere Gewichtung der Anwendungsszenarien bzw. ein Filter auf Anforderungen einzelner Szenarien würde die Gesamtbewertung in vorliegenden Fall nicht nennenswert beeinflussen, da sich die Bewertungen der Anwendungsszenarien (*7-5*, *7-6*, *7-7* und *7-8*) von *LinOTP*, *OpenOTP MFA Suite* und *SecureAuth IdP + core security* nur marginal unterscheiden und *Defender* bereits aussortiert wurde.

[17] Gewichtete euklidische Distanz zwischen zwei Produktbewertungen p_1 und p_2:
$dist(p_1, p_2)_{euklid.} = \sqrt{\sum_{a \in Teilanf(Root)} Gewicht(a) * (Bewertung(a, p_1) - Bewertung(a, p_2))^2}$

	LinOTP	Defender	OpenOTP	SecureAuth
LinOTP	0.0	**5.718**	0.252	1.236
Defender	**5.718**	0.0	**5.804**	**5.919**
OpenOTP	0.252	**5.804**	0.0	1.189
SecureAuth	1.236	**5.919**	1.189	0.0

Tabelle 5.15.: Gewichtete euklidische Distanz zwischen den Bewertungen der Toplevel-Anforderungen

den *3-3-1 Funktionsumfang der Verwaltungssoftware* bzw. den Self-Service-Umfang betrifft, wurden die beiden open-source-Tools etwas höher bewertet als das Windows-Tool.

Mit den in Kapitel *3.4 Erstellung des Anforderungskataloges* gewählten Anforderungsgewichtungen gleichen sich diese Unterschiede aber weitgehend aus, weshalb man die in Tabelle 5.14 aufgeführten Gesamtergebnisse erhält. Für die gewählten Anwendungsszenarien scheinen alle drei Produkte etwa gleich gut geeignet.

Obwohl diese Produktbewertung kein eindeutiges Ergebnis lieferte und auf Basis des erstellten Anforderungskataloges formal wohl alle drei Produkte gleichermaßen geeignet sind, möchte diese Bachelorarbeit hier eine Empfehlung in Richtung der beiden open-source-Produkte aussprechen.

Als Betreiber einer homogenen Windows-Umgebung könnte man dem Windows-Tool *SecureAuth IdP + core security* vielleicht den Vorzug geben. Auch die Option für risikobasierte MFA (im teureren *prevent*-Lizenzpaket) ist attraktiv. Die deutlich höheren Lizenzkosten und die am LRZ häufiger eingesetzten Linux-Maschinen sprechen aber eher gegen *SecureAuth IdP + core security*. Bzgl. der MFA dürften die beiden anderen Produkte bald nachziehen. *OpenOTP MFA Suite* zeigt hier ja bereits Ansätze.

Da *LinOTP* und *OpenOTP MFA Suite* nur Standard-Webtechnologien für deren Interfaces benutzen, könnte eine modernere GUI z. B. im Rahmen eines Softwareentwicklungspraktikums von Studierenden beigesteuert werden. „API first“-Ansatz, offene Architektur und Modul Development Guides[18] ermöglichen eine leichte Integration in z. B. bestehende eigene Self-Service-Tools oder eine Erweiterung um individuelle Komponenten.

Keines der drei Produkte ist auf eine konkrete 2F-Technologie festgelegt. Eine App auf dem Smartphone oder ein FIDO bzw. YubiKey empfinden viele Anwender wahrscheinlich bequemer als einen OTP-Schlüsselanhänger. Wenn man sich am LRZ „global“ auf eine 2F-Technologie festlegen möchte, müsste die Wahl wohl auf manuell abzutippende OTPs fallen, da alles andere bei SSH-Konsolensitzungen schwierig einzugeben oder nicht vollständig inhouse abwickelbar ist. Für den Webportallogin wäre ein FIDO U2F-Key jedoch schneller und vielleicht komfortabler als manuelles Abtippen von OTPs. Die Freiheit pro User eine eigene Tokenart zu wählen bzw. dem Anwender diese Wahl zu überlassen, stellen alle Produkte bereit. Da manche Mitarbeiter des LRZs häufiger SSH-Verbindungen initialisieren, andere sich hauptsächlich in Webportale einloggen, kann so individuellen Wünschen der Mitarbeiter Rechnung getragen und der zweite Faktor deren Aufgabenfeld entsprechend gewählt werden.

Für die Wahl von *LinOTP* spricht die Möglichkeit Mobile-Push-Authentifizierung ohne externen Server abwickeln zu können und die lizenzkostenfreien 2FA-Zugänge für Webapplikationen mittels beliebiger OATH-OTP-Smartphone-App. Damit ließe sich 2FA kostengüns-

[18] z. B. `https://www.linotp.org/doc/latest/part-module-dev/index.html#linotp-development-guide`

tig für z. B. NeSSI, das Self-Service-Webportal für Netzverantwortliche des LRZs einrichten, oder gar als kostenfreie Dienstleistung für Kunden anbieten. Alternativ könnte man auch mit *RCDevs SA* über das nächstgrößere Volumenpaket von *OpenOTP MFA Suite*-Lizenzen verhandeln.

Die Marktanalyse hatte zum Ziel, basierend auf dem Konzeptvorschlag für ein 2FA-System am LRZ, derzeit erhältliche Produkte zu suchen und gegen den Anforderungskatalog zu prüfen. Von 75 betrachteten Produktkandidaten schieden 71 in der *5.1.1 Vorselektion* aus (davon 4 nur knapp). Die restlichen 4 Kandidaten wurden in der *5.1.2 Hauptauswahl* gegen den gesamten Anforderungskatalog bewertet und die Ergebnisse davon in Tabelle 5.14 zusammengefasst. Die Bewertungen fanden hauptsächlich auf Basis der Produktdokumentationen und einiger Telefonate bzw. Emails mit dem jeweiligen Support statt. Demoinstallationen bzw. Integrationstests fanden hierfür nicht statt. *5.2 Produktvergleich und -auswahl* stellte 3 der 4 Hauptkandidaten für den Einsatz am LRZ als geeignet heraus.

Dieser Vorschlag soll nun durch prototypische Implementierung überprüft werden.

6. Prototypische Umsetzung

Der Produktvorschlag des vorangegangenen Kapitels soll durch eine Testinstallation auf Rechnern des LRZs bestätigt werden. Um Arbeitsaufwand im Rahmen dieser Bachelorarbeit zu beschränken, soll nur eines der Produkte und dies lediglich bzgl. zweier Szenarien getestet werden. Der Produktvorschlag wird hiermit also nicht vollständig validiert.

Für die Installation wurde das Produkt *LinOTP* von *KeyIdentity GmbH* gewählt. Da es auch bei dem LRZ ähnlichen Institutionen wie bspw. dem Karlsruher Institut für Technologie[1] (KIT) seit Anfang 2017 zum Einsatz kommt, scheint diese Wahl vielversprechend.[SCC] Darüber hinaus ist *LinOTP* lizenzkostenfrei für Webapplikationen.

In Absprache mit den Betreuern dieser Arbeit wurden für die Testinstallation die Szenarien *Szenario 4: Login auf Webapplikationen des LRZs* (am Beispiel des LRZ-*idportals*) und *Szenario 5: Verlust von Authentisierungsinformationen und temporäre Gastzugänge* ausgewählt. Als zweite Faktoren sollen hier kostenlose OATH-TOTP-Softwaretokens dienen.

Der Übersicht halber werden Installationsvorgang und Ergebnisse der Tests hier nicht vollständig in voller Tiefe dokumentiert. Die Ausführungen dieses Kapitels konzentrieren sich auf nötige Abweichungen von der Standardinstallationsroutine, Anwendungskonfigurationen und in positiver oder negativer Hinsicht auffälligen Feststellungen.

Nach einer Beschreibung des Installations- bzw. Konfigurationsvorganges und dabei nötiger Anpassung von Anwendungen werden Testfälle sowie -vorgehen und schließlich die dabei gewonnenen Eindrücke beschrieben.

6.1. Installation und Konfiguration

LinOTP-Test-Instanz Die *LinOTP*-Test-Instanz, Version 2.10.1.1, deutsch, wurde auf `https://[linotpinstanz].lrz.de/`, einer frischen VM mit Betriebssystem *Debian 9 stretch* eingerichtet. Die Installation erfolgte aus den Paketquellen entsprechend der Anleitung in der *LinOTP*-Dokumentation unter `https://www.linotp.org/download.html` und dem anschließenden Installations-Wizard.

An zusätzlicher Konfiguration war lediglich die Anbindung eines *UserIDResolvers* notwendig. Hierzu diente der Test-LDAP-Server `ldaps://[testldap].lrz.de`, der im Dezember 2018 einen Dump der LDAP-Livedaten des LRZs vom Mai 2018 enthielt. Die Anbindung erfolgte über die Management-GUI im Abschnitt *LinOTP Konfiguration* > *UserIDResolver* durch von *Server-URI*, *BaseDN*, *BindDN* und *Bind Password* des Test-LDAP-Servers. Als *LinOTP*-`UID-Typ` war das LDAP-Attribut `uid`, anstatt des Standardwertes `entryUID` zu verwenden. Alle weiteren Attribute wurden für diesen Test nicht gemapt.

Anpassung des idportals Diesen Tests zu Grunde lag ein Klon[2] der in Perl geschriebenen Version des *idportals*, welcher auf `https://[idportalklon]/trunk/RELEASE/entry.pl`

[1] `https://www.kit.edu/`

[2] Kopie des *idportal*-trunk-Ordners, Stand 19.11.2018, 15 Uhr

ausgerollt wurde. Nötige Anpassungen beschränkten sich auf das Modul zur Validierung der eingegebenen Benutzerdaten (`PERLLIB/Valid.pm`). Nach der (unveränderten) Validierung von Benutzerkennung und Passwort wird ein HTTPS-Request[3] mit Nutzerkennung und eingegebenem OTP an die *LinOTP*-Test-Instanz gesendet. Die zugehörige Antwort im JSON-Format (siehe Anhang B.2) enthält das Ergebnis der Überprüfung des zweiten Faktors. Durch Konkatenation von Passwort mit OTP entfielen Erweiterungen der Eingabemaske. Das OTP wird als die letzten sechs Zeichen der Eingabe extrahiert.

Änderungen am Quellcode der Datei `PERLLIB/Valid.pm` sind in Anhang B.1 aufgeführt.[4] Hierbei handelt es sich lediglich um einen Prototypen, der von jedem Nutzer ein genau sechsstelliges Integer-OTP fordert. Für den Produktiveinsatz sollte die Validierung des zweiten Faktors dahingehend erweitert werden, dass die Art der für den Nutzer registrierten Tokens berücksichtigt (sechs-, acht- oder zehnstelliges OTP, mit oder ohne Nutzer-PIN, Tagesersatz-Passwort, Mobile-Push, FIDO etc.) und ein zweiter Faktor nur dann gefordert wird, wenn für den Nutzer ein 2FA-Token in der *LinOTP*-Instanz hinterlegt ist.

Die Validierung der HTTPS-Zertifikate wurde für den Prototypen deaktiviert, da die *LinOTP*-Test-Instanz nur ein selbst-signiertes Zertifikat verwendet, welches der eingesetzte Perl-HTTPS-Client nicht akzeptiert.

Vorbereitung temporärer Zugänge Für das *Szenario 5: Verlust von Authentisierungsinformationen und temporäre Gastzugänge* waren keine zusätzlichen Anpassungen oder Konfigurationen erforderlich.

6.2. Testdurchführung

Im Folgenden werden Durchführung und Ergebnisse der beiden betrachteten Testszenarien beschrieben.

6.2.1. 2FA für idportal-Nutzer

Für das Testszenario *Szenario 4: Login auf Webapplikationen des LRZs* (am Beispiel des LRZ-*idportals*) wurden im Rahmen der prototypischen Implementierung OATH-TOTP-Softwaretokens als zweite Faktoren gewählt. Über die *LinOTP*-GUI ließen sich solche generieren und manuell an Nutzer ausrollen[5], d. h. an deren LRZ-Kennung koppeln, die der *LinOTP*-Test-Instanz durch Anbindung an den LRZ-Test-LDAP-Server `ldaps://[testldap].lrz.de` bekannt waren. Die Seeds der Token wurden in einen TOTP-Software-Generator in Form einer Smartphone-App wie z. B. *Google Authenticator* via QR-Code eingebunden. Im Rahmen dieses Tests wurden zwei Kennungen exemplarisch mit einem zweiten Faktor ausgestattet. Der Login in das *idportal* war für die betroffenen Nutzer anschließend nur noch via Passwort konkateniert mit dem aktuellen, sechsstelligen TOTP möglich.

[3] API-Call: `https://[linotpinstanz].lrz.de/validate/check?user=[NUTZERKENNUNG]&pass=[TOKEN]`

[4] Die vollständige `PERLLIB/Valid.pm` liegt der elektronischen Abgabe bei. Der gesamte Quellcode des angepassten *idportal*-Klons ist in `gitlab.lrz.de:mmizani/linotp-idportal.git`, Commit `e999678c` zu finden.

[5] Siehe z. B. *LinOTP*-Quickstart-Guide, `https://linotp.org/doc/2.10.1.1/part-management/quickstart.html`

6.2.2. Temporäre Zugänge

Das Testszenario *Szenario 5: Verlust von Authentisierungsinformationen und temporäre Gastzugänge* gliedert sich in zwei Fälle auf, die ebenfalls am Beispiel des LRZ-*idportals* betrachtet wurden: Die Einrichtung temporärer Gastzugänge für bspw. Projektmitarbeiter sowie den Verlust des zweiten Faktors.

Gastzugänge Die Einrichtung von temporären Gastzugängen erfolgt analog zum initialen Rollout eines zweiten Faktors an einen Nutzer. Sie wurde oben unter *6.2.1 2FA für* idportal-*Nutzer* beschrieben. Zusätzlich kann hier nach Auswahl des Tokens in der *LinOTP*-Management-GUI (Abbildung 6.1) über *Tokeninfo* in der *Tokenansicht* eine Beschränkung der maximalen Zahl erfolgreicher Authentifizierungsvorgänge sowie ein Gültigkeitsintervall mit Start- und Endzeitpunkt festgelegt werden. Der Token besitzt somit nur temporäre Gültigkeit, die an den Aufenthalt des Gastes angepasst werden kann.

Abgelaufene Token werden von *LinOTP* bzw. dem angepassten *idportal* nicht akzeptiert.

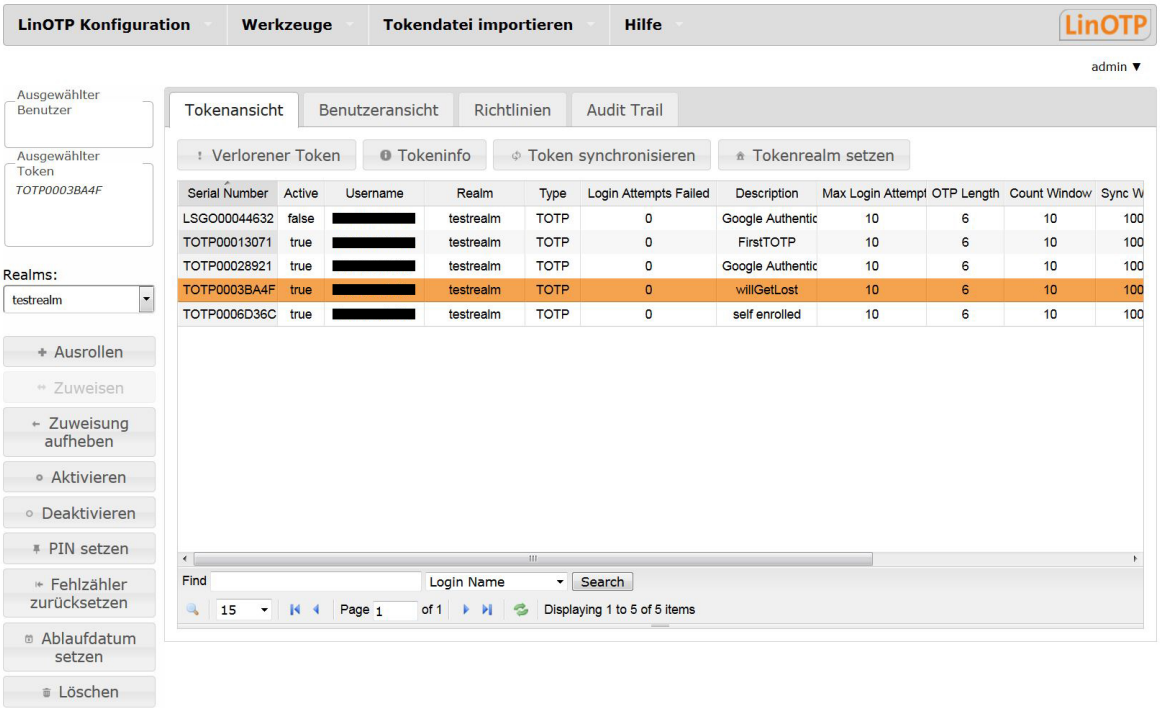

Abbildung 6.1.: *Tokenansicht* der *LinOTP*-Management-GUI

Verlust von Authentisierungsinformationen Beim Verlust von Authentisierungsinformationen lässt sich unterscheiden in einen dauerhaften Verlust des Tokens und eine kurzfristige Nichtverfügbarkeit, wenn der Nutzer seinen Token bspw. zu Hause oder vor der Dienstreise im Büro vergaß. Präventiv ließen sich einem Nutzer auch mehrere zweite Faktoren zuordnen.

Ist der Token dauerhaft verloren, kann er über die *LinOTP*-Management-GUI deaktiviert und gelöscht werden. Dem Nutzer wird dann bei nächster Gelegenheit auf regulärem Weg ein neuer zweiter Faktor ausgerollt.

Benötigt der Nutzer (nur) übergangsweise einen Ersatztoken, bietet sich die *LinOTP*-Funktion *Verlorener Token* an, die in der Management-GUI für den Admin oder Helpdesk markant positioniert wurde. (Siehe Abbildung 6.2) Sie entspricht einem Shortcut, der den ausgewählten Token deaktiviert und einen Ersatztoken in Form eines Passwortes, Email- oder SMS-Tokens mit festgelegtem Ablaufdatum generiert. Im Falle eines Passwortes ist dies dem Nutzer als Antwort auf seine Helpdeskanfrage mitzuteilen. Alternativ kann dem Nutzer auch über den regulären Weg ein gänzlich neuer Token (mit Ablaufdatum) ausgerollt werden.

Mit ersatzweise ausgestellten OTP-Tokens war ein Login in das angepasste *idportal* möglich. Ersatztoken in Form eines Passwortes wurden nur über die *LinOTP*-API akzeptiert[6], da das prototypisch angepasste *idportal* wie oben erwähnt sechsstellige Integer-OTPs fordert.

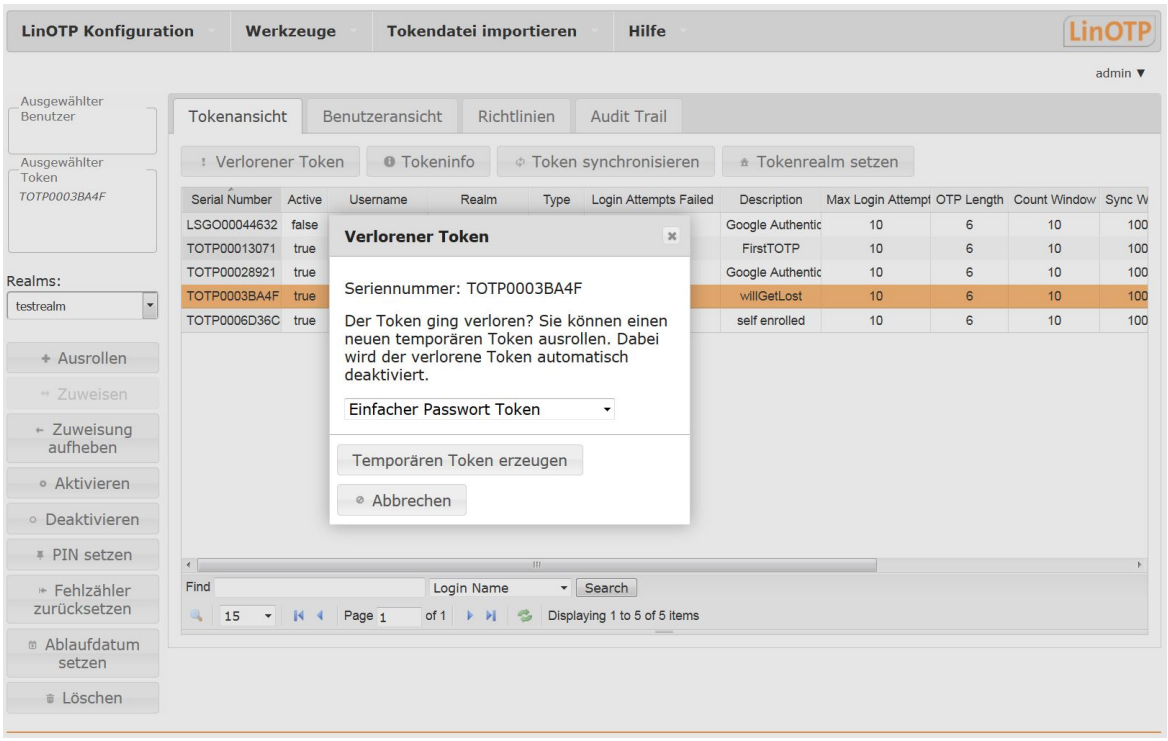

Abbildung 6.2.: *Lost Token*-Szenario der *LinOTP*-Management-GUI

Szenario 5: Verlust von Authentisierungsinformationen und temporäre Gastzugänge bedeutet für den *LinOTP*-Admin bzw. -Helpdesk durch markant positionierte Shortcuts verhältnismäßig geringen manuellen Aufwand – eine hinreichende Identifizierung des anfragenden Nutzers über andere Faktoren vorausgesetzt. Ein gewisser Schutz vor Missbrauch lässt sich durch die *LinOTP*-OTP-PIN-Funktion erzielen. Hiermit lässt sich zu jedem Token eine PIN festlegen, die bei während der Authentifizierung dem OTP voranzustellen ist. Die *Verlorener Token*-Funktion behält diese PIN dann bei und überträgt sie auf den Ersatztoken.

Würde man zum Login in das *LinOTP*-Self-Service-Portal keinen zweiten Faktor fordern und den Nutzern dort erlauben, selbst Token auszurollen, käme man im betrachteten Test-

[6] Der entsprechende API-Call `https://[linotpinstanz].lrz.de/validate/check?user=[NUTZERKENNUNG]&pass=[(OTP|Ersatztoken)]` gibt das Ergebnis im JSON-Format zurück. (Siehe Anhang B.2)

fall ohne Interaktion mit dem Helpdesk aus. Der Nutzer könnte sich bei Verlust schlicht selbst einen neuen Token ausrollen. Der Einsatz von 2FA wäre somit jedoch ohne weitere Maßnahmen wie bspw. einer Aktivierung des selbst ausgerollten Tokens durch den Admin nahezu wertlos.

6.3. Gewonnene Eindrücke

Insgesamt konnte *LinOTP* im hier untersuchten Testszenario überzeugen. Mit Hilfe der Dokumentation gelang die Installation aus den Paketquellen auf einer *Debian*-VM ebenso wie die LDAP-Anbindung zügig. Eine Produktivinstanz dürfte jedoch noch einigen Konfigurationsaufwand erfordern. Die Management-GUI wirkt übersichtlich und eine Integration von 2FA in Webapplikationen wie das *idportal* gelang über *LinOTP*-API-Calls (in diesen einfachen Tests) reibungslos.

Das quelloffene *LinOTP* bietet viele Personalisierungsmöglichkeiten, welche z. B. im Self-Service-Portal zum Tragen kommen. Neben dem GUI-Design lassen sich hier auch verschiedene Funktionalitäten freischalten, welche – wie oben erwähnt – die Sicherheit des 2FA-Systems ggf. unterwandern könnten. Ein Konzept zum Einsatz des Self-Service-Portals scheint also erforderlich.

Leicht negativ anzumerken sind stellenweise kleinere Tippfehler bzw. sprachliche Inkonsistenzen in (zumindest der deutschen Version) der GUI: Unter *Tokenkonfiguration* bspw. lautet die Überschrift im Reiter *ORCA2-Token* „OCRA2-Token Einstellungen“; die Überschrift im Reiter *ORCA-Token* „OCRA**2**-Token**-**Einstellungen“. Derartige Auffälligkeiten wurden dem Hersteller-Support gemeldet.

Die im Rahmen dieser Bachelorarbeit durchgeführten Tests konzentrierten sich auf wesentliche Aspekte zweier der in *3.1 Szenarien der Authentifizierung am LRZ* definierten Szenarien. Dabei wurde nur das Produkt *LinOTP* getestet und der ebenfalls vorgeschlagene Kandidat *OpenOTP MFA Suite* nicht betrachtet.

7. Zusammenfassung und Ausblick

Dieses Kapitel fasst die Ergebnisse der Arbeit zusammen und diskutiert sie kurz.

Zusammenfassung Ziel dieser Bachelorarbeit war es, ein geeignetes System zur Stärkung der aktuell am Leibniz-Rechenzentrum eingesetzten Authentifizierungsverfahren durch einen zweiten Faktor vorzuschlagen. Dazu wurden zuerst Arten, Methoden und Herausforderungen der Authentifizierung aufgezeigt und somit der Einsatz von Zwei-Faktor-Authentifizierung motiviert. Darauf folgte mit Kapitel 2 eine vergleichende Übersicht möglicher Besitzfaktoren (OTP, OATH, CR, FIDO) und der Person anhängender Merkmale. Auch Möglichkeiten des Einbezugs von Smartphones in den Authentifizierungsvorgang wurden untersucht.

Nach der Identifikation typischer Szenarien der Authentifizierung am LRZ in Kapitel 3.1 wurden daraus Anforderungen an ein MFA-System am LRZ abgeleitet, entsprechend sortiert, gewichtet und zu einem Katalog mit letztlich 109 Einzelanforderungen zusammengesetzt. Anhand dieses Anforderungskataloges wurde in Kapitel 4 eine technologische Vorauswahl getroffen und ein Rahmenkonzept zur MFA am LRZ erarbeitet, wonach hauptsächlich OTP-, FIDO U2F- und Mobile-Push-Verfahren als geeignet angesehen werden.

Im ersten Schritt der in Kapitel 5 anschließenden Suche nach auf dem Markt verfügbaren Produkten, welche die Techniken des Rahmenkonzeptes implementieren, konnten mittels einiger als Ausschlusskriterien definierter Anforderungen des Kataloges 71 der 75 gefundenen Kandidaten vorab aussortiert werden. Die übrigen 4 Produkte (*KeyIdentity GmbH*: *LinOTP*, *RCDevs SA*: *OpenOTP MFA Suite*, *One Identity (Quest Software Inc.)*: *Defender*, *SecureAuth Corporation*: *SecureAuth IdP + core security*) wurden im zweiten Schritt der Produktauswahl dem Bewertungsschema gemäß gegen den gesamten Anforderungskatalog geprüft und bewertet. *Defender* konnte hierbei wegen mangelnder Unterstützung für 2FA in Webapplikationen nicht überzeugen. Die beiden open-source Produkte *LinOTP* und *OpenOTP MFA Suite* sowie das Windows-Tool *SecureAuth IdP + core security* liegen in der Gesamtbewertung zahlenmäßig zwar gleich auf, nach Diskussion in 5.2 schlägt die vorliegende Bachelorarbeit jedoch vor, zum Einsatz am LRZ eines der beiden open-source Produkte zu präferieren. Alle 3 Produkte scheinen für jedes der definierten Anwendungsszenarien geeignet, erfüllen das Rahmenkonzept, unterstützen verschiedene Arten von zweiten Faktoren und bieten eine umfangreiche Verwaltungssoftware inklusive Nutzer-Self-Service-Funktionen.

Das Produkt *LinOTP* wurde in Kapitel 6 prototypisch implementiert und konnte beim Test gegen zwei der definierten Szenarien überzeugen.

Jedes der drei Produkte unterstützt eine Liste an verschiedenen Arten von zweiten Faktoren. Die Wahl eines konkreten zweiten Faktors kann also dem einzelnen Nutzer überlassen oder durch eine LRZ-weite Richtlinie festgelegt werden. Entscheidungsgrundlagen hierfür wurden ausführlich in Kapitel 2.5, in Kapitel 4 und zuletzt in Kapitel 5.2 diskutiert.

Einschränkungen Trotz eines nicht geringen Rechercheaufwands kann die Produktkandidatenliste nicht als vollständig gelten. Auch wurde die Vorselektion der Kandidaten maßgeblich durch Wahl der Anwendungsszenarien bzw. Ausschlusskriterien geprägt: Keine Möglichkeit zum on-premise-Hosting oder fehlende Integration in SSH oder MacOS-Desktoplogin waren wohl die häufigsten Ursachen zum Ausschluss eines Produktes. Mit Verzicht auf eines dieser Kriterien wären auch einige andere (und möglicherweise dann geeignetere) Produkte in die Hauptauswahl aufgenommen worden.

Die Produktbewertungen dieser Arbeit basieren hauptsächlich auf den technischen Dokumentationen der jeweiligen Hersteller und Telefonaten mit deren Support. Sie wurden nur teilweise durch Integrationstests am LRZ validiert. Die Ergebnisse der Produktauswahl wurden vom Anforderungskatalog (an dessen Erstellung kein Administrator des LRZs direkt beteiligt war) und v. a. von den darin gewählten Gewichtungen der Anforderungen bestimmt. Sollte eine Folgearbeit den Katalog erweitern, anders gewichten oder eine andere Bewertungsformel ansetzen wollen, kann dies zu anderen Ergebnissen führen.[1] Des Weiteren könnte eine Folgearbeit sich mit ausführlicheren Integrationstests am LRZ der hier vorgeschlagenen Produkte befassen und somit die Einführung eines MFA-Systems weiter vorbereiten. Auch die 4 in der Produktauswahl dieser Arbeit nur knapp durchgefallenen Kandidaten sollten berücksichtigt und auf zwischenzeitliche Verbesserungen geprüft werden. [2]

Die Integration der MFA in konkrete (Web-)Anwendungen des LRZs wie bspw. *idportal*, *NeSSi*, *LRZ Sync+Share* aber auch z. B. dem *SuperMUC*-Zugang könnten als studentische Softwareentwicklungspraktika ausgegeben werden. Weitere Arbeiten für das LRZ könnten sich mit Single-Sign-On-Lösungen via SAML in Kombination mit dem eingesetzten MFA-System befassen, Erweiterungen zur kontext- oder risikobasierten Multi-Faktor-Authentifizierung entwickeln oder Verfahren zur Authentifizierung mit Mobile-Push als zweitem Faktor entwerfen, die keine externen Server involvieren.

Mehr-Faktor-Authentifizierung ist für Nutzer und Betreiber immer mit zusätzlichem Aufwand und Kosten verbunden. In Bezug auf die einleitend dargestellten Unzulänglichkeiten einfacher Authentifizierungsverfahren scheint die Einführung von MFA jedoch lohnend. Die Akzeptanz in der Nutzerschaft sollte im Change-Management nicht vernachlässigt werden. Der Einsatz von zweiten Faktoren mit hohem Nutzungskomfort und schrittweises Rollout könnten hierfür förderlich sein.

Im Verlauf dieser Bachelorarbeit wurden neue Standards wie bspw. FIDO2 veröffentlicht. Auch wenn der Themenbereich der Zwei- bzw. Mehr-Faktor-Authentifizierung keinesfalls jung ist, sind hierin sicherlich noch einige Entwicklungen in Richtung Benutzerfreundlichkeit und Sicherheit zu erwarten.

[1] Zur Erleichterung dessen sind Anforderungskatalog, Gewichtungen und Bewertung pro Produkt der elektronischen Abgabe in SQL-Form beigefügt.

[2] Auch andere Einrichtungen als das LRZ können von der vorliegenden Bachelorarbeit profitieren. Die hier gewählten Szenarien scheinen keine untypischen zu sein, da viele der betrachteten Produkte ein oder mehrere derer implementieren. Der hieraus abgeleitete Anforderungskatalog kann als Vorlage genommen und individuell angepasst werden. Die vorliegende Kandidatenliste reduziert den Rechercheaufwand einer Marktanalyse.

Danksagung

Ich danke meinen sehr geduldigen und hilfsbereiten Betreuern Stefan Metzger und Jule Ziegler sowie Herrn Ralf Ebner vom Leibniz-Rechenzentrum, der die prototypische Implementierung begleitete. Vielen Dank auch meiner Mutter für orthographische Anmerkungen und den Freunden, die mich während des Verfassens mit aufmunternden Worten unterstützten.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[AM09] Alghathbar, Khaled ; Mahmoud, Hanan A.: Noisy password scheme: A new one time password system. In: *Electrical and Computer Engineering, 2009. CCECE'09. Canadian Conference on* IEEE, 2009, S. 841–846

[AR16] Ahlbom, Per ; Richter, Martin: Investigating Open Source Alternatives for an Electronic Identity System. (2016)

[AZEH09] Aloul, Fadi ; Zahidi, Syed ; El-Hajj, Wassim: Two factor authentication using mobile phones. In: *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on* IEEE, 2009, S. 641–644

[BCZ17] Buriro, Attaullah ; Crispo, Bruno ; Zhauniarovich, Yury: Please hold on: Unobtrusive user authentication using smartphone's built-in sensors. In: *Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on* IEEE, 2017, S. 1–8

[Ber18] Berger, Daniel: Mozilla, Google und Microsoft unterstützen WebAuthn und damit Logins ohne Passwörter. In: *heise online* (2018), apr. `https://heise.de/-4017525`. – Zuletzt abgerufen am 29.10.2018

[BgM+17] Brenner, Michael ; gentschen Felde, Nils ; Metzger, Stefan ; Reiser, Helmut ; Schaaf, Thomas: *Praxisbuch ISO-IEC 27001: Management der Informationssicherheit und Vorbereitung auf die Zertifizierung*. Carl Hanser Verlag GmbH & Company KG, 2017. – ISBN 9783446452602

[BR02] Brenner, Michael ; Radisic, Igor: A Criteria Catalog Based Methodology for Analyzing Service Management Processes. (2002)

[BSI13a] *4 Glossar und Begriffsdefinitionen*. Version: 2013. `https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html` Zuletzt abgerufen am 29.10.2018

[BSI13b] Bundesamt für Sicherheit in der Informationstechnik: M 4.113 Nutzung eines Authentisierungsservers bei Remote-Access-VPNs. 2013. – BSI Grundschutzkatalog

[BSI13c] Bundesamt für Sicherheit in der Informationstechnik: M 4.133 Geeignete Auswahl von Authentikationsmechanismen. 2013. – BSI Grundschutzkatalog

[BSI13d] Bundesamt für Sicherheit in der Informationstechnik: M 4.176 Auswahl einer Authentisierungsmethode für Webangebote . 2013. – BSI Grundschutzkatalog

[BSI13e] Bundesamt für Sicherheit in der Informationstechnik: M 4.250 Auswahl eines zentralen, netzbasierten Authentisierungsdienstes. 2013. – BSI Grundschutzkatalog

[BSI13f] Bundesamt für Sicherheit in der Informationstechnik: M 4.392 Authentisierung bei Webanwendungen. 2013. – BSI Grundschutzkatalog

[BSI13g] Bundesamt für Sicherheit in der Informationstechnik: M 4.415 Sicherer Betrieb der biometrischen Authentisierung unter Windows. 2013. – BSI Grundschutzkatalog

[BSI13h] Bundesamt für Sicherheit in der Informationstechnik: M 5.34 Einsatz von Einmalpasswörtern. 2013. – BSI Grundschutzkatalog

[BSI14a] Bundesamt für Sicherheit in der Informationstechnik: M 4.441 Multifaktor-Authentisierung für den Cloud-Benutzerzugriff. 2014. – BSI Grundschutzkatalog

[BSI14b] Bundesamt für Sicherheit in der Informationstechnik: M 4.456 Authentisierung bei Web-Services. 2014. – BSI Grundschutzkatalog

[BSI16] Bundesamt für Sicherheit in der Informationstechnik: M 2.164 Auswahl eines geeigneten kryptographischen Verfahrens. 2016. – BSI Grundschutzkatalog

[BSI18a] Bundesamt für Sicherheit in der Informationstechnik: Kryptographische Verfahren: Empfehlungen und Schlüssellängen (BSI TR-02102-1). 2018. – Technische Richtlinie

[BSI18b] Bundesamt für Sicherheit in der Informationstechnik: Kryptographische Verfahren: Empfehlungen und Schlüssellängen – Teil 4 - Verwendung von Secure Shell (SSH) (BSI TR-02102-4). 2018. – Technische Richtlinie

[DCDFN13] De Cristofaro, Emiliano ; Du, Honglu ; Freudiger, Julien ; Norcie, Greg: A comparative usability study of two-factor authentication. In: *arXiv preprint arXiv:1309.5344* (2013)

[DP00] Dhamija, Rachna ; Perrig, Adrian: Déjà Vu: a user study using images for authentication. In: *Proceedings of the 9th conference on USENIX Security Symposium-Volume 9* USENIX Association, 2000, S. 4–4

[DZZ13] Draffin, Benjamin ; Zhu, Jiang ; Zhang, Joy: Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction. In: *International Conference on Mobile Computing, Applications, and Services* Springer, 2013, S. 184–201

[EAK11] Eldefrawy, Mohamed H. ; Alghathbar, Khaled ; Khan, Muhammad K.: OTP-based two-factor authentication using mobile phones. In: *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on* IEEE, 2011, S. 327–331

[Eck13] Eckert, Claudia: *IT-Sicherheit: Konzepte-Verfahren-Protokolle*. Walter de Gruyter, 2013

[Eik11] Eikenberg, Ronald: RSA tauscht nach Hack bis zu 40 Millionen SecurID-Tokens aus. In: *heise online* (2011), Jun. `https://heise.de/-1256298`. – Zuletzt abgerufen am 29.10.2018

[fid17a] FIDO Alliance: FIDO UAF Architectural Overview v1.1. Version: Februar 2017. `https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/fido-uaf-overview-v1.1-ps-20170202.pdf`. 2017. – Forschungsbericht

[fid17b] FIDO Alliance: Universal 2nd Factor (U2F) Overview v1.2. Version: April 2017. `https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-overview-v1.2-ps-20170411.pdf`. 2017. – Forschungsbericht

[gar18] *SecureAuth's ranking in Gartner peer insights*. `https://www.gartner.com/reviews/market/access-management/vendor/secureauth-core-security`. Version: Oktober 2018. – Zuletzt abgerufen am 29.10.2018

[GND17] Galdi, Chiara ; Nappi, Michele ; Dugelay, Jean-Luc: Secure User Authentication on Smartphones via Sensor and Face Recognition on Short Video Clips. In: *International Conference on Green, Pervasive, and Cloud Computing* Springer, 2017, S. 15–22

[HA94] Haller, N. ; Atkinson, R.: On Internet Authentication / RFC Editor. RFC Editor, October 1994 (1704). – RFC. – ISSN 2070–1721

[Hal95] Haller, N.: The S/KEY One-Time Password System / RFC Editor. RFC Editor, February 1995 (1760). – RFC. – ISSN 2070–1721

[Har12] Hardt, D.: The OAuth 2.0 Authorization Framework / RFC Editor. Version: October 2012. `http://www.rfc-editor.org/rfc/rfc6749.txt`. RFC Editor, October 2012 (6749). – RFC. – ISSN 2070–1721. – `http://www.rfc-editor.org/rfc/rfc6749.txt`

[HM96] Haller, N. ; Metz, C.: A One-Time Password System / RFC Editor. RFC Editor, May 1996 (1938). – RFC. – ISSN 2070–1721

[HMIS98] Haller, Neil ; Metz, Craig ; II, Philip J. N. ; Straw, Mike: A One-Time Password System / RFC Editor. Version: February 1998. `http://www.rfc-editor.org/rfc/rfc2289.txt`. RFC Editor, February 1998 (2289). – RFC. – ISSN 2070–1721. – `http://www.rfc-editor.org/rfc/rfc2289.txt`

[HNM14] Hamdare, Safa ; Nagpurkar, Varsha ; Mittal, Jayashri: Securing SMS based one time password technique from man in the middle attack. In: *arXiv preprint arXiv:1405.4828* (2014)

[KBC97] Krawczyk, Hugo ; Bellare, Mihir ; Canetti, Ran: HMAC: Keyed-Hashing for Message Authentication / RFC Editor. Version: February 1997. `http://www.rfc-editor.org/rfc/rfc2104.txt`. RFC Editor, February 1997 (2104). – RFC. – ISSN 2070–1721. – `http://www.rfc-editor.org/rfc/rfc2104.txt`

[KD15] Kaur, Navpreet ; Devgan, Mandeep: A Comparative Analysis of Various Multistep Login Authentication Mechanisms. In: *International Journal of Computer Applications* 127 (2015), Nr. 9

[KDB16] Kaur, Navpreet ; Devgan, Mandeep ; Bhushan, Shashi: Robust login authentication using time-based OTP through secure tunnel. In: *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on* IEEE, 2016, S. 3222–3226

[KMB17] Kogan, Dmitry ; Manohar, Nathan ; Boneh, Dan: T/Key: Second-Factor Authentication From Secure Hash Chains. In: *arXiv preprint arXiv:1708.08424* (2017)

[Lam81] Lamport, Leslie: Password authentication with insecure communication. In: *Communications of the ACM* 24 (1981), Nr. 11, S. 770–772

[LRZ] Leibniz-Rechenzentrum der bayerischen Akademie der Wissenschaften: LRZ Image Broschüre. `https://www.lrz.de/wir/lrz-flyer/lrz_Image_broschuere.pdf`. – Forschungsbericht. – Zuletzt abgerufen am 29.10.2018

[LRZ15a] Leibniz-Rechenzentrum der bayerischen Akademie der Wissenschaften: Active Directory Services im MWN. Version: 2015. `https://www.lrz.de/services/client-server/activedirectory/`. 2015. – Forschungsbericht. – Zuletzt abgerufen am 29.10.2018

[LRZ15b] Leibniz-Rechenzentrum der bayerischen Akademie der Wissenschaften: Verschlüsselung, digitale Signaturen, Zertifikate. Version: 2015. `https://www.lrz.de/services/pki/einf/#auth-autor`. 2015. – Forschungsbericht. – Zuletzt abgerufen am 29.10.2018

[LRZ16] Leibniz-Rechenzentrum der bayerischen Akademie der Wissenschaften: LRZ Jahresbericht 2016. Version: 2016. `https://www.lrz.de/wir/berichte/JB/JBer2016.pdf`. 2016. – Forschungsbericht. – Zuletzt abgerufen am 10.07.2018

[MAM95] McDonald, Daniel ; Atkinson, Randall J. ; Metz, Craig: One-Time Passwords in Everything (OPIE): Experiences with Building and Using Strong Authentication. In: *USENIX Security Symposium*, 1995

[MBH+05] M'Raihi, D. ; Bellare, M. ; Hoornaert, F. ; Naccache, D. ; Ranen, O.: HOTP: An HMAC-Based One-Time Password Algorithm / RFC Editor. RFC Editor, December 2005 (4226). – RFC. – ISSN 2070–1721

[MBSS13] Mulliner, Collin ; Borgaonkar, Ravishankar ; Stewin, Patrick ; Seifert, Jean-Pierre: SMS-based one-time passwords: attacks and defense. In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* Springer, 2013, S. 150–159

[MGS10] McCallister, Erika ; Grance, Tim ; Scarfone, Karen: Identifiable Information (PII). (2010)

[mic18] *Windows Hello and FIDO2 Security Keys enable secure and easy authentication for shared devices.* https://www.microsoft.com/en-us/microsoft-365/blog/2018/04/17/windows-hello-fido2-security-keys/. Version: April 2018. – Zuletzt abgerufen am 29.10.2018

[MMPR11] M'Raihi, D. ; Machani, S. ; Pei, M. ; Rydell, J.: TOTP: Time-Based One-Time Password Algorithm / RFC Editor. Version: May 2011. http://www.rfc-editor.org/rfc/rfc6238.txt. RFC Editor, May 2011 (6238). – RFC. – ISSN 2070–1721. – http://www.rfc-editor.org/rfc/rfc6238.txt

[MRB+11] M'Raihi, D. ; Rydell, J. ; Bajaj, S. ; Machani, S. ; Naccache, D.: OCRA: OATH Challenge-Response Algorithm / RFC Editor. RFC Editor, June 2011 (6287). – RFC. – ISSN 2070–1721

[NIS18] National Institute of Standards and Technology: Digital Identity Guidelines – Authentication and Lifecycle Management (NIST.SP.800-63b). Version: 2018. https://doi.org/10.6028/NIST.SP.800-63b. 2018. – NIST Special Publication

[oat05] Initiative for Open AuTHentication (OATH): Attacks on SHA-1. Version: 2005. https://openauthentication.org/wp-content/uploads/2015/09/TechnicalWhitePaper.pdf. 2005. – Forschungsbericht

[oat07] Initiative for Open AuTHentication (OATH): OATH Reference Architecture, Release 2.0. Version: 2007. https://openauthentication.org/wp-content/uploads/2015/09/ReferenceArchitectureVersion2.pdf. 2007. – Forschungsbericht

[oat15] Initiative for Open AuTHentication (OATH): An Industry Roadmap for Open Strong Authentication. Version: 2015. https://openauthentication.org/wp-content/uploads/2015/09/AnIndustryRoadmapforOpenStrongAuthentication.pdf. 2015. – Forschungsbericht

[O'G03] O'Gorman, Lawrence: Comparing passwords, tokens, and biometrics for user authentication. In: *Proceedings of the IEEE* 91 (2003), Nr. 12, S. 2021–2040

[one18] *Gartner has named One Identity a Leader in its February 2018 MQ for Identity Governance and Administration.* https://www.oneidentity.com/whitepaper/gartner-has-named-one-identity-a-leader-in-its-february-2018-mq-for-id8131105/. Version: Februar 2018. – Zuletzt abgerufen am 29.10.2018

[PS10] Paterson, Kenneth G. ; Stebila, Douglas: One-time-password-authenticated key exchange. In: *Australasian Conference on Information Security and Privacy* Springer, 2010, S. 264–281

[RR12] Radha, V ; Reddy, D H.: A survey on single sign-on techniques. In: *Procedia Technology* 4 (2012), S. 134–139

[RWRS00] Rigney, C. ; Willens, S. ; Rubens, A. ; Simpson, W.: Remote Authentication Dial In User Service (RADIUS) / RFC Editor. Version: June 2000. http://www.rfc-editor.org/rfc/rfc2865.txt. RFC Editor, June 2000 (2865). – RFC. – ISSN 2070–1721. – http://www.rfc-editor.org/rfc/rfc2865.txt

[SCC] *SCCnews 02/2017*. http://www.scc.kit.edu/downloads/oko/SCCnews_02_2017_web.pdf. – Zuletzt abgerufen am 21.11.2018

[SD07] Suh, G E. ; Devadas, Srinivas: Physical Unclonable Functions for Device Authentication and Secret Key Generation. (2007)

[Shi07] Shirey, R.: Internet Security Glossary, Version 2 / RFC Editor. Version: August 2007. http://www.rfc-editor.org/rfc/rfc4949.txt. RFC Editor, August 2007 (4949). – RFC. – ISSN 2070–1721. – http://www.rfc-editor.org/rfc/rfc4949.txt

[Sim96] Simpson, William A.: PPP Challenge Handshake Authentication Protocol (CHAP) / RFC Editor. Version: August 1996. http://www.rfc-editor.org/rfc/rfc1994.txt. RFC Editor, August 1996 (1994). – RFC. – ISSN 2070–1721. – http://www.rfc-editor.org/rfc/rfc1994.txt

[Smi01] Smith, Richard E.: *Authentication: from passwords to public keys*. Addison-Wesley Longman Publishing Co., Inc., 2001

[SPK16] Sivakorn, Suphannee ; Polakis, Jason ; Keromytis, Angelos D.: I'm not a human: Breaking the Google reCAPTCHA. (2016)

[SYJ+11] Shi, Weidong ; Yang, Jun ; Jiang, Yifei ; Yang, Feng ; Xiong, Yingen: Senguard: Passive user identification on smartphones using multiple sensors. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on* IEEE, 2011, S. 141–148

[ver17] *2017 Data Breach Investigations Report*. https://www.verizonenterprise.com/resources/reports/rp_DBIR_2017_Report_en_xg.pdf. Version: 2017. – Zuletzt abgerufen am 29.10.2018

[ver18] *2018 Data Breach Investigations Report*. https://www.verizonenterprise.com/resources/reports/rp_DBIR_2018_Report_execsummary_en_xg.pdf. Version: 2018. – Zuletzt abgerufen am 29.10.2018

[Wü18] Würz, Robin: *Aufbau einer Open Badges Infrastruktur*, Ludwig–Maximilians–Universität München, Diplomarbeit, Jan 2018. http://www.nm.ifi.lmu.de/pub/Fopras/wuer18/

[w3c18] World Wide Web Consortium (W3C): Web Authentication: An API for accessing Public Key Credentials Level 1. Version: August 2018. https://www.w3.org/TR/webauthn/. 2018. – Forschungsbericht. – Zuletzt abgerufen am 29.10.2018

[Web12] Weber, Thomas: *Authentifizierung mit dem neuen Personalausweis (nPA)*, Ludwig–Maximilians–Universität München, Diplomarbeit, Jul 2012. http://www.nm.ifi.lmu.de/common/pub/Fopras/

[WWSB06] Wiedenbeck, Susan ; Waters, Jim ; Sobrado, Leonardo ; Birget, Jean-Camille: Design and evaluation of a shoulder-surfing resistant graphical password scheme. In: *Proceedings of the working conference on Advanced visual interfaces* ACM, 2006, S. 177–184

[YKN09] Yamamoto, Takumi ; Kojima, Yuko ; Nishigaki, Masakatsu: A Shoulder-Surfing-Resistant Image-Based Authentication System with Temporal Indirect Image Selection. In: *Security and Management*, 2009, S. 188–194

[yub18] *The Yubico Pluggable Authentication Module (PAM) extends secure hardware-backed YubiKey two-factor authentication to existing Linux/Unix user authentication infrastructure.* `https://www.yubico.com/why-yubico/for-business/computer-login/linux/`. Version: 2018. – Zuletzt abgerufen am 29.10.2018

[ZWWZ13] Zhu, Jiang ; Wu, Pang ; Wang, Xiao ; Zhang, Joy: Sensec: Mobile security through passive sensing. In: *Computing, Networking and Communications (ICNC), 2013 International Conference on* IEEE, 2013, S. 1128–1133

A. Verwendete Abkürzungen

- 2F – Zweiter Faktor
- 2FA – Zwei-Faktor-Authentifizierung
- 3DES – Tripple-DES
- AAA – Authentifizierung, Autorisierung, Accounting
- AD – Active Directory
- ADFS – Active Directory Federation Services
- AES – Advanced Encryption Standard
- AGPL – Affero General Public License
- API – Application Programming Interface
- BAC – Basic Access Control
- CAPTCHA – Completely Automated Public Turing test to tell Computers and Humans Apart
- CHAP – Challenge Handshake Authentication Protocol
- CR – Challenge-Response
- CSS – Cascading Style Sheet
- CTAP – Client to Authentication Protocol
- DES – Data Encryption Standard
- DFN – Deutsches Forschungsnetz
- EAC – Extended Access Control
- FAQ – Frequently Asked Questions
- FIDO – Fast IDentity Online
- FIDO U2F – Universal Second Factor
- FIDO UAF – Universal Authentication Framework
- FTP – File Transfer Protocol
- GPL – GNU General Public License
- GSM – Global System for Mobile Communications
- GUI – Graphical User Interface
- HMAC – keyed-Hash Message Authentication Code
- HOTP – HMAC-based One-time Password
- HTTP – Hypertext Transfer Protocol
- HTTPS – Hypertext Transfer Protocol Secure
- IAM – Identity and Access Management
- IDaaS – Identity as a Service
- IdP – Identity Provider
- IIS – Microsoft Internet Information Services
- IP – Internet Protocol
- IT – InformationsTechnik

- JSON – JavaScript Object Notation
- KIT – Karlsruher Institut für Technologie
- LDAP – Lightweight Directory Access Protocol
- LRZ – Leibniz-Rechenzentrum
- LTE – Long Term Evolution
- MFA – Multi-Faktor-Authentifizierung
- MNO – Mobile Network Operator
- MS – Microsoft
- MWN – Münchner Wissenschaftsnetz
- NeSSI - Network Self Service Interface des LRZs
- NFC – Near Field Communication
- NTP – Network Time Protokoll
- nPA – neuer Personalausweis (und die damit eingeführten Authentisierungsmöglichkeiten)
- OATH – Open Authentication
- OAuth – Open Authorization
- OPIE – One-time Passwords In Everything
- ORCA – OATH Challenge Response Algorithm
- OTP – One Time Password
- PACE – Password Authenticated Connection Establishment
- PAM – Plugable Authentication Module
- PIN – Persönliche IdentifikationsNummer
- PKI – Public Key Infrastructure
- PPP – Point-to-Point Protocol
- QR-Code – Quick Response Code
- RADIUS – Remote Authentication Dial-In User Service
- RDP – Remote Desktop Protocol
- REST – Representational State Transfer
- RFC – Request For Comments
- RFID – Radio Frequency Identification
- RSA – ein nach den Entwicklern Rivest, Shamir und Adleman benanntes asymmetrisches kryptographisches Verfahren
- SaaS – Software as a Service
- SAML – Security Assertion Markup Language
- SDK – Software Development Kit
- SHA – Secure Hash Algorithm
- SMS – Short Message Service
- SQL – Structured Query Language
- SSH – Secure Shell
- SSL – Secure Sockets Layer
- SSO – Single Sign On
- SUBROSA – Simple User Based Resource Oriented Segmentation Architecture
- TAN – TransAktionsNummer
- TLS – Transport Layer Security

- TOTP – Time-based One-time Password
- UMTS – Universal Mobile Telecommunications Systems
- VM – Virtual Machine
- VPN – Virtual Private Network
- WLAN – Wireless Local Area Network

B. Prototypische Implementierung

B.1. Prototypische Anpassung des idportals

```
                                        [...]
    //Zusätzliche Libraries und globale Variablen
208 use JSON; # [mm]: Parsen von LinOTP-Antworten im JSON-Format
209 my $OTP_LENGTH = 6; # [mm]: Laenge von LinOTP-OTPs
210 use LWP::Simple; # [mm]: HTTP-get auf URL, um LinOTP-Antwort in JSON-Format
    ↪  zu erhalten
211 use LWP::UserAgent;
212 my $ua = LWP::UserAgent->new; # [mm]: Fuer Demo noetig, da hiermit
    ↪  Ueberpruefung von HTTPS-Zertifikaten ausschaltbar und LinOTP ein self
    ↪  signed Cert. verwendet.
213 $ua->ssl_opts(
214 SSL_verify_mode => 0,#IO::Socket::SSL::SSL_VERIFY_NONE, # [mm]: ACHTUNG!
    ↪  HTTPS-Zertifikate ignorieren! -> self-signed fuer Demo!
215 verify_hostname => 0
216 );

                                        [...]
    //Bisherige Überprüfung von Kennung und Passwort...
3032 #£fehler = simpwcr_check (£kennung, £pwd, £plafoliste); #[mm] fuer OTP-Test
    ↪  weiter unten aufgerufen.
3033
3034 //...nach Extraktion des OTPs aufzurufen
3035 #------------------------------------------------------- [mm]
3036 #[mm]: Validierung des OTPs gegen die LinOTP-Instanz:
3037 # OTP wird mit dem Passwort konkateniert (OTP hinten angehaengt)
3038
3039 my $pwd_userpwd = $pwd;
3040 my $pwd_otp = "";
3041
3042 if (length($pwd)>8) { # OTP vom Passwort abtrennen
3043 $pwd_userpwd = substr($pwd,0,length($pwd)-$OTP_LENGTH);
3044 $pwd_otp = substr($pwd,length($pwd)-$OTP_LENGTH,$OTP_LENGTH);
3045 }
3046
3047 # Nutzername-Passwort-Kombination pruefen
3048 $fehler = simpwcr_check ($kennung, $pwd_userpwd, $plafoliste);
3049
3050
3051 ## OTP validieren
3052 my $linotp_url = 'https://simsrv17.sim.lrz.de';
3053 my $linotp_request = $linotp_url . '/validate/check?user=' . $kennung .
    ↪  '&pass=' . $pwd_otp;
3054 my $linotp_response_raw = $ua->get ( $linotp_request )->content;
3055
3056
3057 my $linotp_response = decode_json($linotp_response_raw);
3058
```

```
3059 my $linotp_response_result_status = $linotp_response->{result}->{status};
3060 my $linotp_response_result_value  = $linotp_response->{result}->{value};
3061 my $linotp_result = $linotp_response_result_status &&
     ↪  $linotp_response_result_value;
3062
3063 if (!$linotp_result) {
3064 print "Das OTP konnte nicht verifiziert werden.\n<br>";
3065 $fehler = "Passwort falsch";
3066 }
3067
3068
3069 #---------------------------------------------------- [mm]
```

[...]

Abbildung B.1.: Erweiterung der Datei `PERLLIB/Valid.pm` zur Validierung beim *idportal*-Login eingegebener OTPs gegen eine *LinOTP*-Test-Instanz

B.2. LinOTP-API-Call und -Result zur Validierung von OTPs

`https://[linotpinstanz]/validate/check?user=[NUTZERKENNUNG]&pass=[TOKEN]`

```
1 {
2         "version": "LinOTP 2.10.1.1",
3         "jsonrpc": "2.0802",
4         "result": {
5                 "status": true,
6                 "value": true
7         },
8         "id": 0
9 }
```

Abbildung B.2.: Rückgabewert eines *LinOTP*-API-Calls zur Validierung eines übergebenen OTPs im JSON-Format. `result.value` enthält das – hier positive – Ergebnis der Validierung.

C. Gesamtfassung des Anforderungskataloges

Anforderung	*Gewichtung*
1 Rahmenanforderungen *Rahmenbedingungen für das gesamte 2FA-System des LRZs*	2
1-1 Universelle Lösung *Die gefundene 2FA-Lösung ist für alle Szenarien einsetzbar und erfordert nicht mehrere Teilbereichslösungen.*	3
1-2 Einbezug bestehender Komponenten als zweiten Faktor *Bereits vorhandene Komponenten können zur Authentifizierung oder als Träger des zweiten Faktors einbezogen werden und so Kostenersparnis oder höheren Nutzungskomfort erzielen. (z.B. Smartphones, Email, RFID-Chips, ...)*	1
1-3 Verbreitungsgrad und Akzeptanz *Die 2FA-Lösung ist auf dem Markt akzeptiert und weit verbreitet. Das System gilt als ausgereift und hat sich im Unternehmenseinsatz bewährt.*	3
1-4 inhouse hosting *Das 2FA-System kann vollständig inhouse/on-premise gehostet werden ohne auf einen externen Dienstleister angewiesen zu sein. Am Authentifizierungsvorgang sind nur Anwenderterminal und Dienste des LRZs beteiligt.*	4
2 Einrichtungsaufwand *Zeit- und Ressourcenaufwand der Implementierung und Einführung der 2FA-Lösung*	3
2-1 Integration in bestehende AAA-Systeme *Die Authentifizierung des zweiten Faktors lässt sich in bestehende AAA-Systeme integrieren.*	4
2-2 Zusätzliche IT-Infrastruktur *Es ist möglichst wenig zusätzliche IT-Infrastruktur (Hard- und Software) im AAA-Backend nötig.*	2
2-3 Aufwand der Einführung *Der Zeit- und Ressourcenaufwand der Inbetriebnahme des 2FA-Systems (Rollout) ist gering.*	3
2-3-1 Import bestehender Anwenderkennungen *Alle bestehenden Anwenderkennungen lassen sich initial mit geringem Aufwand in der 2FA-Verwaltungssoftware registrieren.*	4
2-3-2 Hardwareanpassung der Anwenderterminals *An den Terminals der mit 2FA auszustattenden Anwendungen sind möglichst keine Hardwareanpassungen nötig (neue Schnittstellen, Lesegeräte etc.).*	3

	2-3-3 Anpassung der Anwendungen *Es sind möglichst wenig bzw. unkomplizierte Anpassungen der mit 2FA auszurüstenden Anwendungen nötig (neue Eingabefelder, APIs etc.).*	2
	2-3-4 Verteilen der zweiten Faktoren an die Anwender *Das Verteilen der zweiten Faktoren an die Anwender ist mit möglichst geringem Aufwand verbunden.*	3
2-4 Einrichtungsaufwand für den Anwender *Der Nutzer kann seinen zweiten Faktor mit möglichst geringem Initialaufwand in Betrieb nehmen.*		2
	2-4-1 Clientseitige Einrichtung auch durch technisch nicht-versierte *Die clientseitige Einrichtung der 2FA erfordert keine gesteigerten technischen Fachkenntnisse.*	3
	2-4-2 Fremdrechner *Bei der Nutzung eines nicht vorkonfigurierten Fremdrechners ist kein großer Einrichtungsaufwand für die Nutzung von 2FA gegenüber LRZ-Diensten und insb. kein Administratorzugang erforderlich. (Ausnahme: Desktoplogin am Fremdrechner)*	3
2-5 Technische Dokumentation *Alle Komponenten des 2FA-Systems sind ausführlich dokumentiert.*		3
2-6 Anleitungsmaterial für Endanwender *Der Hersteller liefert (Vorlagen für) an Endanwender gerichtete Anleitungen zur Interaktion mit dem 2FA-System (u.a. 2F-Login, Self-Service-Funktionen, Vorgehensweisen bei Diebstahl/Verlust). Für das LRZ entsteht kaum redaktioneller Aufwand zur Erstellung solcher.*		1
3 Verwaltung, Wartung, Erweiterbarkeit *Administrative Tätigkeiten wie bspw. Hinzufügen neuer Nutzer oder Vergabe von temporären Zugängen (Szenario 5)*		4
3-1 Erweiterbarkeit *Neue Dienste und Nutzeraccounts können aufwandsarm mit 2FA ausgestattet werden.*		3
	3-1-1 Neue Dienste *Das 2FA-System lässt sich nachträglich mit geringem Aufwand auf weitere Anwendungen ausweiten. (Der Aufwand ist deutlich geringer als für die Einrichtung des 2FA-Systems.)*	2
	3-1-2 Neue Anwender *Die Anzahl der 2FA-Zugänge ist praktisch nur durch das Lizenzpaket limitiert.*	4
	3-1-3 Weitere 2FA-Technologien *Neben der hauptsächlich gewählten, lassen sich nach der Einführung auch weitere Arten von zweiten Faktoren in das 2FA-System integrieren.*	1
	3-1-4 Adaption an den „Stand der Technik“ *Das 2FA-System ist offen für verbesserte Versionen eingesetzter 2FA-Technologien (z.B. höhere Schlüssellängen o.ä.).*	3
3-2 Wartung *Der Aufwand zur Instandhaltung des 2FA-Systems ist gering.*		3
	3-2-1 Supportzusage *Technologie und Support der Anbieterfirma werden nicht in absehbarer Zeit eingestellt. Die Software wird aktiv gewartet und weiterentwickelt.*	3

	3-2-2 Kontinuierlicher Aufwand des zweiten Faktors *Die kontinuierliche Pflege jedes zweiten Faktors erfordert wenig Aufwand (Batteriewechsel, Aktualisierung von Keys etc.).*	3
	3-2-3 Kontinuierlicher Aufwand der 2FA-Dienste *Der kontinuierliche Aufwand für Schutz und Pflege des 2FA-Servers sowie der ausgestatteten Anwendungen ist gering.*	2
	3-2-4 Backup *Das 2FA-System lässt sich mit allen relevanten Komponenten sichern und wiederherstellen.*	4
3-3 Verwaltung *Typische Verwaltungsaufgaben erfordern wenig Aufwand (Arbeitszeit).*		4
	3-3-1 Funktionsumfang der Verwaltungssoftware *Die Software unterstützt die Verwaltung des 2FA-Systems maßgeblich.*	3
	3-3-1-1 Automatisierbarkeit von Routinearbeiten *Die Software unterstützt u.a. automatische Vergabe und Rollout von zweiten Faktoren.*	4
	3-3-1-2 Unterstützung verschiedener 2FA-Technologien *Die Software kann verschiedene Arten von zweiten Faktoren über ein einheitliches Interface verwalten.*	1
	3-3-1-3 Self-Service *Die Software bietet umfangreiche Funktionen zur Verwaltung der eigenen 2FA durch den Anwender.*	3
	3-3-1-4 Helpdesk-Rolle *Neben Nutzer- und Administrator-Rolle verfügt die Verwaltungssoftware auch über eine Helpdesk-Rolle.*	1
	3-3-1-5 Verwaltung von Hardwaretoken *Die Software unterstützt bei der Verwaltung von Hardwaretoken (Inventarisierung, Verteilung).*	1
	3-3-2 Auswirkung auf bisherigen Prozess zur Kennungsverwaltung *Im bisherigen Prozess zur Vergabe, Verwaltung und Überprüfung von Berechtigungen und Zugangsdaten sind keine großen Änderungen nötig.*	3
	3-3-3 Änderung des zweiten Faktors *Art oder Key des zweiten Faktors eines Accounts ist nachträglich änderbar oder neu ausstellbar (durch Nutzer oder Admin).*	3
	3-3-4 Wiederherstellung nach Diebstahl/Verlust *Für Ersatz nach Diebstahl oder dauerhaftem Verlust des zweiten Faktors ergibt sich für den Admin oder Helpdesk höchstens geringer Aufwand.*	2
	3-3-5 Sperren von zweiten Faktoren *Nach Diebstahl oder dauerhaftem Verlust des zweiten Faktors kann dieser mit geringem Aufwand gesperrt werden.*	3
	3-3-6 Anwender-Self-Service nach Verlust des zweiten Faktors (Fallback) *Ein Anwender kann seinen zweiten Faktor nach Vergessen oder dauerhaftem Verlust selbst sperren oder wiederherstellen (Recovery-Codes o.ä.).*	2
	3-3-7 Mehrere zweite Faktoren pro Anwender *Einem Anwenderaccount können mehrere zweite Faktoren zugeordnet werden. (z.B. als Backup bei Verlust)*	1

	3-3-8 Geltungsbereich der 2FA *Die 2FA ist pro Dienst und Anwender(-gruppe) durch den Admin (und ggf. den Nutzer) einzeln aktivierbar.*	4
	3-3-9 Ausweitung auf synchronisierte Benutzerverwaltungen *Die Einführung der 2FA darf sich nicht zwangsweise auch auf die zentralen Benutzerverwaltungen der Universitäten und Institute erstrecken.*	4
	3-3-10 Portierungsmöglichkeit *Ein bestehender zweiter Faktor kann einem anderen Nutzer zugeordnet werden. (teure Hardwaretoken)*	1
	3-3-11 Temporäre Zugänge *Zweite Faktoren lassen sich auch temporär (tagesweise) oder ersatzweise ausstellen, falls der eigene Token bspw. zu Hause vergessen wurde.*	3
	3-3-11-1 Gastzugänge *Zeitlich beschränkte neue Zugänge mit 2FA sind ausstellbar.*	3
	3-3-11-2 Tagesersatz für zweiten Faktor *Ein temporärer Ersatz für den zweiten Faktor, falls dieser z.B. von einem Dienstreisenden zu Hause vergessen wurde, lässt sich unter Wahrung der Authentizität zuteilen.*	2
	3-3-11-3 Einrichtungsaufwand für temporäre Zugänge *Der Einrichtungsaufwand für temporäre bzw. 2FA-Gastzugänge ist gering.*	3
	3-3-12 Einfluss auf Verwaltung des ersten Faktors *Der zweite Faktor bringt kaum zusätzlichen Aufwand zur Verwaltung des ersten Faktors mit sich.*	2
	3-3-13 Rücksetzung des ersten Faktors *Eine Rücksetzung des ersten Faktors zieht nicht notwendigerweise eine Rücksetzung des zweiten Faktors mit sich. (z.B. teure Hardwaretoken)*	2
4 Sicherheitsniveau *Anforderungen aus Perspektive der IT-Sicherheit*		3
	4-1 Nichtabstreitbarkeit *Die Kombination aus erstem und zweitem Faktor lässt zweifelsfrei auf Authentizität des Anwenders schließen.*	2
	4-2 Angemessenes Sicherheitsniveau *Das 2FA-System verfügt über ein ausreichend hohes Sicherheitsniveau für die kommenden fünf Jahre.*	3
	4-2-1 Stand der Technik *Der eingesetzte zweite Faktor wird voraussichtlich für die nächsten fünf Jahre noch als „Stand der Technik“ gelten.*	3
	4-2-2 Zweiter Faktor nicht ableitbar *Der zweite Faktor ist weder aus Kenntnis des ersten, noch durch Wissen über vorangegangene Verwendungen ableitbar.*	3
	4-2-3 Angemessene Verschlüsselungsverfahren *Im Zuge der 2FA kommen jederzeit angemessene, dem „Stand der Technik“ entsprechende Verschlüsselungsverfahren für Übertragung und Speicherung zum Einsatz.*	3
	4-2-4 Kontextbasierte MFA *Das System verfügt über die Möglichkeit Kontextinformationen als weitere Faktoren in den Authentifizierungsvorgang mit einzubeziehen.*	1

	4-3 Schutz gegen Spoofing *Der Mindestaufwand für erfolgreiches Spoofing ist hoch.*	3
	4-3-1 Fund des zweiten Faktors nicht ausreichend *Fund oder Diebstahl eines zweiten Faktors allein eröffnet kein wesentliches Risiko für erfolgreiches Spoofing (Smartcard mit Fingerabdrucksensor, OTP-App mit PIN, FIDO etc.).*	3
	4-3-2 Replay *Das 2FA-System verhindert Replay-Angriffe nach Phishing oder Sniffing auf dem Übertragungskanal.*	3
	4-4 Übertragung des zweiten Faktors *Der zweite Faktor wird während des Authentifizierungsvorgangs in angemessen sicherer Weise übertragen.*	3
	4-4-1 Out-of-Band *Der zweite Faktor wird „out-of-Band“ übertragen.*	1
	4-4-2 Anzahl der Übertragungen *Der zweite Faktor wird höchstens einmal übertragen (Client nach Server)*	1
	4-4-3 Kein Klartext *Der zweite Faktor wird nie im Klartext übertragen.*	4
	4-5 Injektivität *Unterschiedlichen Benutzeraccounts werden unterschiedliche Ausprägungen des zweiten Faktors zugeordnet (ein Nutzer hat nicht den gleichen 2F für alle seine Accounts)*	2
	4-6 Verfügbarkeit *Eine Zwei-Faktor-Authentifizierung ist zu jedem Zeitpunkt möglich.*	4
	4-6-1 Ausfallsicherheit des 2FA-Systems *Die Dienste zur Authentifizierung des zweiten Faktors lassen sich redundant auslegen und Ausfallsicherheit so gewährleisten.*	4
	4-6-2 Intervall der Authentifizierungsvorgänge *Die nötige Wartezeit zwischen zwei Authentifizierungsvorgängen ist akzeptabel gering.*	1
	4-6-3 Physische Robustheit *Der zweite Faktor ist robust gegen physische Beschädigungen.*	2
	4-7 Manipulation des zweiten Faktors *Der zweite Faktor bzw. dessen Träger sowie entsprechende Lesegeräte sind gegen unbemerkte Manipulation geschützt.*	3
	4-8 Keine Speicherung des zweiten Faktors *Der zweite Faktor kann von Anwendung oder Terminal nicht in einer zur erfolgreichen späteren Analyse geeigneten Art gespeichert werden.*	2
5 Kosten(-effizienz) *Finanzielle Aspekte für das LRZ sowie einzelne Anwender*		2
	5-1 Für das LRZ *Kosten für das LRZ für Einführung und Betreib einer 2FA*	3
	5-1-1 Pro Mitarbeiter *Kosten des zweiten Faktors pro Mitarbeiter.*	3
	5-1-1-1 Anschaffung pro Mitarbeiter *Anschaffungskosten des zweiten Faktors pro Mitarbeiter*	1
	5-1-1-2 Kontinuierlich pro Mitarbeiter *Kontinuierliche Kosten des zweiten Faktors pro Mitarbeiter*	3

	5-1-1-3 Wiederherstellung nach Verlust pro Mitarbeiter *Kosten für die Wiederherstellung nach Diebstahl/Verlust des zweiten Faktors pro Mitarbeiter*	1
	5-1-2 2FA-Infrastuktur *Kosten für Anschaffung und Unterhalt der nötigen Hard- und Software des 2FA-Systems (inkl. Lizenzen)*	2
	5-1-3 Verwaltung von 2FA-Authentifizierungsinformationen *Kosten für Erstellen, Verknüpfen oder Zurücksetzen des zweiten Faktors*	3
	5-1-4 Späteres Hinzufügen neuer 2FA-Nutzer *Kosten für das Ausstatten später hinzukommender weiterer Anwenderaccounts mit 2FA*	1
	5-1-5 Pro auszustattender Anwendung *Kosten pro Anwendung(sinstanz), deren Authentifizierung mittels 2FA gestärkt werden soll*	2
	5-2 Für Nutzer von LRZ-Diensten *Kosten für Nicht-Mitarbeiter, die 2FA für LRZ-Dienste nutzen wollen*	2
	5-2-1 Anschaffung für den Nutzer *Anschaffungskosten des zweiten Faktors pro Nutzer*	2
	5-2-2 Kontinuierlich für den Nutzer *Kontinuierliche Kosten des zweiten Faktors pro Nutzer*	3
	5-2-3 Wiederherstellung nach Verlust für den Nutzer *Kosten für die Wiederherstellung nach Diebstahl/Verlust des zweiten Faktors pro Nutzer*	2
6 Bedienkomfort *Der Authentifizierungsvorgang aus Anwendersicht*		3
	6-1 Einarbeitungszeit für Anwender *Einarbeitungszeit und Schulungsbedarf für neue 2FA-Nutzer bzw. Gäste sind durch intuitive Bedienbarkeit gering.*	3
	6-2 Verlängerung des Authentifizierungsvorgangs *Der zusätzliche Zeitaufwand bei der Authentifizierung (für Anwender und System) ist gering.*	4
	6-3 Unkomplizierte Handhabung *Die Eingabe des zweiten Faktors im Rahmen des Authentifizierungsvorgangs ist für den Anwender wenig aufwändig.*	3
	6-4 Verschiedene Authentisierungsmethoden *Dem Nutzer stehen verschiedene Authentisierungsmethoden zur Auswahl.*	1
	6-5 Branding *Die 2FA-Lösung lässt sich in das LRZ-Design überführen.*	1
	6-6 Mehrbelastung der Anwender *Der zweite Faktor ist leicht mitzuführen und verursacht nur eine geringe Mehrbelastung der Mitarbeiter.*	4
	6-6-1 Transportierbarkeit *Der zweite Faktor ist leicht physisch mitzuführen.*	4
	6-6-2 Einheitlichkeit *Jeder Anwender soll für LRZ-Dienste möglichst gleichartige und wenige zweite Faktoren benötigen.*	3

	6-7 Private Nutzung *Der zweite Faktor lässt sich ohne Verringerung des Sicherheitsniveaus auch für private Zwecke nutzen.*	1
	6-8 Redundante zweite Faktoren *Pro Anwender lassen sich redundante zweite Faktoren anlegen (OTP-App auf mehreren Geräten, Backup-FIDO-Stick).*	1
	6-9 Design der User-Interfaces *Das Design der User-Interfaces ist übersichtlich und ansprechend gestaltet. Es entspricht modernen Gestaltungsprinzipien. (Loginmasken, Self-Service-Portal etc.)*	2
7 Anwendungsbereiche *Anforderungen aus den Einsatzszenarien 1 bis 4*		4
	7-1 An LRZ-Kennung koppelbar *Der zweite Faktor ist an die LRZ-Kennung des Anwenders koppelbar.*	4
	7-2 RADIUS-Integration *Die Authentifizierung des zweiten Faktors ist in den zentralen RADIUS-Dienst integrierbar.*	4
	7-3 LDAP-Integration *Die Authentifizierung des zweiten Faktors ist in den zentralen OpenLDAP-Dienst integrierbar.*	4
	7-4 Mobilgeräte *Der zweite Faktor ist von Mobilgeräten wie Smartphone oder Tablet aus einsetzbar.*	3
	7-4-1 Android *Die Eingabe des zweiten Faktors ist auf Android-Geräten möglich.*	4
	7-4-2 Blackberry *Die Eingabe des zweiten Faktors ist auf Blackberry-Geräten möglich.*	2
	7-4-3 iOS *Die Eingabe des zweiten Faktors ist auf iOS-Geräten möglich.*	4
	7-4-4 USB-Schnittstelle nicht erforderlich *Es gibt eine Möglichkeit den zweiten Faktor auch ohne USB-Schnittstelle eingeben zu können.*	4
	7-5 Fernwartung per SSH (Szenario 1) *Das 2FA-System erfüllt die Anforderungen aus Szenario 1: Fernwartung per SSH*	4
	7-5-1 Eingabe über Kommandozeile *Der zweite Faktor lässt sich während einer Konsolensitzung über die Kommandozeile eingeben.*	3
	7-5-2 SSH-Integration *Die Abfrage des zweiten Faktors lässt sich in den Authentifizierungsvorgang (via Username/Password) der SSH-Verbindung integrieren.*	4
	7-5-3 UNIX-Kompatibilität *Benötigte Software zur Authentifizierung des zweiten Faktors lässt sich auf UNIX-Systemen (Debian, Ubuntu, Suse) installieren.*	4
	7-5-4 Dezentrale Authentifizierung *Eine 2FA ist auch gegenüber UNIX-Servern möglich, die nicht an einen zentralen Authentifizierungsdienst wie RADIUS angeschlossen sind.*	1
	7-5-5 Windows-SSH-Clients *Der zweite Faktor lässt sich mit Windows-SSH-Clients wie bspw. PuTTY nutzen.*	3

	7-6 Telearbeit und VPN (Szenario 2) *Das 2FA-System erfüllt die Anforderungen aus Szenario 2: Telearbeit und VPN*	4
	7-6-1 VPN-Server Integration *Die Authentifizierung des zweiten Faktors lässt sich in den authentifizierenden Dienst des VPN-Servers integrieren.*	4
	7-6-2 Kompatibel mit Cisco Anyconnect Desktop-VPN-Client *Der zweite Faktor lässt sich zusammen mit der Desktop-Version des Cisco Anyconnect VPN-Clients nutzen.*	4
	7-6-3 Kompatibel mit weiteren gängigen Desktop-VPN-Clients *Der zweite Faktor lässt sich mit weiteren gängigen Desktop-VPN-Clients nutzen.*	3
	7-6-4 Kompatibel mit Cisco Anyconnect mobile VPN-Client *Der zweite Faktor lässt sich zusammen mit der mobilen Version des Cisco Anyconnect VPN-Clients nutzen.*	3
	7-6-5 Kompatibel mit weiteren gängigen mobilen VPN-Clients *Der zweite Faktor lässt sich mit mobilen Versionen weiterer gängiger VPN-Clients nutzen.*	2
	7-7 Login an Arbeitsplatzrechnern (Szenario 3) *Das 2FA-System erfüllt die Anforderungen aus Szenario 3: Login an Arbeitsplatzrechnern*	4
	7-7-1 UNIX-Desktoploginsysteme *Die Abfrage des zweiten Faktors ist in aktuelle UNIX-Desktoploginsysteme integrierbar. (Debian, Ubuntu, Suse)*	4
	7-7-2 MacOS-Desktoploginsysteme *Die Abfrage des zweiten Faktors ist in aktuelle MacOS-Desktoploginsysteme integrierbar.*	4
	7-7-3 Windows-Desktoploginsysteme *Die Abfrage des zweiten Faktors ist in aktuelle Windows-Desktoploginsysteme integrierbar.*	4
	7-7-4 MS Active Directory *Die Authentifizierung des zweiten Faktors lässt sich in das MWN-ADS (Microsoft Active Directory) integrieren.*	4
	7-7-5 Sperrbildschirm *Das 2FA-System bietet dem Administrator die Möglichkeit nach einem Sperrbildschirm die Eingabe des zweiten Faktors nicht zu fordern.*	1
	7-7-6 Offline Modus *Ein Desktoplogin mittels 2FA ist (für Ausnahmefälle) auch ohne Internet-Verbindung möglich. (z.B. Laptop im Zug)*	2
	7-8 Webapplikationen (Szenario 4) *Das 2FA-System erfüllt die Anforderungen aus Szenario 4: Webapplikationen*	4
	7-8-1 Webbrowserkompatibilität *Die Authentifizierung des zweiten Faktors kann über einen gängigen Webbrowser erfolgen.*	4
	7-8-1-1 Apple Safari *Die Eingabe des zweiten Faktors ist in mobiler und Desktop-Version von Apple Safari möglich.*	3
	7-8-1-2 Google Chrome *Die Eingabe des zweiten Faktors ist in mobiler und Desktop-Version von Google Chrome möglich.*	3

	7-8-1-3 Mozilla Firefox *Die Eingabe des zweiten Faktors ist in mobiler und Desktop-Version von Mozilla Firefox möglich.*	3
	7-8-2 Integration in Webanwendungen *Das 2FA-System stellt Schnittstellen zur Authentifizierung des zweiten Faktors in verschiedenen Programmiersprachen des Web-Kontextes und Beispielcode oder Plugins für gängige Webanwendungen bereit.*	3
	7-8-2-1 PHP *Der zweite Faktor lässt sich mittels PHP-Modul abfragen und authentifizieren.*	3
	7-8-2-2 JavaScript *Der zweite Faktor lässt sich mittels JavaScript-Modul abfragen und authentifizieren.*	3
	7-8-2-3 Python *Der zweite Faktor lässt sich mittels Python abfragen und authentifizieren.*	2
	7-8-3 Föderiertes Identitätsmanagement *Die Integration des zweiten Faktors in ein föderiertes Identitätsmanagementsystem (Shibboleth) ist möglich.*	1
	7-8-4 Lokale Authentifizierungsdienste *Auch gegenüber Webanwendungen, die nicht an den eigenständigen zentralen Authentifizierungsdienst angebunden sind, kann sich der Nutzer mit seinem zweiten Faktor authentifizieren.*	2
	7-9 Plugins für gängige Anwendungen *Plugins zur Integration der 2FA in gängige Web- und Desktopanwendungen werden bereit gestellt.*	1

Tabelle C.1.: Gesamtfassung des Anforderungskatalogs

D. Gesamtfassung der Produktbewertung

Anforderung	**Gewicht**	**Produktbewertung** *LinOTP*	*Defender*	*OpenOTP*	*SecureAuth*
0 Zwei-Faktor-Authentifizierung am LRZ	**4**	2.56	***0***	2.58	2.58
1 Rahmenanforderungen	2	2.73	2.91	2.73	3.0
1-1 Universelle Lösung	3	3	3	3	3
1-2 Einbezug bestehender Komponenten als zweiten Faktor	1	3	2	3	3
1-3 Verbreitungsgrad und Akzeptanz	3	2	3	2	3
1-4 inhouse hosting	**4**	3	3	3	3
2 Einrichtungsaufwand	3	2.5	2.32	2.43	2.38
2-1 Integration in bestehende AAA-Systeme	**4**	3	3	3	3
2-2 Zusätzliche IT-Infrastruktur	2	**???**	**???**	**???**	**???**
2-3 Aufwand der Einführung	3	2.83	2.58	2.83	2.58
2-3-1 Import bestehender Anwenderkennungen	**4**	3	3	3	3
2-3-2 Hardwareanpassung der Anwenderterminals	3	3	3	3	3
2-3-3 Anpassung der Anwendungen	2	2	2	2	2
2-3-4 Verteilen der zweiten Faktoren an die Anwender	3	3	2	3	2
2-4 Einrichtungsaufwand für den Anwender	2	3.0	3.0	3.0	3.0
2-4-1 Clientseitige Einrichtung auch durch technisch nicht-versierte	3	3	3	3	3
2-4-2 Fremdrechner	3	3	3	3	3
2-5 Technische Dokumentation	3	3	2	3	3
2-6 Anleitungsmaterial für Endanwender	1	2	3	1	1
3 Verwaltung, Wartung, Erweiterbarkeit	**4**	2.72	2.54	2.71	2.65
3-1 Erweiterbarkeit	3	3.0	2.5	3.0	3.0
3-1-1 Neue Dienste	2	3	2	3	3

	3-1-2 Neue Anwender	**4**	3	3	3	3
	3-1-3 Weitere 2FA-Technologien	1	3	3	3	3
	3-1-4 Adaption an den „Stand der Technik“	3	3	2	3	3
3-2 Wartung		3	2.5	2.5	2.5	2.5
	3-2-1 Supportzusage	3	3	3	3	3
	3-2-2 Kontinuierlicher Aufwand des zweiten Faktors	3	3	3	3	3
	3-2-3 Kontinuierlicher Aufwand der 2FA-Dienste	2	**???**	**???**	**???**	**???**
	3-2-4 Backup	**4**	3	3	3	3
3-3 Verwaltung		**4**	2.67	2.6	2.64	2.51
	3-3-1 Funktionsumfang der Verwaltungssoftware	3	3.0	2.9	2.7	2.5
	3-3-1-1 Automatisierbarkeit von Routinearbeiten	**4**	3	3	3	3
	3-3-1-2 Unterstützung verschiedener 2FA-Technologien	1	3	2	3	3
	3-3-1-3 Self-Service	3	3	3	3	2
	3-3-1-4 Helpdesk-Rolle	1	3	3	0	3
	3-3-1-5 Verwaltung von Hardwaretoken	1	3	3	3	1
	3-3-2 Auswirkung auf bisherigen Prozess zur Kennungsverwaltung	3	**???**	**???**	**???**	**???**
	3-3-3 Änderung des zweiten Faktors	3	3	3	3	3
	3-3-4 Wiederherstellung nach Diebstahl/Verlust	2	3	3	3	3
	3-3-5 Sperren von zweiten Faktoren	3	3	3	3	3
	3-3-6 Anwender-Self-Service nach Verlust des zweiten Faktors (Fallback)	2	2	1	3	3
	3-3-7 Mehrere zweite Faktoren pro Anwender	1	3	3	3	3
	3-3-8 Geltungsbereich der 2FA	**4**	3	3	3	3
	3-3-9 Ausweitung auf synchronisierte Benutzerverwaltungen	**4**	3	3	3	3
	3-3-10 Portierungsmöglichkeit	1	3	3	3	**???**
	3-3-11 Temporäre Zugänge	3	3.0	3.0	2.38	2.12
	3-3-11-1 Gastzugänge	3	3	3	3	3
	3-3-11-2 Tagesersatz für zweiten Faktor	2	3	3	2	1

	3-3-11-3 Einrichtungsaufwand für temporäre Zugänge	3	3	3	2	2
	3-3-12 Einfluss auf Verwaltung des ersten Faktors	2	3	3	3	3
	3-3-13 Rücksetzung des ersten Faktors	2	3	3	3	3
4 Sicherheitsniveau		3	2.27	2.18	2.29	2.38
4-1 Nichtabstreitbarkeit		2	2	2	2	3
4-2 Angemessenes Sicherheitsniveau		3	2.4	2.4	2.5	3.0
	4-2-1 Stand der Technik	3	2	2	2	3
	4-2-2 Zweiter Faktor nicht ableitbar	3	3	3	3	3
	4-2-3 Angemessene Verschlüsselungsverfahren	3	3	3	3	3
	4-2-4 Kontextbasierte MFA	1	0	0	1	3
4-3 Schutz gegen Spoofing		3	1.5	1.0	1.5	1.0
	4-3-1 Fund des zweiten Faktors nicht ausreichend	3	3	2	3	2
	4-3-2 Replay	3	**???**	**???**	**???**	**???**
4-4 Übertragung des zweiten Faktors		3	2.67	2.5	2.67	2.67
	4-4-1 Out-of-Band	1	2	1	2	2
	4-4-2 Anzahl der Übertragungen	1	2	2	2	2
	4-4-3 Kein Klartext	**4**	3	3	3	3
4-5 Injektivität		2	3	3	3	3
4-6 Verfügbarkeit		**4**	2.57	2.57	2.57	2.57
	4-6-1 Ausfallsicherheit des 2FA-Systems	**4**	3	3	3	3
	4-6-2 Intervall der Authentifizierungsvorgänge	1	2	2	2	2
	4-6-3 Physische Robustheit	2	2	2	2	2
4-7 Manipulation des zweiten Faktors		3	2	2	2	2
4-8 Keine Speicherung des zweiten Faktors		2	2	2	2	2
5 Kosten(-effizienz)		2	2.38	2.54	2.44	1.75
5-1 Für das LRZ		3	2.64	2.62	2.73	2.05
	5-1-1 Pro Mitarbeiter	3	2.0	2.6	2.0	1.2
		5-1-1-1 Anschaffung pro Mitarbeiter	1	2	2	2	1
		5-1-1-2 Kontinuierlich pro Mitarbeiter	3	2	3	2	1
		5-1-1-3 Wiederherstellung nach Verlust pro Mitarbeiter	1	2	2	2	2
	5-1-2 2FA-Infrastuktur	2	3	2	3	1
	5-1-3 Verwaltung von 2FA-Authentifizierungsinformationen	3	3	3	3	3

	5-1-4 Späteres Hinzufügen neuer 2FA-Nutzer	1	2	2	3	2
	5-1-5 Pro auszustattender Anwendung	2	3	3	3	3
5-2 Für Nutzer von LRZ-Diensten		2	2.0	2.43	2.0	1.29
	5-2-1 Anschaffung für den Nutzer	2	2	2	2	1
	5-2-2 Kontinuierlich für den Nutzer	3	2	3	2	1
	5-2-3 Wiederherstellung nach Verlust für den Nutzer	2	2	2	2	2
6 Bedienkomfort		3	2.35	2.5	2.45	2.75
6-1 Einarbeitungszeit für Anwender		3	2	3	2	3
6-2 Verlängerung des Authentifizierungsvorgangs		**4**	3	3	3	3
6-3 Unkomplizierte Handhabung		3	2	1	2	2
6-4 Verschiedene Authentisierungsmethoden		1	2	3	2	3
6-5 Branding		1	3	1	3	3
6-6 Mehrbelastung der Anwender		**4**	3.0	3.0	3.0	3.0
	6-6-1 Transportierbarkeit	**4**	3	3	3	3
	6-6-2 Einheitlichkeit	3	3	3	3	3
6-7 Private Nutzung		1	1	1	1	1
6-8 Redundante zweite Faktoren		1	3	3	3	3
6-9 Design der User-Interfaces		2	1	3	2	3
7 Anwendungsbereiche		**4**	2.84	***0***	2.89	2.89
7-1 An LRZ-Kennung koppelbar		**4**	3	3	3	3
7-2 RADIUS-Integration		**4**	3	3	3	3
7-3 LDAP-Integration		**4**	3	3	3	3
7-4 Mobilgeräte		3	2.71	2.86	2.71	2.71
	7-4-1 Android	**4**	3	3	3	3
	7-4-2 Blackberry	2	1	2	1	1
	7-4-3 iOS	**4**	3	3	3	3
	7-4-4 USB-Schnittstelle nicht erforderlich	**4**	3	3	3	3
7-5 Fernwartung per SSH (Szenario 1)		**4**	2.8	2.8	2.8	2.8
	7-5-1 Eingabe über Kommandozeile	3	3	3	3	3
	7-5-2 SSH-Integration	**4**	3	3	3	3
	7-5-3 UNIX-Kompatibilität	**4**	3	3	3	3
	7-5-4 Dezentrale Authentifizierung	1	**???**	**???**	**???**	**???**
	7-5-5 Windows-SSH-Clients	3	3	3	3	3
7-6 Telearbeit und VPN (Szenario 2)		**4**	3.0	2.69	3.0	3.0
	7-6-1 VPN-Server Integration	**4**	3	3	3	3
	7-6-2 Kompatibel mit Cisco Anyconnect Desktop-VPN-Client	**4**	3	3	3	3

	7-6-3 Kompatibel mit weiteren gängigen Desktop-VPN-Clients	3	3	2	3	3
	7-6-4 Kompatibel mit Cisco Anyconnect mobile VPN-Client	3	3	3	3	3
	7-6-5 Kompatibel mit weiteren gängigen mobilen VPN-Clients	2	3	2	3	3
7-7 Login an Arbeitsplatzrechnern (Szenario 3)		**4**	2.84	2.84	2.84	2.84
	7-7-1 UNIX-Desktoploginsysteme	**4**	3	3	3	3
	7-7-2 MacOS-Desktoploginsysteme	**4**	3	3	3	3
	7-7-3 Windows-Desktoploginsysteme	**4**	3	3	3	3
	7-7-4 MS Active Directory	**4**	3	3	3	3
	7-7-5 Sperrbildschirm	1	0	0	0	0
	7-7-6 Offline Modus	2	3	3	3	3
7-8 Webapplikationen (Szenario 4)		**4**	2.57	2.07	2.67	2.67
	7-8-1 Webbrowserkompatibilität	**4**	2.67	2.67	2.67	2.67
	7-8-1-1 Apple Safari	3	2	2	2	2
	7-8-1-2 Google Chrome	3	3	3	3	3
	7-8-1-3 Mozilla Firefox	3	3	3	3	3
	7-8-2 Integration in Webanwendungen	3	3.0	1.0	3.0	3.0
	7-8-2-1 PHP	3	3	1	3	3
	7-8-2-2 JavaScript	3	3	1	3	3
	7-8-2-3 Python	2	3	1	3	3
	7-8-3 Föderiertes Identitätsmanagement	1	2	3	3	3
	7-8-4 Lokale Authentifizierungsdienste	2	2	2	2	2
7-9 Plugins für gängige Anwendungen		1	2	1	3	3

Tabelle D.1.: Gesamtfassung der Produktbewertung