Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Murat Korkmaz

Grundlagen der FIDO-Authentifizierung und Vergleich mit traditionellen Authentifizierungsverfahren

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Murat Korkmaz

Grundlagen der FIDO-Authentifizierung und Vergleich mit traditionellen Authentifizierungsverfahren

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Eingereicht am: 4. April 2017

Murat Korkmaz

Thema der Arbeit
Grundlagen der FIDO-Authentifizierung und Vergleich mit traditionellen Authentifizierungsverfahren

Stichworte
FIDO-Allianz, UAF, U2F, Authentifizierung

Kurzzusammenfassung
Im Zeitalter des Internets gewinnen Authentifizierungsverfahren immer mehr an Bedeutung. Um die etablierte Authentifizierung mittels Benutzernamen und Passwort abzulösen wurde die FIDO-Allianz gegründet, welche offene und lizenzfreie Standards entwickelt. Die Standards der FIDO-Allianz ermöglichen Benutzern sich ohne einen mit dem Server geteilten Geheimnis zu authentifizieren. In dieser Arbeit wird die FIDO-Allianz sowie deren Standards vorgestellt und es werden die notwendigen technischen Grundlagen erläutert. Außerdem wird ein Standard mit bereits etablierten Authentifizierungsmethoden verglichen und anschließend wird die Entwicklung der FIDO-Allianz betrachtet.
Murat Korkmaz

Title of the paper
Basics of FIDO-Authentication and comparison with traditional authentication methods.

Keywords
FIDO-Alliance, UAF, U2F, Authentication

Abstract
In the age of the Internet, authentication methods are gaining in importance. In order to replace the established authentication by user name and password, the FIDO-Alliance was founded, which develops open and license-free standards. The standards of the FIDO-Alliance allow users to authenticate themselves without sharing a secret with the server. In this bachelor thesis, the FIDO-Alliance and their standards are presented and the necessary technical principles are explained. In addition, a standard is compared with already known authentication methods and then the growth of the FIDO-Alliance is considered.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

In einer Gesellschaft, in der unser Einkauf, die Buchung der nächsten Reise, das Verschicken sensibler Daten und vieles mehr über ein ungeschütztes und unsicheres Netzwerk läuft, spielt die Authentifizierung eine immer größere Rolle. Man besitzt mehrere Benutzernamen und Passwörter, die man sich alle merken muss, nur um beweisen zu können, dass man auch wirklich die vorgegebene Person ist. Dieses Verfahren basiert ausschließlich auf einem einzigen Nachweis der Identität, nämlich das Wissen über ein Geheimnis.

Dieses Verfahren der Authentifikation mittels Benutzername und Passwort hat sich etabliert und ist nur noch schwer aus unserem Alltag weg zu denken, ungeachtet von vielen sicherheitsrelevanten Aspekten. Die Verantwortung, eine sichere Authentifizierung zu ermöglichen, wird zum Teil dem Benutzer überlassen. Dadurch kann es dazu kommen, dass Benutzer oft einfache Passwörter wählen und diese dann für mehrere Profile verwenden. Möchten Benutzer eine hohe Sicherheit in dem Verfahren mittels Passwort erreichen, müssen Passwörter erstellt, verwaltet und geheim gehalten werden.

Um die Authentifizierung jedoch einfacher, schneller und insbesondere sicherer zu machen, wurde unter dem Namen "Fast Identity Online-Alliance", im Folgenden mit FIDO-Allianz abgekürzt, eine neue Organisation gegründet, welche neue Methoden der Authentifizierung entwickelt und standadisiert hat. In dieser neuen Methode der Authentifizierung wird ein Schlüsselpaar erzeugt, welches für die Authentifizierung benutzt werden kann. Durch die besondere Eigenschaft der Schlüssel, welche im Grundlagenkapitel genauer erläutert wird, wird ein Geheimnis, welches der Benutzer mit dem Server teilt, nicht mehr notwendig. Die Authentifizierung wird auf ein Geheimnis reduziert, welches sich ausschließlich auf dem Gerät des Benutzers befindet und einen öffentlichen Teil der dieses Geheimnis verifizieren kann, ohne das Geheimnis zu kennen. Um sich authenfizieren zu können, gibt der Benutzer beispielsweise mit einem Scan des Fingerabdrucks oder PIN diesen Schlüssel bzw. Geheimnis für die Authentifizierung frei. Der Schlüssel wird für die Authetifizierung benutzt und der Fingerabdruck muss nicht an den Server weitergegeben werden, der Fingerabdruck wird nur für die lokale Authentifizierung am Gerät verwendet. Somit ist der Server nicht in dem Besitz des Fingerabdrucks und man kann man sich sehr schnell und einfach authentifizieren. Alternativ ist es anhand eines Standards möglich, einfache Passwörter zu wählen mit einer weiteren Authentifizierungsmethode zu erweitern. Hierfür werden Geräte, wie beispielsweise USB-Sticks, für die Authentifizierung notwendig.

Insbesondere die Standardisierung der Authentifizierungsmethode bringt große Vorteile mit sich. In einer Passwort basierten Authentifizierung existieren keine Standards und die

Entwicklung kann von Entwickler zu Entwickler abweichen. Aufgrund der Standardisierung der Authentifizierungsmethode, kann solch ein Problem in der FIDO-Authentifizierung nicht auftreten. Außerdem können alle Produkte die solch eine Authentifizierung umsetzen, mit anderen Produkten kommunizieren und sind kompatibel. Das Einhalten der Spezifikation wird durch das zertifizieren der Produkte gewährleistet und ist eines der wichtigsten Aspekte der FIDO-Allianz.

1.1 Zielsetzung

Ziel dieser Arbeit ist die Darlegung des grundlegenden Vorgehens der Authentifizierung, welches die FIDO-Allianz standardisiert hat. Außerdem sollen die Vorteile der Authentifikation und Zertifikation sowie Abgrenzungen zu traditionellen Authentifizierungsverfahren ersichtlich und ein Überblick über die Bekanntheit der FIDO-Allianz gegeben werden.

1.2 Zielgruppe

Die Zielgruppe dieser Arbeit sind Personen, welche Grundkenntnisse der Authentifizierung mittels Benutzernamen und Passwort besitzen, sich im allgemeinen über das verbundene Sicherheitsrisiko bewusst und auf der Suche nach einer sicheren und einfachen Alternative sind.

1.3 Struktur

Zunächst werden im Grundlagenkapitel die Grundlagen der technischen Voraussetzungen erläutert. Diese Grundlagen sind für das Verständnis der Authentifizierung mittels der FIDO-Standards notwendig. In diesem Kapitel wird das Prinzip der öffentlichen und privaten Schlüssel, die Public Key Infrastruktur sowie der Aufbau einer sicheren Verbindung dargestellt.

Im Kapitel der FIDO-Authentifizierung wird zunächst die FIDO-Allianz kurz vorgestellt, da die FIDO-Allianz die Standards entwickelt. Nach der Vorstellung der FIDO-Allianz werden die Grundlagen der Authentifizierung mittels FIDO-Standards erläutert. Es werden wichtige Komponenten des Standards dargelegt sowie auf Bereiche der Kommunikation eingegangen.

Da ein Produkt, welches ein FIDO-Standard nutzt, zertifiziert werden kann und diese Zertifikation ein Hauptbestandteil der FIDO-Allianz ist, wird im Folgenden Kapitel der Ablauf des Zertifizierungsprozesses erläutert.

Nachdem ein Grundverständnis über die FIDO-Authentifizierung herrscht, wird ein Vergleich mit anderen etablierten Authentifizierungsverfahren gezogen.

Da ein Ziel der FIDO-Allianz die Ablösung der Authentifizierung mittels Benutzernamen

und Passwort ist, wird im Kapitel “Entwicklung der FIDO-Allianz” die Entwicklung der FIDO-Allianz betrachtet. Es wird dargelegt, wie viele Produkte zu welchem Zeitpunkt zertifiziert wurden und wann Führungsmitglieder in die FIDO-Allianz beigetreten sind.

Abschließend wird ein Fazit über die FIDO-Allianz sowie zu den Standards gezogen und die FIDO-Authentifizierung beurteilt.

1.4 Definitionen

Da folgende Begriffe grundlegend für das Verständnis dieser Arbeit sind, werden diese Begriffe im Folgenden definiert.

1.4.1 Authentifizierung

Der Begriff Authentifizierung beschreibt einen Vorgang, in dem jemand oder etwas die eigene Identität nachweist. Durch diesen Nachweis der Identität, können Zugänge oder Rechte vergeben werden. Ein Beispiel eines solchen Vorgangs ist die wissensbasierte Authentifizierung. In der wissensbasierten Authentifizierung wird der Benutzer aufgefordert seinen Benutzernamen einzugeben, wobei der Benutzername die Identität darstellt. Anschließend wird der Benutzer aufgefordert das passende Passwort einzugeben, um die Identität zu beweisen. Es existieren jedoch mehrere Möglichkeiten die Identität zu beweisen. Diese Möglichkeiten sind auch bekannt als Faktoren. Folgende Faktoren sind für eine Authentifizierung möglich:

- Das Wissen über ein Geheimnis (Beispiel: Passwort)
- Der Besitz von etwas, welches nur die zu authentifizierende Person besitzt (Beispiel: Schlüssel)
- Ein körperliches Merkmal, welches nur der zu authentifizierenden Person zuzuordnen ist (Beispiel: Fingerabdruck)

Die wissensbasierte Authentifizierung benutzt lediglich einen Faktor - das wissen über ein Geheimnis. Jedoch ist auch die Kombination von mehreren Faktoren möglich. Möchte man einen weiteren Faktor zum Beweis der Identität verwenden, bezeichnet man dies als eine Zwei-Faktor-Authentifizierung. Außerdem ist noch zu erwähnen, dass durch den Besitz von etwas einmaligem eine Angabe der Identität nicht mehr notwendig ist. Dies ist möglich, weil der Besitz auch implizit die Identität vorgibt.

1.4.2 FIDO-Authentifizierung

Es existieren zwei Standards, der UAF- und der U2F-Standard, die von der FIDO-Allianz entwickelt wurden. Diese Standards werden im Kapitel der FIDO-Authentifizierung genauer erläutert. In dem Kontext dieser Arbeit wird die FIDO-Authentifizierung als die Authentifizierung mittels der UAF- und U2F-Standards verstanden.

2 Grundlagen

Da die FIDO-Authentifizierung auf bereits vorhandenen Konzepten basiert, werden diese in dem Kapitel der Grundlagen erläutert. Zunächst wird in dem Abschnitt der asymmetrischen Kryptographie das Prinzip der öffentlichen und privaten Schlüssel erklärt. Basierend auf diesem Prinzip wird im Anschluss die Public-Key-Infrastruktur dargelegt. Anschließend wird im Abschnitt der Transport-Layer-Security der Aufbau einer sicheren und verschlüsselten Verbindung durch den TLS-Handshake erläutert.

2.1 Asymmetrisches Kryptosystem

In diesem Abschnitt soll das Prinzip des asymmetrischen Kryptosystems veranschaulicht werden. Dieses kryptographische Verfahren ist eines der Hauptbestandteile der FIDO-Authentifizierung. In diesem kryptographischen Verfahren gibt es kein gemeinsames Geheimnis, wie bei einem symmetrischen Kryptosystem. Daraus folgt, dass beide Kommunikationspartner nicht den selben geheimen Schlüssel besitzen, sondern jeweils ein Schlüsselpaar. Somit wird ein Austausch sensibler Schlüssel nicht mehr notwendig.

2.1.1 Prinzip

In dem asymmetrischen Kryptosystem wird ein Schlüsselpaar erzeugt. Dieses kann entweder vom Server, Client oder von beiden erzeugt werden. Jedes Schlüsselpaar besteht aus einem öffentlichen Teil und aus einem privaten Teil. Der öffentliche Schlüssel kann und sollte jedem zugänglich sein, somit darf der Austausch und Besitz dieses Schlüssels kein sicherheitsrelevantes Risiko darstellen. Der private Schlüssel hingegen darf nur dem Erzeuger des Schlüsselpaares bekannt sein, darf nicht ausgetauscht werden und darf auch nur im Besitz des Erzeugers sein. Eine Anforderung an das asymmetrische Kryptosystem ist, dass eine mit dem öffentliche Schlüssel aus dem Schlüsselpaar S verschlüsselte Nachricht nur mit dem privaten Schlüssel aus S entschlüsselt werden kann und eine mit dem privaten Schlüssel aus S signierten Nachricht auch nur mit dem öffentlichen Schlüssel aus S verifiziert werden kann. Wie zu sehen ist, wird in der asymmetrischen Kryptographie zwischen dem ent- und verschlüsseln von Nachrichten und dem signieren und verifizieren von Nachrichten unterschieden. [Gigovic (2014)]

2.1.2 Verschlüsselung

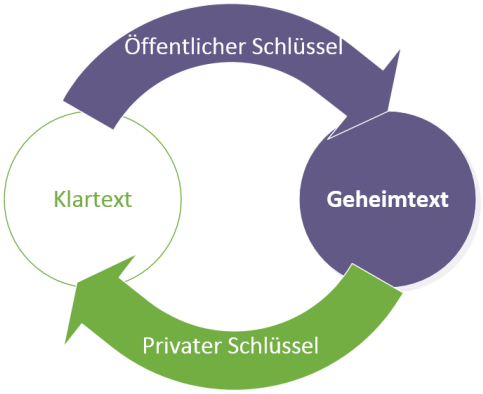

Abb. 2.1: Mit öffentlichem Schlüssel verschlüsselt und privatem Schlüssel entschlüsselt

Mit dem Besitz des öffentlichen Schlüssels des Kommunikationspartners können Nachrichten, die für den Eigentümer des privaten Schlüssels gedacht sind, wie folgt ausgetauscht werden. Nachrichten bzw. Daten die nur der Erzeuger lesen bzw. verarbeiten darf, können von dem Kommunikationspartner mit dem öffentlichen Schlüssel verschlüsselt werden. Somit kann nur der Erzeuger diese Daten lesen, weil dieser im Besitz des privaten Schlüssels ist. [PKI]

2.1.3 Signierung

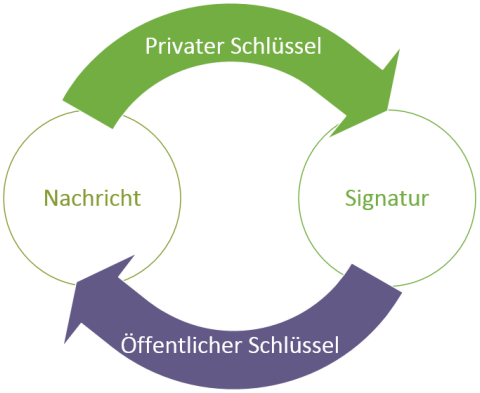

Abb. 2.2: Signieren mit privatem Schlüssel und verifizieren mit öffentlichem Schlüssel

Interessanter für die FIDO-Authentifizierung sind die digitalen Signaturen. Hierbei werden mit dem privaten Schlüssel Daten signiert und mit dem öffentlichen Schlüssel die Signatur verifiziert. Die Kombination der digitalen Signatur und Verifikation dient als Beweis für den Besitz des privaten Schlüssels und als Beweis dafür, dass der Besitzer des privaten Schlüssels der Verfasser bzw. Ersteller der signierten Nachricht ist. Um diesen Beweis für die Authentifikation nutzen zu können, muss jedoch der öffentliche Schlüssel zweifelsohne dem Erzeuger zugeordnet werden können. [PKI]

2.2 Public-Key-Infrastrukur (PKI)

Um einen öffentlichen Schlüssel einer Partei zweifelsohne zuordnen zu können, braucht man die Public-Key-Infrastruktur, kurz PKI. Durch die PKI ist eine Verwaltung und Zuordnung asymmetrischer Schlüssel möglich. Es ist möglich öffentliche Schlüssel zu verteilen und diese öffentliche Schlüssel jemandem zuzuordnen. Die Verbindung der Person und dem öffentlichem Schlüssel wird von einer Zertifizierungsstelle übernommen und in Form von Zertifikaten realisiert. Durch das agieren von verschiedenen Rollen, wird auch eine Validierung der Person sowie der Gültigkeit gewährleistet. Dies ist jedoch nur möglich, wenn einer Rolle in dieser Infrastruktur vollkommen vertraut wird. Ziel einer PKI ist, dass in einem unsicheren Netzwerk eine sichere Datenübertragung ermöglicht werden soll. Beispielsweise können so Geldtransaktionen durchgeführt werden.

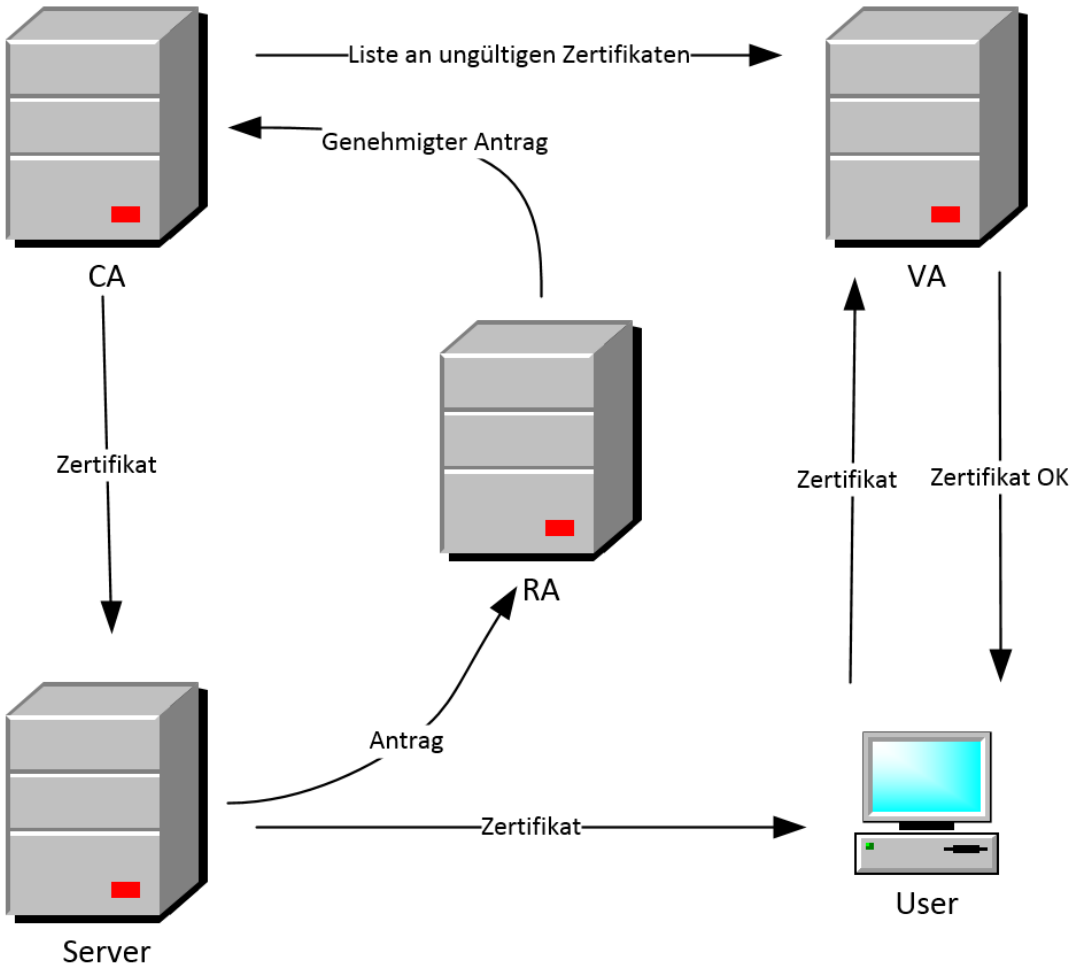

Abbildung 2.3: Public Key Infrastruktur

Certificate Authorities (CA)
Die CA ist eine Zertifizierungsstelle die Zertifikate ausstellt. Der öffentliche Schlüssel der Zertifizierungsstelle muss jedem zugänglich sein. Der Zertifizierungsstelle wird vollkommen vertraut und durch das ausstellen eines gültigen Zertifikates wird dieses Vertrauen dem Zertifikatsinhaber übertragen. Solche Zertifizierungsstellen werden auch als Root-CA bezeichnet. Somit ist die Root-CA das Fundament der Public Key Infrastruktur. Wird die Root-CA kompromittiert, ist die "Vertrauensweitergabe" nicht mehr möglich und eine Identifikation kann somit nicht mehr erfolgen. Wird der Root-CA jedoch noch vertraut, wird das Zertifikat durch die Root-CA mit dem privaten Schlüssel signiert und gilt fortan als Beweis für die Vertrauenswürdigkeit des Zertifikatsinhabers sowie als Beweis der Identität. Der Client im obigen Beispiel kann also das Zertifikat vom Server, von der CA oder von einer anderen Stelle erhalten. Wichtig hierbei ist nur, dass der Client in der Lage ist an den öffentlichen Schlüssel der Zertifizierungsstelle zu gelangen, um so das Zertifikat verifizieren zu können. [PKI]

Registration Authorities (RA)
Die RA ist eine Registrierungsstelle, in der ein Zertifikat beantragt werden kann. Oft kommuniziert die Zertifizierungsstelle nicht direkt mit einem Antragsteller. Es wird oft die Registrierungsstelle als eine Ebene zwischen diesen Partei für die Kommunikation verwendet. Ein Server kann somit einen Antrag an die Registrierungsstelle stellen, die Registrierungsstelle verifiziert die Identität des Servers und anschließend kann die Zertifizierungsstelle ein Zertifikat ausstellen. [PKI]

Validation Authorities (VA)
Die VA ist eine Validierungsstelle und wie die RA optional. Da die Zertifizierungsstelle in der Lage ist Zertifikate zu widerrufen, kann eine Liste an ungültigen Zertifikaten an die Validierungsstelle weitergeleitet werden. Die Validierungsstelle kann für die Überprüfung von Zertifikaten genutzt werden. [PKI]

Zertifikat
Das Zertifikat wird von der CA ausgestellt und beinhaltet Informationen wie beispielsweise den öffentlichen Schlüssel des Zertifikatsinhabers, den Namen des Zertifikatsinhaber und die Gültigkeit des Zertifikates. Ein derzeitiges Standard für die Public Key Infrastruktur zum Erstellen von Zertifikaten ist das x.509.

x.509
Version
Seriennummer
Algorithmen-ID
Aussteller
Gültigkeit
von
bis
Zertifikatinhaber
Zertifikatinhaber-Schlüsselinformationen
Public-Key-Algorithmus
Public Key Zertifikatinhabers
Eindeutige ID des Ausstellers (Optional)
Eindeutige ID des Inhabers (Optional)
Erweiterungen (Optional)

Abbildung 2.4: X.509 Standard

2.2.1 Vertrauenskette

Wie bereits erwähnt ist es notwendig der Zertifizierungsstelle vollkommen zu vertrauen. Jedoch wird nur einigen Zertifizierungsstellen vertraut. Diese Zertifizierungsstellen werden als Root-CA bezeichnet. Andere Zertifizierungsstellen (CA), denen nicht ohne weiteres vertraut werden kann, erhalten Zertifikate von der Root-CA und gelten durch diese Zertifikate als Vertrauenswürdig. Weitere Zertifizierungsstellen können Zertifikate von der CA erhalten. Alle Zertifikate zwischen dem Root-Zertifikat und dem vom Benutzer erhaltenen Zertifikat, werden als intermediate certificate bezeichnet. Ist die Root-CA nicht bekannt oder unvertrauenswürdig, ist der Aufbau einer Vertrauenskette nicht möglich. Somit wird allen Zertifikatsinhabern in der Kette nicht vertraut. Wird der Root-CA vertraut, kann der gesamten Kette vertraut werden. Folgende Abbildung soll solch eine Kette veranschaulichen. [PKI-Cert-Chain]

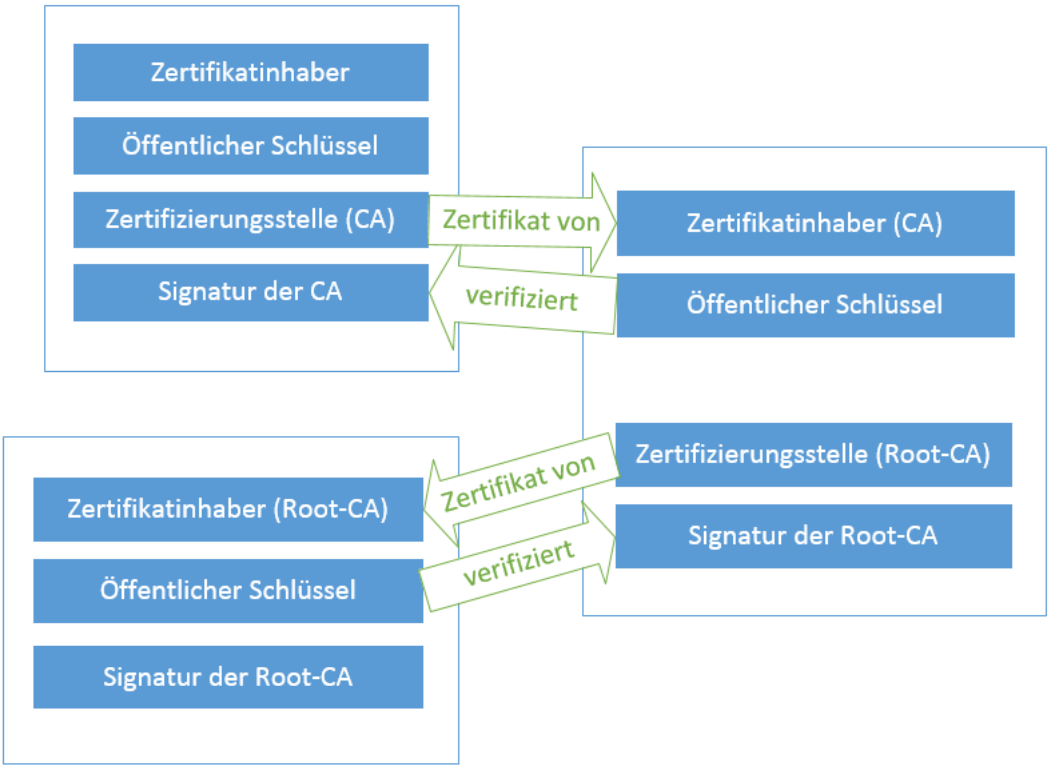

Abbildung 2.5: PKI-Vertrauenskette

2.3 Transport-Layer-Security

Für die FIDO-Authentifizierung ist neben der asymmetrischen Kryptographie und der PKI, das Transport-Layer-Security (kurz TLS) von großer Bedeutung. Die FIDO-Authentifizierung benötigt eine sichere Verbindung zwischen zwei Endpunkten um für die Authentifizierung relevante Daten senden zu können ohne das eine unbefugte dritte Person diese Daten lesen oder manipulieren kann. Diese sichere Verbindung kann durch das TLS-Protokoll aufgebaut werden. Im Zusammenhang mit einer sicheren Verbindung werden folgende Begriffe benutzt: Vertraulichkeit, Integrität und Authentizität. Der folgende Abschnitt über das Transport-Layer-Security basiert auf dem RFC 5246. [RFC5246 (2008)]

Vertraulichkeit

Unter dem Begriff der Vertraulichkeit versteht man in diesem Zusammenhang das vertrauen darauf, dass Lauschangriffe auf versendete Nachrichten nicht stattfinden können.

- **Beispiel**: Wenn ein Benutzer (sensible) Informationen an den Server übermittelt, darf keine dritte Personen diese sensiblen Informationen lesen können.

Integrität

Die Integrität steht für die Gewährleistung unveränderter Daten.

- **Beispiel**: Sendet der Sender Daten müssen diese Daten unverändert, also genau wie bei der Versendung, bei dem Empfänger ankommen.

Authentizität

Unter der Authentizität versteht man die Echtheit eines Kommunikationspartners. Im TLS-Protokoll beweist oft nur der Server seine Echtheit, jedoch besteht auch die Möglichkeit, dass beide Kommunikationspartner ihre Echtheit beweisen.

- **Beispiel**: Möchte der Benutzer eine Webseite aufrufen, so muss Webseite beweisen, dass die Webseite auch die vorgegebene Webseite ist.

Diese drei Anforderungen werden an das TLS-Protokoll gestellt. Um nachvollziehen zu können wie das TLS-Protokoll diese drei Anforderungen erfüllt, ist nachfolgend ein vereinfachter Ablauf über den Aufbau einer sicheren Verbindung mittels TLS-Handshake dargestellt.

2.3.1 TLS-Handshake

Das folgende Diagramm zeigt den Aufbau der sicheren Verbindung zwischen zwei Endpunkten und nennt sich TLS-Handshake. In diesem Beispiel wird nur die Echtheit bzw. die Authentizität des Servers bewiesen.

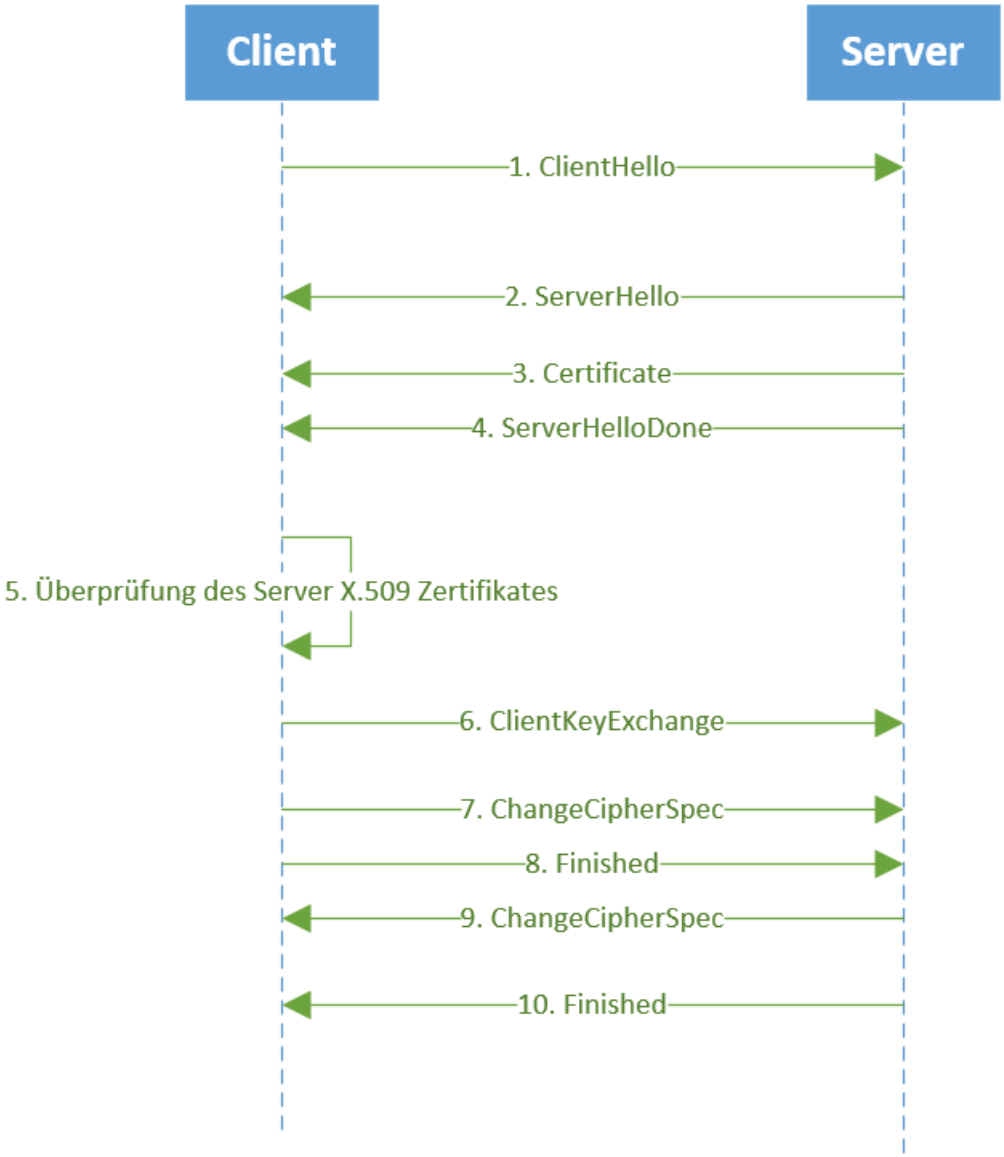

Abbildung 2.6: TLS-Handshake Sequenzdiagramm

1. ClientHello und 2. ServerHello Nachrichten
Der Client sendet eine ClientHello Nachricht worauf der Server mit einer ServerHello Nachricht antworten muss. Sollte dies nicht geschehen, erhält der Client einen FatalError und der Verbindungsaufbau schlägt fehl. Diese beiden Nachrichten werden benötigt um sich für eine Protokoll-Version, Session-ID (Ein Identifikator welcher für die Kommunikation benutzt wird), Cipher-Suite (Kryptographische Verfahren für den Schlüsselaustausch, Authentifizierung sowie für die Verschlüsselung) und Komprimierungsalgorithmus zu einigen. Außerdem werden zwei zufällige Werte generiert und ausgetauscht. Somit besitzt der Client einen 32-byte langen zufälligen Wert, welcher dem Server bekannt ist und der Server besitzt einen 32-byte langen zufälligen Wert, welcher dem Client bekannt ist. Diese zufälligen Werte können später für die Generierung eines Master-Secrets, also einen symmetrischer Schlüssel (ein Geheimnis, welcher beiden Kommunikationspartnern bekannt ist), verwendet werden.

3. Certificate
Diese Nachricht enthält das Server Zertifikat nach dem x.509 Standard und folgt immer der ServerHello Nachricht.

4. ServerHelloDone
Nach dem senden des Zertifikates sendet der Server die ServerHelloDone Nachricht und zeigt die Beendigung der ServerHello Nachricht an. Nach dieser Nachricht wartet der Server auf eine Nachricht vom Client.

5. Überprüfung des Server Zertifikates
In diesem Schritt überprüft der Client das Zertifikat. Falls das X.509 Zertifikat von einer Root-CA ausgestellt wurde, kann der Client die Echtheit des Zertifikates mit dem öffentlichen Schlüssel der Root-CA prüfen. Wurde das Zertifikat von einer CA ausgestellt, muss anhand der Vertrauenskette das Zertifikat überprüft werden. Solange also die Echtheit des Zertifikates mit dem öffentlichen Schlüssel der Root-CA überprüft werden kann, ist es nicht relevant woher der Client das Zertifikat bekommt, er muss lediglich der Root-CA vertrauen und in Besitz des öffentlichen Schlüssel der Root-CA gelangen können. Der Client kann nun mit der Überprüfung der Inhalte, wie beispielsweise der Gültigkeit, beginnen.

6. ClientKeyExchange
War die Überprüfung des Zertifikates erfolgreich, so kann der Client beginnen den öffentlichen Schlüssel des Server zu benutzen. Die ClientKeyExchange Nachricht folgt direkt der ServerHelloDone Nachricht. Aufgrund der vorherigen Übereinstimmung über das kryptographische Verfahren für den Schlüsselaustausch, wird jetzt ein symmetrischer Schlüssel ausgetauscht. Beispielsweise kann der Client einen symmetrischen Schlüssel generieren und, mit dem öffentlichen Schlüssel des Server verschlüsselt, an den Server übertragen. Es sind jedoch auch andere Verfahren möglich. wie beispielsweise die Diffie-Hellman Schlüsselvereinbarung, indem die ausgetauschten zufälligen Werte für die Schlüsselgenerierung benutzt werden.

7. ChangeCipherSpec und **9. ChangeCipherSpec**
Diese Nachrichten geben dem jeweiligen Kommunikationspartner an, dass ab sofort die ausgehandelten Schlüssel und Algorithmen für die Kommunikation verwendet werden.

8. Finished und **10. Finished**
Die Finished Nachrichten geben an, dass der Schlüsselaustausch sowie die Identifikation des Servers erfolgreich war. Diese Nachrichten folgen direkt den jeweiligen ChangeCipherSpec Nachrichten. Diese Nachrichten sind die ersten durch die ausgehandelten Algorithmen und Schlüssel geschützten Nachrichten. Wichtig hierbei ist, dass nicht mehr mit der asymmetrischen Verschlüsselung Daten verschlüsselt werden, sondern wird nun der symmetrische Schlüssel verwendet. Dieser symmetrische Schlüssel ist an dieser Stelle beiden Kommunikationspartnern bekannt.

3 FIDO-Authentifizierung

Die FIDO-Allianz wurde von mehreren Unternehmen im Zusammenschluss gegründet, um offene und lizenzfreie Standards für die Authentifizierung im Internet zu entwickeln. Sie ist eine nichtkommerzielle Organisation, die im Februar 2013 mit dem Ziel gegründet wurde, die weit verbreitete Authentifizierung mittels Benutzernamen und Passwort abzulösen. Dies soll durch die Standards U2F und UAF realisiert werden. [History]

Universal Authentication Framework (UAF)
Das UAF-Protokoll erlaubt es einem Benutzer sich ohne das verwenden eines Passwortes zu authentifizieren. Das Protokoll soll das etablierte Authentifizierungsverfahren ablösen, indem der Benutzer dazu aufgefordert wird sich z.B. per Fingerabdruck an einem Gerät zu authentifizieren. Das besondere an diesem Verfahren ist, dass der Fingerabdruck nicht an den Server weitergeleitet wird und nur zu einer lokalen Authentifizierung beiträgt. Nach der lokalen Authentifizierung wird von dem Gerät die asymmetrische Kryptographie verwendet, um den Benutzer am Server authentifizieren zu können. [Approach-Vision]

Universal Second Factor (U2F)
Das U2F-Protokoll kann dazu verwendet werden, ein bereits existierendes Verfahren zur Benutzerauthentifizierung zu verstärken, indem es das Verfahren um eine weitere Authentifizierungsmethode erweitert. Diese Erweiterung kann z.B. die Benutzung eines USB-Sticks sein, welches Informationen beinhaltet die erst die Authentifizierung ermöglichen. Wie bei dem UAF-Protokoll wird auch bei dem U2F-Protokoll ein kryptographisches Verfahren verwendet. Durch das Anwenden des U2F-Protokolls ist es für Benutzer möglich einfache Passwörter zu verwenden, ohne das Sicherheitsniveau zu senken. [Approach-Vision]

Durch den Einsatz der Standards wird die Abhängigkeit an Passwörter reduziert. Das Merken vieler verschiedener und langer Passwörter wird damit nicht mehr notwendig und eine schnellere Authentifizierung wird ermöglicht. Man versucht diese Standards weltweit zu verbreiten um einheitliche, sichere und erweiterbare Standards in der Industrie schneller und kostengünstiger einsetzen zu können.

Im Folgenden wird die FIDO-Allianz sowie das UAF- und U2F-Standard vorgestellt.

3.1 FIDO-Allianz

Die Gründer der FIDO-Allianz, somit auch die ersten Mitglieder, sind PayPal, Lenovo, NokNok Labs, Synaptics, Infineon und Angnitio. Heute sind über 250 Unternehmen, wie beispielsweise Google, Intel, Microsoft oder Samsung, in der FIDO-Allianz vertreten [History]. Die Mitgliedschaft ermöglicht es Unternehmen an Standards mit zu wirken und somit ein tieferes Verständnis über die FIDO-Produkte zu erhalten. Weitere Vorteile einer Mitgliedschaft werden von der FIDO-Allianz in folgende Kategorien unterteilt [Membership]:

- Leadership Benefits (Möglichkeit die FIDO-Allianz mit zu gestalten und bei der Führung teilzuhaben)
- Participation Benefits (Mitwirkung bei der FIDO-Allianz)
- Commercial Benefits (Kommerzielle Vorteile die eine Mitgliedschaft beinhaltet)

Die Mitgliedschaft der FIDO-Allianz wird in folgende Arten aufgeteilt:

Board
Als Board-Mitglied kann ein Unternehmen alle Vorteile der FIDO-Allianz nutzen und bei Entscheidungsfindungen teilhaben. Ein Board-Mitglied wird ein Teil der Führung der FIDO-Allianz und besitzt damit eine große Verantwortung. Somit kann ein Board-Mitglied alle drei Arten der Vorteile vollständig nutzen. Die Gebühr für eine Mitgliedschaft beträgt $ 50.000.

Sponsor
Die Sponsor-Mitgliedschaft kann Arbeitsgruppen leiten und an Veranstaltungen teilnehmen. Bei einer Sponsor-Mitgliedschaft muss eine Gebühr von $ 25.000 bezahlt werden.

Associate
Bei der Associate-Mitgliedschaft wird den Mitgliedern die wenigste Verantwortung zugeteilt. Diese können aber auch nur wenige Vorteile der FIDO-Mitgliedschaft nutzen. Um Associate-Mitglied werden zu können, muss ein Betrag zwischen $ 2.500 - $15.000 bezahlt werden.

Für die Nutzung von Spezifikationen der FIDO-Allianz ist eine Mitgliedschaft nicht notwendig. Neben den Standards bzw. Spezifikationen, besteht die Möglichkeit ein Produkt, welches durch die FIDO-Authentifizierung erweitert wurde, zu zertifizieren. Diese Zertifizierung, welche im vierten Kapitel vorgestellt wird, wird ebenfalls von der FIDO-Allianz Angebot. Dabei muss das Produkt verschiedene Phasen durchlaufen und am Ende hat es die Möglichkeit ausgezeichnet zu werden. Durch diese Auszeichnung wird aufgezeigt, dass das Produkt zu anderen von der FIDO-Allianz zertifizierten Produkten kompatibel ist und über das UAF- oder U2F-Protokoll kommunizieren kann [Certification].

Die Anzahl der zertifizierten Produkte sowie die aktuellen Führungsmitglieder der FIDO-Allianz werden im sechsten Kapitel vorgestellt und die Entwicklung der FIDO-Allianz betrachtet.

3.2 Universal Authentication Framework (UAF)

Das UAF-Authentifizierungsverfahren steht auch für das Passwortlose Authentifizieren und funktioniert nach folgendem Prinzip. Der Benutzer registriert ein Gerät, oft ein Mobiltelefon, an einem Server. Für die Registrierung wird z.B. eine PIN-Nummer, ein Vergleich des Fingerabdrucks, der Stimme oder des Gesichts benötigt. Da das Mobiltelefon am Server registriert ist, ist auch nur das Mobiltelefon, und somit auch implizit der Eigentümer des Mobiltelefon, in der Lage sich am Server zu authentifizieren. Dafür muss der Benutzer sich zunächst lokal am Mobiltelefon authentifizieren und anschließend kann das Mobiltelefon sich am Server authentifizieren. Die lokale Authentifizierung muss durch den selben Faktor, welche auch bei der Registrierung benutzt wurde (z.B. durch Scannen des Fingerabdruckes), durchgeführt werden. [Approach-Vision] Das nachfolgende Bild soll einen ersten groben Überblick über die FIDO-UAF Architektur geben.

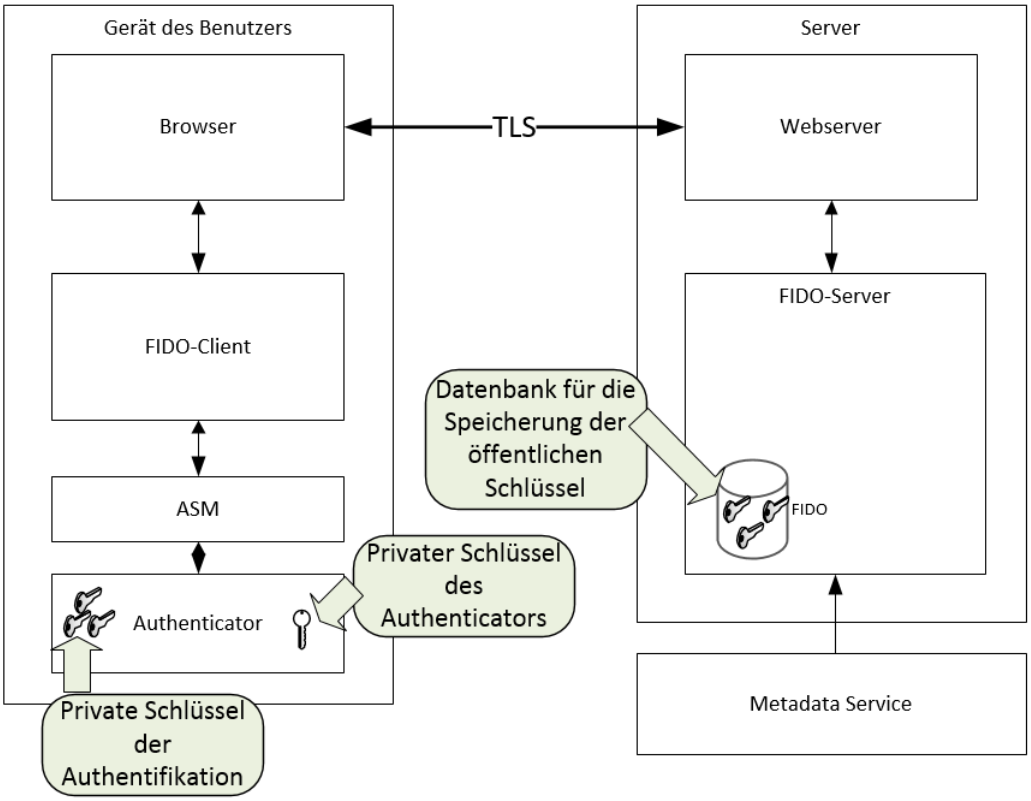

Abbildung 3.1: High-Level FIDO-UAF Architektur [UAF-Overview (2014)]

TLS:
Bevor Nachrichten zwischen dem Gerät des Benutzers und dem Server ausgetauscht werden, wird eine sichere Verbindung mittels TLS aufgebaut. Der sichere Verbindungskanal besitzt eine TLS-Kanalkennung. Die UAF-Nachrichten, die von der Client-Seite an den Server gesendet werden, beinhalten diese TLS-Kanalkennung. Der Server ist somit in der Lage, die TLS-Kanalkennung zu überprüfen. Aufgrund dieser Überprüfung ist es Angreifern nicht mehr Möglich, zwischen den beiden Kommunikationspartnern zu stehen, Einsicht in die versendeten Nachrichten zu erhalten oder diese sogar zu manipulieren. [UAF-Specification (2014)]

FIDO-Client:
Der FIDO-Client interagiert mit dem Authenticator durch das Nutzen der API des Authenticator-Specific-Module, kurz ASM. Außerdem Kommuniziert der FIDO-Client durch den User-Agent mit dem Server. Der FIDO-Client setzt die Client-Seite des UAF-Standards um. [UAF-Overview (2014)]

FIDO-Server
Der FIDO-Server interagiert mit dem Server, um UAF-Protokoll Nachrichten mit dem FIDO-Client austauschen zu können. Außerdem verifiziert der FIDO-Server den Authenticator indem es den Metadata Service nutzt. Weiterhin verwaltet es die öffentlichen Schlüssel der Benutzer und authentifiziert Benutzer. [UAF-Overview (2014)]

Wie in der Architektur (Abb. 3.1) dargestellt, existiert in dem UAF-Protokoll der bereits erwähnte Authenticator. Dieser Authenticator wird im nächsten Abschnitt genauer erläutert. Außerdem ist das Verstehen der Registrierung, der Authentifizierung und der Transaktion für das Verstehen des Kommunikationsflusses sehr wichtig, somit auch essentiell für die Grundlagen der FIDO-Authentifizierung. In den nachfolgenden Abschnitten wird dies genauer erklärt.

3.2.1 Authenticator

In der FIDO-Authentifizierung spielt der Authenticator eine wichtige Rolle. Sie ist eine Sicherheitskomponente und befindet sich auf dem Gerät des Benutzers, also auf der Client-Seite. Der Authenticator ist für die Verwaltung der Schlüssel sowie für das Erzeugen des Schlüsselmaterials zuständig. Außerdem besitzt der Authenticator einen eigenen privaten Schlüssel, welcher nicht erneuert oder verändert werden kann. Beispielsweise wird bereits bei der Herstellung von Mobiltelefonen der FIDO-Authenticator integriert. Der Authenticator besitzt also schon bereits bei der Herstellung des Mobiltelefons einen privaten Schlüssel und kann somit Signierungen durchführen [UAF-Overview (2014)]. Der Benutzer ist bereits mit dem Erhalt eines Mobiltelefons im Besitzt des Authenticators. Da der Authenticator die Schlüsselgenerierung für die Authentifizierung übernimmt, wird implizit der Beweis der Identität mit dem Besitzt des Mobiltelefons erbracht.

3.2.1.1 Authenticator Metadata Service

Der Authenticator besitzt einen privaten Schlüssel zum Signieren, jedoch wird zum Verifizieren der öffentliche Schlüssel des Authenticators benötigt. Diesen öffentliche Schlüssel erhält man nicht vom Gerät, beispielsweise Mobiltelefon, sondern von einer vertrauenswürdigen Stelle, durch das Nutzen des Authenticator Metadata Service. Eine vertrauenswürdige Stelle könnte beispielsweise der Hersteller des Mobiltelefons sein. Der passende öffentliche Schlüssel wird anhand der Authenticator Attestion ID, kurz AAID, gefunden. Unabhängig vom Gerät oder Hersteller ist die AAID einmalig und der Authenticator kann aufgrund dieser Eigenschaft identifiziert werden. Nach der Identifikation ist der Metadata Service in der Lage ein Zertifikat, genannt Attestation Certificate, an den Server zu senden. In diesem Zertifikat ist der öffentliche Schlüssel des Authenticators zu finden. Das Zertifikat kann anhand der Vertrauenskette, welches im Grundlagenkapitel erklärt wurde, validiert werden. [Metadata-Service (2014)]

3.2.2 Registrierung

Die Registrierung ist eines der Kommunikationsbereiche des UAF-Standards. Die Registrierung beginnt der Server. Es ist möglich, dass der Benutzer bereits an dem Online-Service, wie z.B. einem Webserver, mit einem anderen Authentifizierungsverfahren authentifiziert ist, beispielsweise mit einem Benutzernamen und Passwort, oder sich neu registriert. [UAF-Specification (2014)]

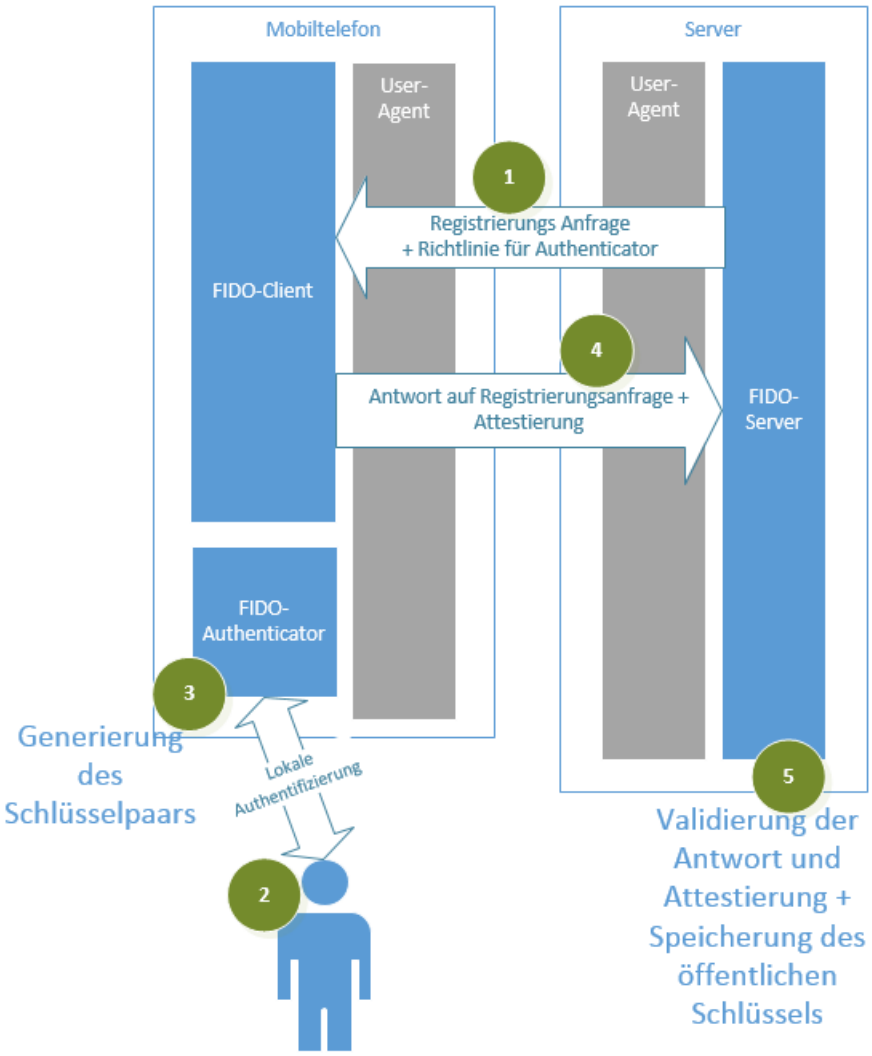

Abbildung 3.2: UAF-Registrierung [UAF-Overview (2014)]

Registrierungsanfrage + Richtlinie für Authenticator
Die Registrierungsanfrage geht von dem Server aus. Bei dieser Anfrage wird der Benutzername, die Richtlinien für den Authenticator sowie eine Challenge an den Client gesendet. Anhand der Richtlinien kann der FIDO-Client den passenden Authenticator wählen. Der Server kann in den Richtlinien beispielsweise vorgeben, welche lokale Authentifizierung der Benutzer für die Authentifizierung benutzen darf (z.B. Scannen des Fingerabdrucks).

Lokale Authentifizierung
Wurde die passende Authentifizierungsmethode gewählt (z.B. Scannen des Fingerabdrucks), muss der Benutzer sich am Gerät lokal authentifizieren.

Generierung des Schlüsselpaars
Nach der lokalen Authentifizierung des Benutzers kann der Authenticator ein Schlüsselpaar erzeugen, welcher für zukünftige Authentifizierungen am Server benutzt werden kann. Außerdem erstellt der Authenticator mit seinem privaten Schlüssel eine Attestierung. Diese Attestierung beinhaltet die AAID des Authenticators, die Challenge sowie den generierten öffentlichen Schlüssel, welcher für den Authentifizierungsvorgang generiert wurde,

Antwort auf Registrierungsanfrage + Attestierung
Nachdem der Authenticator die lokale Authentifizierung, die Generierung des Schlüsselpaares und die Erstellung der Attestierung vollzogen hat, übergibt der Authenticator diese Daten an den FIDO-Client. Der FIDO-Client erstellt nun eine Antwort auf die Registrierungsanfrage. Die Antwort beinhaltet die AAID sowie den öffentlichen Schlüssel. Die Antwort und die Attestierung wird nun vom FIDO-Client über den User-Agent an den FIDO-Server gesendet.

Validierung der Antwort und Attestierung + Speicherung des öffentlichen Schlüssels
Mit dem Erhalt der Attestierung und der Registrierungsantwort ist der FIDO-Server in der Lage die Registrierung zu verifizieren. Dafür nimmt der FIDO-Server den öffentlichen Schlüssel des Authenticators, welche er zuvor gespeichert oder durch den Metadata Service erhalten hat. Nach der Verifizierung kann der öffentliche Schlüssel des Benutzers gespeichert werden.

3.2.3 Authentifizierung

Die Authentifizierung ist ein weiterer Kommunikationsbereich des UAF-Protokolls. Der Benutzer wird ohne Eingabe eines Passworts am Online-Service authentifiziert, es findet lediglich eine lokale Authentifizierung des Benutzers statt. Aufgrund dessen, wird das UAF-Standard auch das passwortlose Authentifizieren genannt. Selbstverständlich muss ein Endgerät bereits am Online-Service, wie in dem Part der Registrierung bereits beschrieben, registriert sein. [UAF-Specification (2014)]

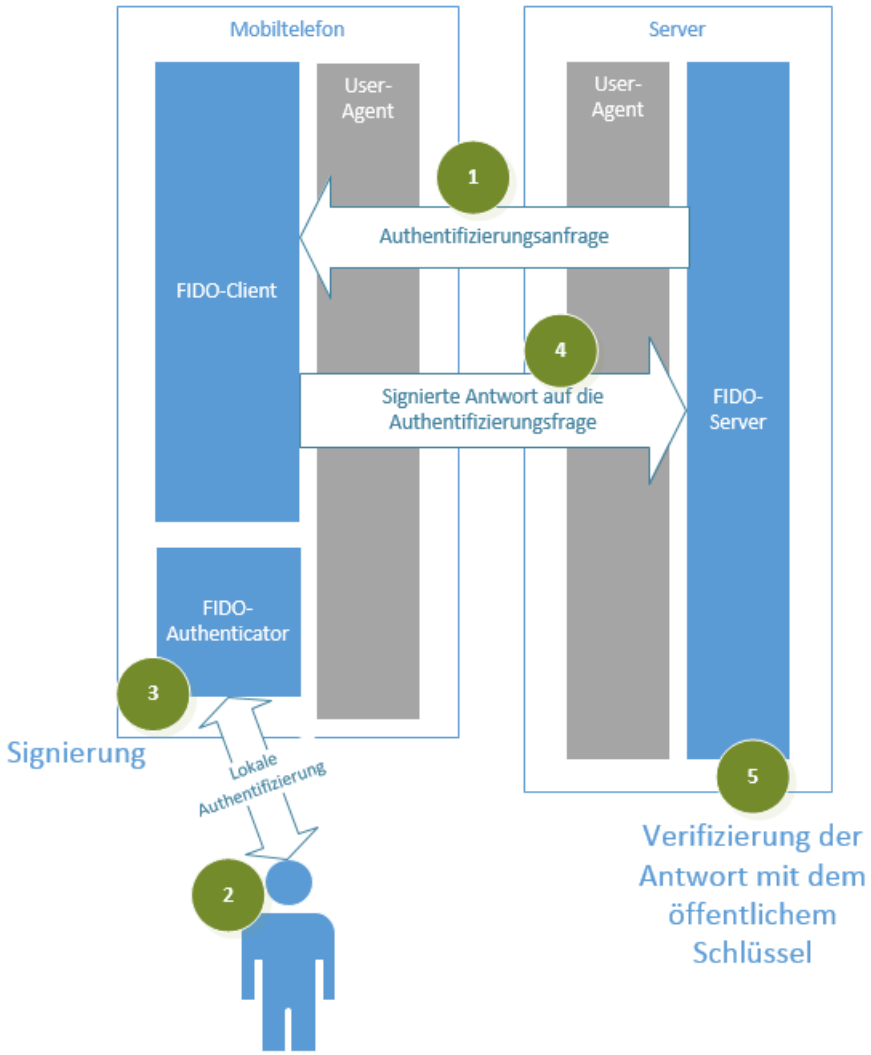

Abbildung 3.3: UAF-Authentifizierung [UAF-Overview (2014)]

Authentifizierungsanfrage
Der Server sendet eine Authentifizierungsanfrage an den FIDO-Client. Die Anfrage beinhaltet eine Challenge, sowie Richtlinien für den Authenticator. Der Authenticator muss der selbe Authenticator sein, welcher auch bei der Registrierung benutzt wurde. Außerdem muss der Authenticator den Richtlinien entsprechen.

Lokale Authentifizierung
Wurde der passende Authenticator gewählt, muss der Benutzer sich am Gerät lokal authentifizieren. Da der selbe Authenticator wie bei der Registrierung benutzt wurde, ist dem Authenticator

der für die Authentifizierung benötigte private Schlüssel bekannt.

Signierung
Wurde der Benutzer lokal erfolgreich authentifiziert, wird anhand des in der Registrierung generierten privaten Schlüssels die Challenge, welche von dem Server gesendet wurde, signiert.

Signierte Antwort auf die Authentifizierungsanfrage
Die Signierung, welche der Authenticator vorgenommen hat, wird an den FIDO-Client weitergeleitet. Der FIDO-Client sendet diese signierte Antwort weiter an den Server.

Verifizierung der Antwort mit dem öffentlichem Schlüssel
Mit dem Erhalt der signierten Antwort kann der Server die Antwort mit dem gespeicherten öffentlichen Schlüssel verifizieren und die Challenge mit der vom Server gesendeten Challenge vergleichen. War dies erfolgreich, ist der Benutzer am Server authentifiziert.

3.2.4 Transaktion

In manchen Fällen ist es notwendig den Benutzer bei einer bestimmten Transaktion nochmals zu authentifizieren. Beispiel hierfür wäre eine Geld-Transaktion bzw. Überweisung, welches über das Internet getätigt wird. Vor solch einer Überweisung muss ein Benutzer diese Transaktion oft freigeben. Diese Freigabe ähnelt dem Authentifizierungsprozess. Bei der Transaktion wird zusätzlich zu der Challenge, welche durch den Authenticator signiert wird, eine für den Menschen lesbarer Text an den Benutzer mitgesendet. Dieser Text wird dem Benutzer angezeigt und anschließend wird vom Benutzer die Transaktion, beispielsweise durch den Scan des Fingerabdrucks, freigegeben. Aus dem Text wird anschließend ein Hashwert gebildet und mit dem privaten Schlüssel signiert. Der FIDO-Server verifiziert nun nicht nur die Authentifizierungsanfrage, sondern auch den signierten Hashwert. [UAF-Overview (2014)]

3.3 Universal 2nd Factor (U2F)

Der U2F-Standard gehört zu den beiden Standards der FIDO-Allianz. Dieser Standard wurde dazu konzipiert, eine bereits bestehende Authentifizierungsmethode um einen zweiten Faktor zu erweitern. Durch die U2F-Erweiterung wird die Abhängigkeit an einen starken Passwort reduziert. Benutzer können selbst Passwörter wählen, die lediglich aus der Kombination von vier Zahlen bestehen. Dies macht das U2F-Standard möglich, indem zusätzlich zu der bestehenden Authentifizierungsmethode beispielsweise ein USB-Stick benutzt wird. Wie auch beim UAF-Standard, wird in dem U2F-Standard die Eigenschaften der asymmetrischen Kryptographie verwendet. Bei dem U2F-Standard ist die Registrierung des U2F-Geräts und die Authentifizierung möglich. [U2F-Overview (2016)]

3.3.1 Registrierung

Die Registrierung einer zusätzlichen Authentifizierungsmethode anhand des U2F-Standards funktioniert wie folgt. [U2F-Overview (2016)]

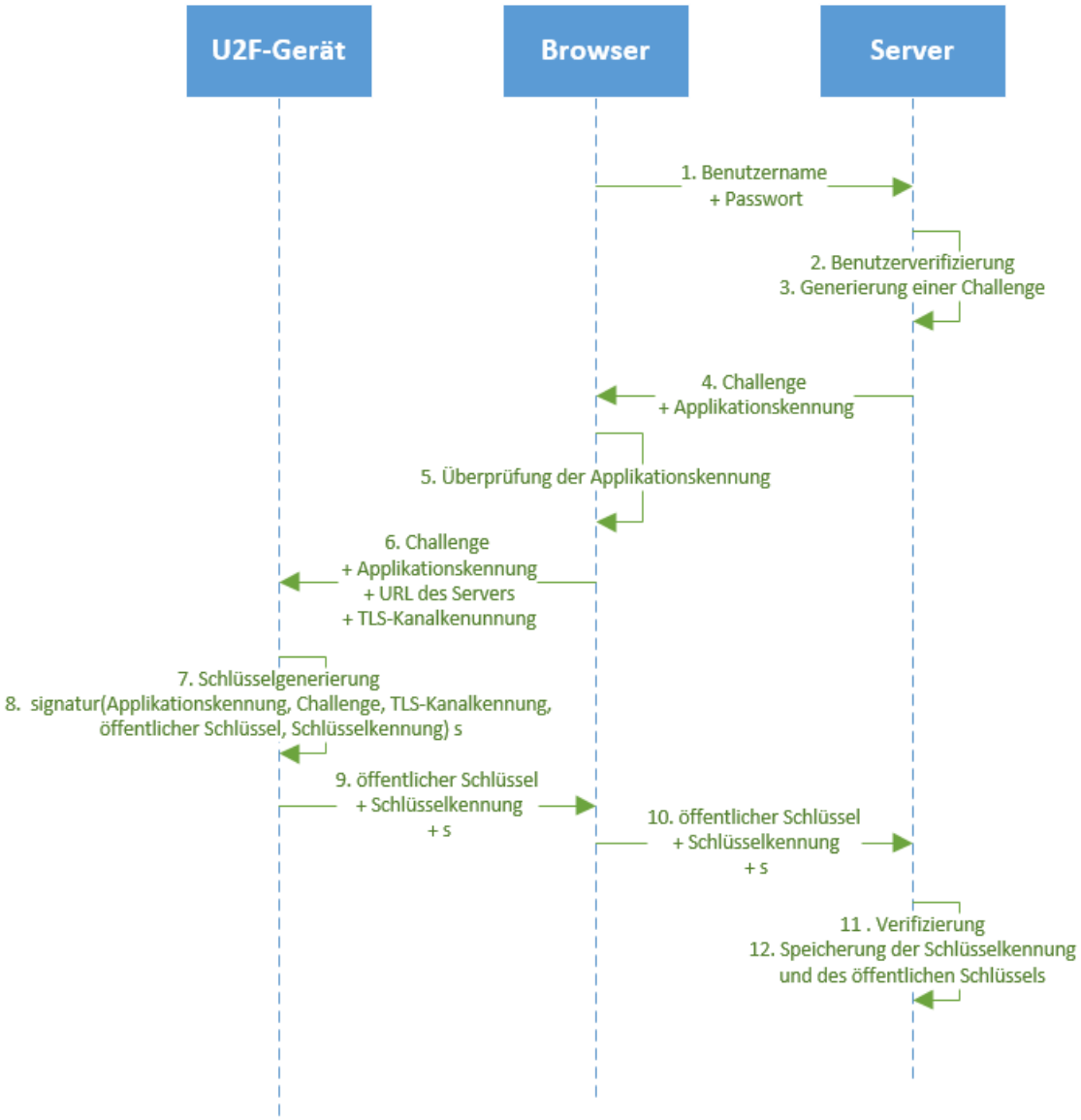

Abbildung 3.4: U2F-Registrierung

- Nach einer erfolgreichen Authentifizierung des Benutzers, beispielsweise durch die Eingabe des Benutzernamens und Passworts, sendet der Server eine Challenge und eine Applikationskennung an den Browser.
- Der Browser überprüft die Applikationskennung. Anschließend generiert der Browser, anhand der Applikationskennung, Challenge, TLS-Kanalkennung und der URL des Servers, eine Anfrage, Diese Anfrage leitet der Browser weiter an das U2F-Gerät.
- Das U2F-Gerät generiert anhand der Applikationskennung ein asymmetrisches Schlüsselpaar.
- Anschließend erstellt das U2F-Gerät anhand der Applikationskennung, Challenge, TLS-Kanalkennung, dem öffentlichen Schlüssel und der Schlüsselkennung eine digitale Signatur. Die digitale Signatur, der öffentliche Schlüssel und die Schlüsselkennung sendet das U2F-Gerät an den Browser, welcher diese Informationen weiter an den Server leitet.
- Der Server verifiziert die Signatur mit dem öffentlichen Schlüssel und anschließend die TLS-Kanalkennung.
- Nach der Verifizierung speichert der Server den öffentlichen Schlüssel und die Schlüsselkennung.

3.3.2 Authentifizierung

Nach einer erfolgreichen Registrierung, wird das U2F-Gerät für die Authentifizierung wie foglt Benutzt. [U2F-Overview (2016); U2F-YUBICO-OVERVIEW]

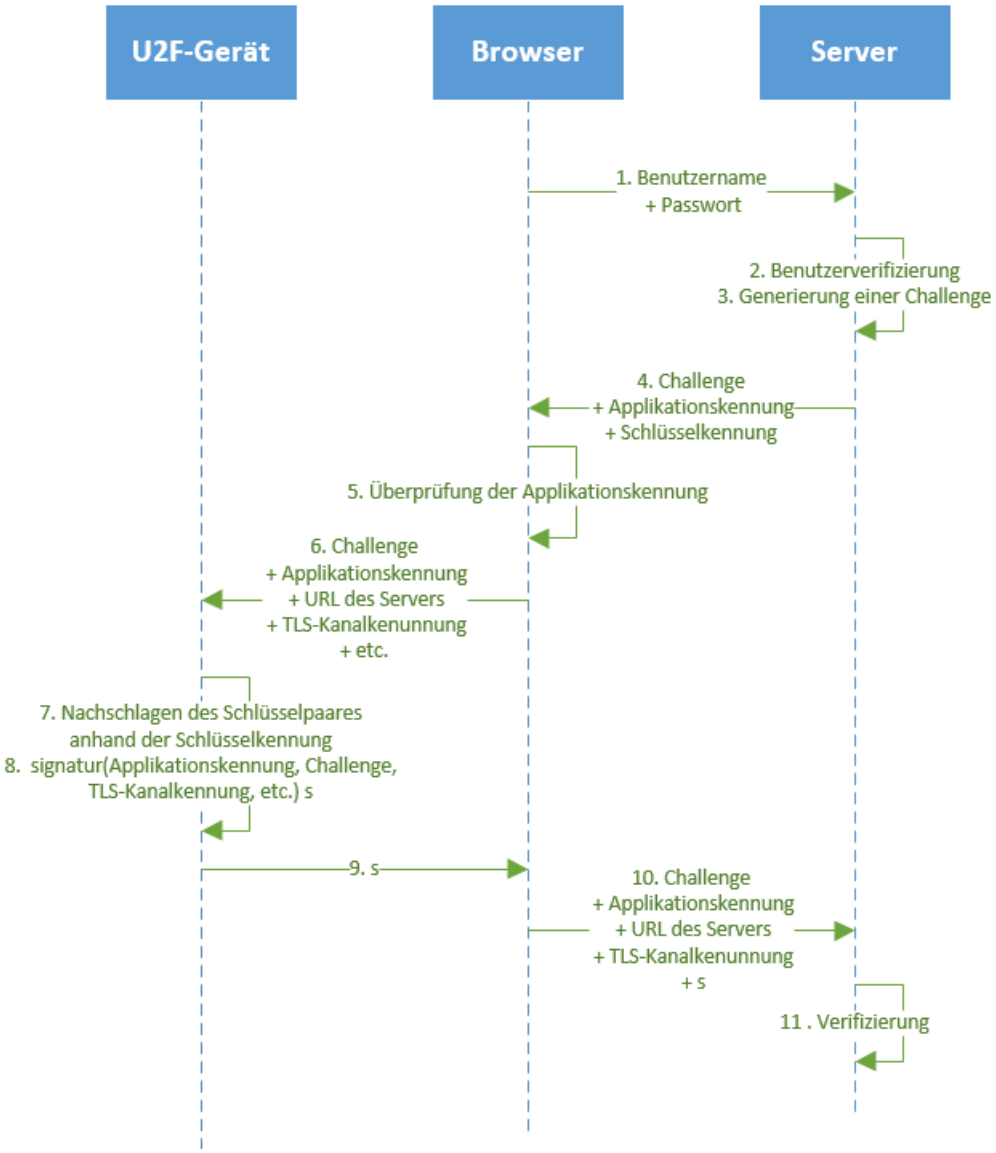

Abbildung 3.5: U2F-Registrierung

- Nach einer erfolgreichen Authentifizierung des Benutzers, beispielsweise durch die Eingabe des Benutzernamens und Passworts, sendet der Server eine Challenge, die Applikationskennung und die Schlüsselkennung an den Browser.

- Der Browser überprüft die Applikationskennung. Anschließend generiert der Browser, anhand der Applikationskennung, Challenge, TLS-Kanalkennung, Schlüsselkennung und weitern Informationen, eine Anfrage. Diese Anfrage sendet der Browser weiter an das U2F-Gerät.
- Anhand der Schlüsselkennung wählt das U2F-Gerät den passenden Schlüssel und generiert mit den erhaltenen Informationen eine digitale Signatur. Die Signatur leitet das U2F-Gerät weiter an den Browser.
- Die Signatur sowie die Informationen, wie beispielsweise die Challenge und die TLS-Kanalkennung, sendet der Browser weiter an den Server.
- Der Server ist bereits im Besitz des öffentlichen Schlüssels und kann somit die Signatur verifizieren. Anschließend kann der Server die TLS-Kanalkennung verifizieren.

4 Zertifizierung

In dem Kontext der FIDO-Authentifizierung wird der Begriff Certification (Zertifizierung), als ein Ablauf verstanden, in dem das Produkt bestimmten Phasen durchläuft. Diese Zertifizierung wird von der FIDO-Allianz angeboten. Hat man jede Phase erfolgreich durchlaufen, besteht die Möglichkeit das Produkt zu kennzeichnen. Diese Kennzeichnung beweist Kunden oder anderen Unternehmen die korrekte Implementation eines FIDO-Standards. Außerdem zeigt solch eine Kennzeichnung, dass das Produkt, welches erfolgreich durch eine FIDO-Implementierung erweitert wurde, in der Lage ist mit anderen FIDO-Implementierung eine sichere Zusammenarbeit zu gewährleisten. Somit wird dies für Partner besonders interessant. In den nächsten Abschnitten werden die einzelnen Phasen bis zum Erhalt des Zertifizierungs-Logos genauer erläutert. [Certification]

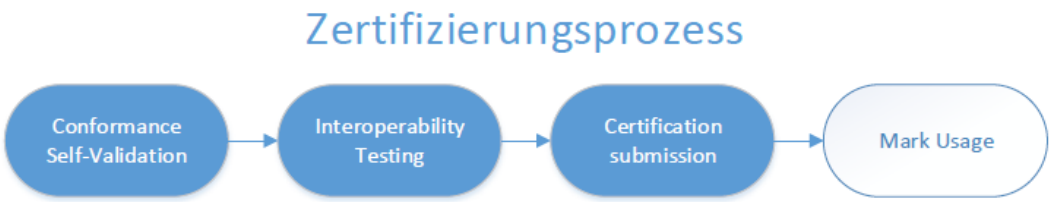

Abbildung 4.1: Zertifizierungsprozess

4.1 Conformance Self-Validation

Die Conformance Self-Validation, also die Selbst-Validierung, ist ein notwendiger Schritt in dem Zertifizierungsprozess. Um das Produkt oder den Service selbst validieren zu können, werden von der FIDO-Allianz Werkzeuge für das Testen angeboten. Unterschieden wird anhand der zwei Standards, somit gibt es das UAF-Test-Tool und U2F-Test-Tool. Beide Werkzeuge erfordern eine vorherige Registrierung. Die FIDO-Allianz möchte in der Registrierung wissen, zu welchem Unternehmen man angehört und ob das Unternehmen bereits ein Mitglied der FIDO-Allianz ist. Nach der Registrierung besteht Zugang zu den Werkzeugen, diese Werkzeuge werden anhand von zwei Bereichen unterschieden. Das manuelle und das automatisierte Testen. [Conformance]

Manuelles Testen
Es ist notwendig den Client sowie den Server manuell zu testen. Mit Hilfe des Inhaltes werden die Registrierungs-, Authentifizierungs- und Deregistrierungsnachricht, welche im Ablauf der

FIDO-Authentifizierung erzeugt werden, getestet. Außerdem wird der Authenticator anhand der Funktionalitäten getestet. Hat man alle Tests bestanden, werden die automatisierten Tests durchlaufen.

Automatisiertes Testen
Der Server und der Client werden in diesem Bereich anhand der FIDO-Spezifikation getestet. Hierbei muss der Server einen Adapter oder einen Proxy besitzen, der zwischen dem Test-Tool und dem UAF-Server agiert. In dem Test-Tool kann man nun die URL des Adapters angeben und die automatisierten Tests starten.

4.2 Interoperability Testing

Nach der "Conformance Self-Validation" ist der nächste notwendige Schritt im Zertifizierungsprozess, das Interoperability Testing. In diesem Abschnitt des Zertifizierungsprozesses besteht die Möglichkeit die Implementation anhand der zwei folgenden Optionen zu testen. [Interoperability]

4.2.1 Interoperability Testing Event

Das Interoperability Testing Event ist eine Veranstaltung, bei der Entwickler ihre Implementationen auf Kompatibilität testen und validieren können. Bezogen auf die UAF-Spezifikationen bedeutet dies, dass beispielsweise ein UAF-Client mit allen an der Veranstaltung teilnehmenden UAF-Servern und Authenticators auf Kompatibilität getestet wird. Auf diese Weise wird sichergestellt, dass alle Implementationen mit allen anderen Implementationen der Veranstaltung funktionieren.

Solche Veranstaltungen finden, laut der FIDO-Allianz, alle 90 Tage statt. Um sich für die Veranstaltung registrieren zu können, muss das "Conformance Self-Validation" erfolgreich abgeschlossen worden sein und die Implementation darf vor der Veranstaltung nicht verändert werden. Außerdem besteht für teilnehmende Unternehmen die Möglichkeit, an einem "pre-testing" teilzunehmen. In dem "pre-testing", welches vor der eigentlichen Veranstaltung stattfindet, wird den Entwicklern gezeigt, wie die Implementation miteinander getestet und Metadaten geteilt werden.

4.2.2 On Demand Testing

Die Alternative zu dem Interoperability Testing Event ist das On Demand Testing. Statt einer Teilnahme an einer Veranstaltung besteht die Möglichkeit, die eigene Implementationen mit bereits von der FIDO-Alliance zertifizierten Implementationen auf Kompatibilität zu testen. Hierfür können Unternehmen ihre zertifizierten Implementationen zur Verfügung stellen. In dem On Demand Testing besteht die Möglichkeit Implementationen anhand einer von drei folgenden Optionen zu testen.

Virtual
In der Option "Virtual" müssen Zugriffe auf die Schnittstellen der Implementation für die Zertifizierungsstelle der FIDO-Allianz freigegeben werden. Außerdem ist es notwendig eine Kontaktperson für die Zertifizierungsstelle bereitzustellen, um Fragen oder Probleme klären zu können.

Shipped
Wird die Option "Shipped" gewählt, muss die Software an die Zertifizierungsstelle der FIDO-Allianz versendet werden. Auch in dieser Option ist eine Kontaktperson notwendig. Die Zertifizierungsstelle teilt die Testergebnisse dem Unternehmen mit und versendet die Software dem Unternehmen zurück.

In-Person
Soll die Software mit der Option "In-Person" getestet werden, wird die Testphase vom Unternehmen begleitet. Hierfür reist ein Tester aus der Zertifizierungsstelle der FIDO-Allianz zu dem Unternehmen und führt notwendige Tests in drei Testphasen durch. Jede einzelne Testphase darf maximal drei Tage andauern. In dieser Option fallen höhere Kosten für das Unternehmen an. Jede zu testende Implementierung kostet dem Unternehmen $ 10.000. Auch die Reisekosten des Testers müssen vom Unternehmen getragen werden.

4.3 Certification Submission

Der letzte notwendige Schritt in dem Zertifizierungsprozess ist die "Certification Submission". Wurde die Implementation durch das "Conformance Self-Validation" und das "Interoperability Testing" erfolgreich getestet, muss ein Formular, die "FIDO Vendor Self-Assertion Checklist", ausgefüllt werden. In diesem Formular sichert das Unternehmen der FIDO-Allianz zu, die Sicherheitsanforderungen der FIDO-Allianz zu erfüllen. Folgende Punkte sind Ausschnitte aus der "FIDO Vendor Self-Assertion Checklist": [Certification-Submission]

- Nach der "Conformance Self-Validation", dürfen an der zu zertifizierenden Software nur Änderungen vorgenommen werden, wenn die Einhaltung der FIDO-Spezifikationen gewährleistet wird.
- Metadaten des Authenticators, müssen mit den Eigenschaften der von der FIDO-Allianz für den FIDO-Authenticator vorgeschriebenen Eigenschaften übereinstimmen.

Sind alle Anforderungen erfüllt, muss noch eine Gebühr für die Zertifizierung bezahlt werden. Mitglieder der FIDO-Allianz müssen weniger zahlen. So müssen Mitglieder für die Zertifizierung eine Gebühr in Höhe von $5.000 USD bezahlen. Besteht keine Mitgliedschaft, ist eine Gebühr

in Höhe von $6.500 USD zu bezahlen. Nach der Bezahlung und der Einreichung der Software kann eine Zertifizierung bis zu 10 Werktagen andauern.

4.3.1 Derivative Certification

Das “Derivative Certification” ist für Unternehmen gedacht, die mehrere Implementationen anhand einer bereits zertifizierten Implementation zertifizieren wollen. Die Implementationen, die per “Derivative Certification” zertifiziert werden sollen, müssen nicht das “Interoperability Testing” durchlaufen. Ein weiterer Vorteil besteht darin, dass die Gebühr für Implementation, die per “Derivative Certification” zertifiziert werden sollen, geringer ist. Auch hier spielt es eine Rolle, ob das Unternehmen Mitglied in der FIDO-Allianz ist. Mitglieder zahlen pro Zertifikat eine Gebühr von $ 500. Ist man kein Mitglied, bezahlt man $ 750 pro Zertifizierung. Folgende Beispiele sollen zeigen, wann eine Implementation für die “Dertivative Certification” geeignet ist und wann nicht.

- Falls eine neue Version einer Software entwickelt wurde, jedoch die FIDO-Komponenten nicht verändert wurden, so ist die neue Version für die Zertifizierung durch die “Derivative Certification” geeignet.
- Falls eine neue Version einer Software entwickelt und dabei auch die FIDO-Komponenten angepasst, hinzugefügt oder entfernt wurden, so muss der gesamte Zertifizierungsprozess erneut durchlaufen werden.

4.4 Mark Usage

Eine Kennzeichnung der Software bzw. der Implementierung durch das FIDO-Logo ist dem Unternehmen überlassen. Nach einer Unterzeichnung der Vereinbarung über das Benutzen des FIDO-Logos, kann die Software gekennzeichnet werden. Durch die Kennzeichnung der Software zeigt man Kunden und Partner, dass die Sicherheitsanforderungen und Datenschutzrichtlinien der FIDO-Allianz erfüllt wurden. Außerdem ist es so für Partner möglich, von einer sicheren Kompatibilität mit der eigenen, von der FIDO-Allianz zertifizierten, Software ausgehen zu können. [Mark-Usage]

5 Vergleich mit anderen Authentifizierungsverfahren

Nachdem ein gewisses Grundverständnis über die FIDO-Authentifizierung und dessen Zertifizierung besteht, kann ein Vergleich mit traditionellen Authentifizierungsverfahren erfolgen. In dem Kontext dieser Bachelorarbeit wird zu den traditionellen Authentifizierungsverfahren die wissensbasierte Authentifizierung mit Benutzernamen und Passwort, die Authentifizierung mittels One-Time-Passwords und die Client-Authentifizierung gezählt. Da der UAF-Standard etablierte Authentifizierungsverfahren ablösen möchte, werden im Folgenden die etablierte Authentifizierungsverfahren mit dem UAF-Standard verglichen. Bei dem Vergleich wird bei der FIDO-Authentifizierung davon ausgegangen, dass das UAF-Protokoll angewandt und die Authentifizierung mittels Fingerabdruck am Mobiltelefon ausgeführt wird.

5.1 Wissensbasierte Authentifizierung (Benutzername und Passwort)

Die wissensbasierte Authentifizierung ist die am meist verbreitetste Authentifizierungsmethode [Evoulution-of-Authentication (2010)]. In fast jeder Webseite sind zwei Felder zu sehen. Ein Feld für die Eingabe des Benutzernamens und ein Feld für die Eingabe des Passwortes.

5.1.1 Prinzip

Die wissensbasierte Authentifizierung basiert auf lediglich einem Faktor - das Wissen über ein Geheimnis. Dieses Geheimnis ist das Passwort und wird vom Benutzer gewählt. Hat der Benutzer sich für ein Passwort entschieden, so muss das Passwort über eine sichere Verbindung, z.B. eine mit TLS-verschlüsselten Kommunikationskanal, an den Server übermittelt werden. Ist das Passwort sicher am Server angelangt, wird das Passwort vor der Speicherung mit einer Hashfunktion und beliebigen Daten, genannt Salt, verändert. Der aus dem Passwort und Salt gewonnene Hashwert wird nun stellvertretend für das echte Passwort in der Datenbank gespeichert. Somit kann man das originale Passwort nicht mehr aus der Datenbank ablesen und der Salt führt dazu, dass selbst gleiche Passwörter in der Datenbank anders aussehen.

5.1.2 Vergleich

Beim authentifizieren mittels Passwort besteht die Gefahr an dem Passwort selbst. Oft werden zu einfache Passwörter gewählt und können von dritten selbst durch ausprobieren erlangt werden. Deshalb sind sichere Passwörter und eine sichere Übertragung des Passwortes besonders wichtig. Folgende Voraussetzung sollte ein Passwort erfüllen [Security-Advice (2009)]:

1. **Kein zu kurzes Passwort:** Das deutsche Alphabet besitzt, ohne Umlaute, 52 Zeichen für die Groß- und Kleinschreibung. Hat ein Benutzer nur die Auswahl an diesen 52 Zeichen, so sind bei einer Passwortlänge von fünf, 380.204.032 Kombinationen möglich und bei einer Passwortlänge von sechs, 19.770.609.664. Entscheide man sich nun sechs statt fünf Zeichen zu wählen, sind ganze 19.390.405.632 Kombinationen mehr möglich. Erhöht man also die Passwortlänge, so erhält man ein immer sichereres Passwort.

2. **Zusammensetzung eines Passworts:** Das Passwort sollte auch Ziffern und Sonderzeichen beinhalten. Somit sind mehr Kombinationen und ein sichereres Passwort möglich.

3. **Passwort sollte kein Wort aus einem Wörterbuch beinhalten:** Unabhängig von der Länge eines Passworts können Angreifer Wörter aus dem Wörterbuch benutzen, um an das Passwort zu gelangen. Aus diesem Grund sollten Passwörter keine Wörter beinhalten.

4. **Passwort sollte nicht aufgeschrieben werden.**

5. **Passwort sollte niemandem mitgeteilt werden.**

6. **Passwort sollte oft geändert werden.**

7. Für **jedes Konto bzw. Profil**, dass eine Authentifizierung erfordert, sollte **ein eigenes Passwort verwendet** werden**.**

Wie hier zu sehen ist, liegt eine große Sicherheitsverantwortung bei dem Benutzer selbst. Durch die Verwendung von Passwort-Richtlinien kann der Benutzer bei der Passwortvergabe beeinflusst werden. Jedoch liegt die Verantwortung der Punkte vier bis sieben allein bei dem Benutzer. Im Vergleich wird bei der FIDO-Authentifizierung dem Benutzer keine Verantwortung übergeben. Im Gegensatz zu dem Passwort bei der wissensbasierten Authentifizierung, sind Informationen des Benutzers, welche für die Authentifizierung relevant sind, durch den privaten Schlüssel des Benutzers attestiert. Bei einer Generierung eines privaten und öffentlichen Schlüsselpaares durch einen RSA-Algorithmus, kann die Länge des Schlüsselpaares angegeben werden. Die UAF-Spezifikation gibt eine Schlüssellänge von 2048 Bit an [UAF-Values (2013)] . Ein Passwort mit einer Länge von 8 Zeichen würde bei einer ASCII-Codierung lediglich 64 Bit betragen.

	Passwort	FIDO-Authentifizierung
Verantwortung	liegt beim Benutzer, kann zum Teil eingeschränkt werden	Schlüsselgenerierung wird vom Gerät durchgeführt
Sicherheit	Benutzer muss sich an Regeln halten, sehr langes und kompliziertes Passwort nicht praktikabel	Durch die Generierung am Gerät entsteht ein Schlüsselpaar. Vorgetäuschtes signieren kaum möglich.
Aufwand Benutzer	Benutzer sollte sich mehrere lange und komplizierte Passwörter ausdenken und merken. Passwortverwaltung liegt beim Benutzer	Schlüsselverwaltung übernimmt das Gerät
Geschwindigkeit bei der Authentifizierung	Durch das Besitzen von mehreren Passwörtern, kann das Erinnern und Eintippen eines Passworter viel Zeit in Anspruch nehmen	Der Benutzer muss sich lediglich lokal am Gerät authentifizieren (z.B. durch das Scannen des Fingerabdrucks).

Tabelle 5.1: Vergleich wissensbasierten Authentifizierung mit FIDO-Authentifizierung - 1

Wie bereits erwähnt, speichert der Server den aus dem Passwort und dem Salt erhaltenen Hashwert in der Datenbank. Dennoch muss für die Hashwert-Bildung das Passwort an den Server übertragen werden. Bei der FIDO-Authentifizierung wird der öffentliche Schlüssel übertragen. Da der öffentliche Schlüssel kein Geheimnis darstellt, ist die Relevants der Sicherheit vom versenden einer Nachricht, die den öffentlichen Schlüssel beinhalten, wesentlich geringer als das Versenden einer Nachricht, welches ein Passwort enthält.

	Passwort	Schlüssel
Übertragung	Geheimnis muss an den Server übertragen werden	Der öffentliche Schlüssel muss übertragen werden
Speicherung	Es muss ein Hashwert gebildet werden, welches stellvertretend für das Passwort in der Datenbank gespeichert wird	Öffentlicher Schlüssel wird in der Datenbank gespeichert

Tabelle 5.2: Vergleich wissensbasierten Authentifizierung mit FIDO-Authentifizierung - 2

Durch das Benutzen eines Passwortes ist man jedoch nicht an ein bestimmtes Gerät gebunden, welches die Schlüsselverwaltung übernimmt. Das bedeutet, dass die Authentifizierung ohne explizite Gebundenheit an einem Gerät funktioniert. Bei der FIDO-Authentifizierung muss das Gerät, welches die Authentifizierung vornimmt, den privaten Schlüssel kennen. Dies könnte mit eines der Gründe für die weite Verbreitung der wissensbasierten Authentifizierung

mittels Passwort sein.

	Authentifizierung mittels Passwort	FIDO-Authentifizierung
Abhängigkeit	Authentifizierung benötigt lediglich das Passwort sowie den Benutzernamen	Authentifizierung ist an das Gerät gebunden, welches den privaten Schlüssel besitzt
Bekanntheit	Die am meisten benutzte Authentifizierungsmethode	Wird momentan wesentlich weniger eingesetzt

Tabelle 5.3: Vergleich wissensbasierten Authentifizierung mit FIDO-Authentifizierung - 3

Ein weiterer Unterschied zwischen beiden Verfahren stellt die Standardisierung dar. Das UAF- sowie das U2F-Protokol sind standardisiert und werden bzw. sollten von jedem Unternehmen nach den vorgegebenen Spezifikationen eingesetzt werden. Dies wird anhand des Zertifizierungsprozesses sichergestellt und stellt somit auch die Kompatibilität der FIDO-Produkte zueinander dar. Bei der wissensbasierten Authentifizierung kann der Kommunikationsfluss, das Erstellen eines Hashwertes basierend auf dem Passwort oder das Speichern des Hashwertes von jedem Entwickler anders realisiert werden. Auch das Erstellen eines Hashwertes basierend auf dem Passwort und dem Salt ist nicht standardisiert. Der Benutzer weiß nicht, wie mit seinem Passwort umgegangen wird, wobei die FIDO-Authentifizierung öffentliche Schlüssel speichert.

	Authentifizierung mittels Passwort	FIDO-Authentifizierung
Standard	Keine Standards vorhanden. Fragestellungen bezüglich der Authentifizierung sind dem Unternehmen bzw. Entwickler überlassen.	FIDO-Produkte, welche das UAF- oder U2F-Protokoll umsetzen, müssen nach den vorgegebenen Spezifikationen realisiert werden.
Kenntnis der Benutzer	Benutzer ist über den Umgang mit seinem Passwort im Unklaren.	Benutzer kann bei zertifizierten Produkten mit Sicherheit davon ausgehen, dass nur der öffentliche Schlüssel gespeichert wird.

Tabelle 5.4: Vergleich wissensbasierten Authentifizierung mit FIDO-Authentifizierung - 4

5.1.3 One Time Password

Das One Time Password (OTP) ist eine generierte Zahlen- bzw. Zeichenfolge, welche für die Authentifizierung einer einzigen Transaktion oder Sitzung benutzt werden kann. Somit besteht ein Vorteil gegenüber dem klassischen Passwort. Sollten Unbefugte das klassische Passwort eines Benutzers besitzen, so sind sie in der Lage das Passwort für zukünftige Authentifizierungen zu nutzen (Reply-Angriff). Bei einem OTP ist dies nicht möglich, da ein OTP nach einmaligem verwenden nicht mehr für die Authentifizierung nutzbar ist. Aufgrund dieses Vorteils wurde das OTP-System entwickelt [RFC2289 (1998)]. Aus diesem OTP-System ist das zeit-basierte OTP (TOTP) sowie das ereignis-basierte OTP (HOTP) entstanden.

Zeitbasiertes OTP
Bei der zeit-basierten Variante wird ein OTP anhand der Zeit erstellt. Die Zeit ist einmalig, somit entsteht bei jeder Berechnung eines OTP ein neuer Wert. Ein Hauptmerkmal einer zeit-basierten OTP ist, dass diese neben den einmaligen Nutzbarkeit auch nur für eine gewisse Zeitspanne gültig ist.

Ereignis-basierte OTP
Die ereignis-basierte OTP wird anhand eines Zählers erstellt. Durch das inkrementieren des Zählers entsteht auch bei dieser Variante bei jeder Berechnung ein neuer Wert. Es können aufgrund des Zählers mehrere OTP erstellt werden, die nicht an eine Zeitspanne gebunden sind.

Folgender Vergleich soll die Unterschiede der zeit-basierten OTP und der ereignis-basierten OTP verdeutlichen: [OTP (2016)]

	Vorteil	Nachteil
Zeit-basiertes OTP	OTP ist nur während einer Zeitspanne gültig	OTP kann während der Eingabe die Gültigkeit verlieren
	OTP kann einfach abgelesen werden	OTP kann durch einfaches ablesen von Angreifer erlangt werden
Ereignis-basiertes OTP	Angreifer benötigen pyshikalischen Kontakt zum Gerät	OTP ist nicht an eine Zeitspanne gebunden und ist solange gültig bis ein OTP benutzt wird
	OTP Gültigkeit abhängig vom Benutzer	Benutzer muss OTP Generierung anstoßen

Tabelle 5.5: Vor- und Nachteile der TOTP und HOTP

Bei der OTP-Authentifizierung wird, aufgrund der Berechnung eines OTP, keine Verantwortung bei der Generierung eines Passwortes dem Benutzer überlassen. Somit verringert

sich auch der Aufwand für den Benutzer erheblich. Jedoch besteht bei der Authentifizierung mittels OTP eine Abhängigkeit zum Gerät, weil dieser für die Generierung der Passwörter verantwortlich ist. Jedoch besteht auch bei der OTP das Risiko in der Übertragung, da das OTP für die Authentifizierung übertragen werden muss.

	OTP	FIDO
Verantwortung	OTP wird am Gerät generiert	Schlüsselgenerierung wird vom Gerät durchgeführt
Sicherheit	Für jede Sitzung bzw. Transaktion wird ein eigenes Passwort verwendet. Wiederverwendung nicht möglich	Durch die Generierung am Gerät entsteht ein Schlüsselpaar. Vorgetäuschtes signieren kaum möglich.
Aufwand Benutzer	Kein Merken von Passwörtern notwendig. Passwortgenerierung übernimmt das Gerät	Schlüsselverwaltung übernimmt das Gerät
Abhängigkeit	Passwortgenerierung an Gerät gebunden	Authentifizierung ist an das Gerät gebunden, welches den privaten Schlüssel besitzt
Übertragung	Geheimnis muss an den Server übertragen werden	Nur der öffentliche Schlüssel muss übertragen werden

Tabelle 5.6: Vergleich OTP mit FIDO-Authentifizierung

5.2 Client-Authentifizierung mittels Zertifikaten

Die Client-Authentifizierung macht sich die Eigenschaften der asymmetrischen Kryptographie zu nutze. Der Client verwendet für die Authentifizierung ein Zertifikat, welches beispielsweise dem Browser hinzugefügt wird. Dieses Zertifikates ist oft nach dem, wie in dem Kapitel der technischen Grundlagen dargestelltem, x.509 Standard standardisiert. Der Unterschied gegenüber dem Server-Zertifikat sind folgende: [Client-Server-Certificates (2012)]

Server-Zertifikat	Client-Zertifikat
Zertifikat wird an den Hostnamen des Server ausgestellt. (z.B. CN = www.example.de)	Zertifikat wird an den Benutzernamen des Benutzers ausgestellt. (z.B. CN = alice)
Wird für Ver- sowie Entschlüsselung benutzt	Wird **nicht** für Ver- sowie Entschlüsselung benutzt

Tabelle 5.7: Vergleich Server-Zertifikat mit Client-Zertifikat

5.2.1 Prinzip

Wie bereits in dem Kapitel der Grundlagen, in dem Bereich “Transport Layer Security”, dargestellt, benutzt der TLS-Handshake die Publik-Key-Infrastruktur um den Server anhand eines Zertifikates zu authentifizieren. Wie bereits erwähnt ist die Client-Authentifizierung optional. Wird diese Option genutzt, besteht für den Server die Möglichkeit wie folgt vorzugehen [Client-Auth (2010)]:

1. **Überprüfung ob die digitale Signatur des Benutzers mit dem öffentlichem Schlüssel, welches im Zertifikat vorhanden ist, verifiziert werden kann.**
   Die digitale Signatur wird von dem Benutzer mit dem Zertifikat an den Server gesendet. Der Server kann anhand dieser beiden Informationen zunächst prüfen, ob der Benutzer auch im Besitz des privaten Schlüssels ist. Wurde also die digitale Signatur mit dem im Zertifikat vorhanden öffentlichen Schlüssel verifiziert, ist dies der Beweis für den Besitzt des privaten Schlüssels. Jedoch ist hier noch nicht bewiesen, ob der Benutzer auch der vorgegebene Benutzer ist. Dafür müssen die Schritte 3 sowie 4 erfolgreich durchlaufen werden.

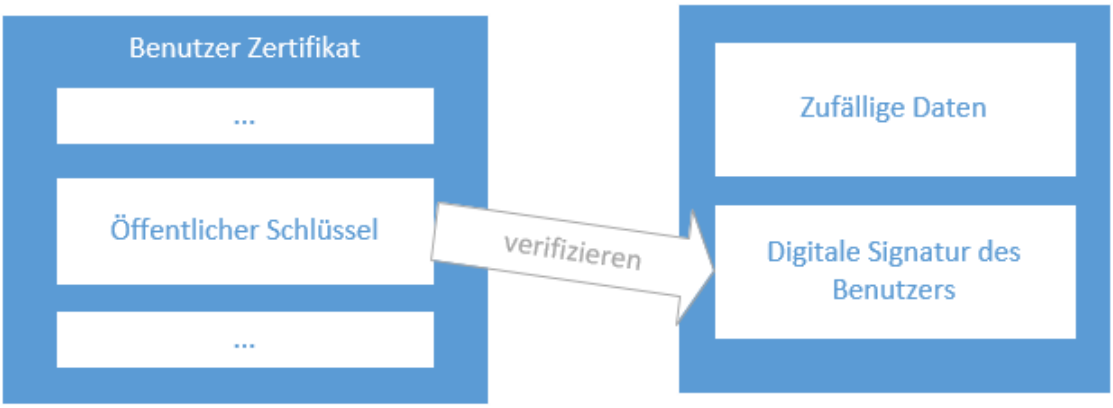

Abbildung 5.1: Client-Authentifizierung - Verifizierung der digitalen Signatur

2. **Wurde die digitale Signatur verifiziert, wird überprüft ob der aktuelle Tag in der im Zertifikat angegeben Zeitspanne liegt.**
   Hier wird die Gültigkeit des Zertifikats überprüft. Nur wenn das Zertifikat gültig ist, kann zum nächsten Schritt übergegangen werden.

3. **Liegt der aktuelle Tag in der angegebenen Zeitspanne, wird überprüft ob der Zertifizierungsstelle vertraut wird.**
   Für die Vertrauensüberprüfung besitzt der Server eine Liste an Zertifizierungsstellen, welchen vertraut wird. Ist die Zertifizierungsstelle in dieser Liste zu finden, kann der Server fortfahren.

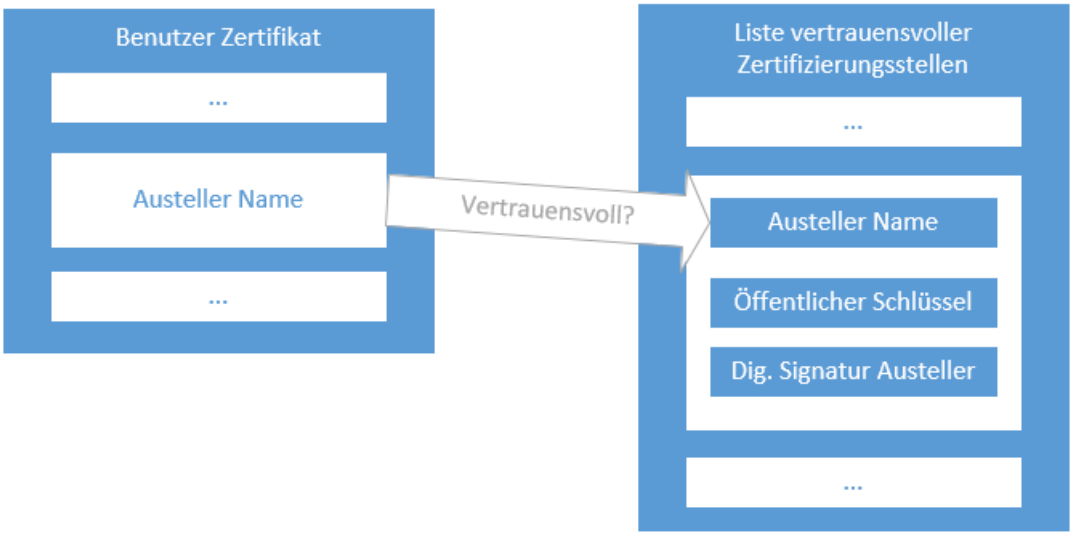

Abbildung 5.2: Client-Authentifizierung - Vertrauensprüfung der Zertifizierungsstelle

4. **Wird der Zertifizierungsstelle vertraut, wird überprüft, ob die digitale Signatur des Zertifikates mit dem öffentlichen Schlüssel der Zertifizierungsstelle verifiziert werden kann.**
   Hier wird die digitale Signatur des Zertifikates mit dem öffentlichen Schlüssel der Zertifizierungsstelle verifiziert. War die Verifizierung erfolgreich, ist der Beweis erbracht, dass der Benutzer auch der vorgegebene Benutzer ist. Außerdem ist durch Schritt 2. auch sichergestellt, dass der Benutzer im Besitz des privaten Schlüssels ist.

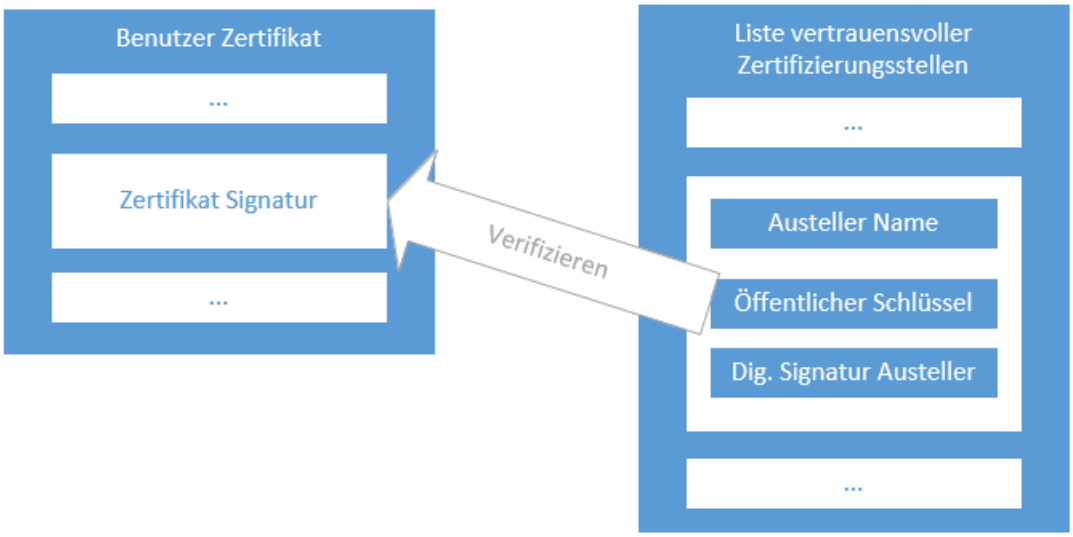

Abbildung 5.3: Client-Authentifizierung - Verifizierung der Zertifikatsignatur

5. **Konnte die digitale Signatur verifiziert werden, wird überprüft, ob der Benutzer Zugang zu der angefragten Stelle besitzt.**
   In diesem letzten Schritt wird Überprüft ob der Benutzer für die Anfrage autorisiert ist. Jedoch ist der Benutzer zu diesem Zeitpunkt bereits authentifiziert.

5.2.2 Vergleich

Die vorgestellte Client-Authentifizierung und die FIDO-Authentifizierung besitzen Gemeinsamkeiten. Aus diesem Grund werden im folgendem die Gemeinsamkeiten sowie die Unterschiede der PKI Client-Authentifizierung und der FIDO-Authentifizierung aufgezeigt, umso eine Differenzierung der beiden Verfahren zu ermöglichen.

5.2.2.1 Gemeinsamkeiten

Die Client-Authentifizierung sowie auch die FIDO-Authentifizierung verwenden das Konzept der Signierung, welches die asymmetrische Kryptographie mit sich bringt. In beiden Verfahren

werden Signierung anhand des privaten Schlüssels erstellt und zur Verifizierung der öffentliche Schlüssel benutzt. Die Verfahren prüfen, ob der Besitzer eines öffentlichen Schlüssels auch der vorgegebene Besitzer ist. In der FIDO-Authentifizierung wird der öffentliche Schlüssel des Authenticators durch den Metadata Service anhand des Attestierungszertifikates geprüft und bei der Client-Authentifizierung wird das Zertifikat des Benutzers anhand der Signierung im Zertifikat überprüft. Für diese Überprüfung wird eine dritte Partei benötigt, welche vollkommenes Vertrauen in der Authentifizierung besitzt. Dies kann bei der FIDO-Authentifizierung der jeweilige Hersteller vom Gerät sein und bei der Client-Authentifizierung eine externe Zertifizierungsstelle oder auch der Server selbst. Außerdem besteht bei beiden Verfahren eine Abhängigkeit. Bei der Client-Authentifizierung ist der Benutzer abhängig vom Zertifikat und bei der FIDO-Authentifizierung ist der Benutzer abhängig vom Authenticator, welche die Verwaltung des Schlüsselmaterials übernimmt. Besteht also kein Zugang zu dem Zertifikat bzw. Authenticator, ist keine Authentifizierung durch das jeweilige Verfahren möglich. In beiden Verfahren wird kein geteiltes Geheimnis benötigt, sodass für die Kommunikation das Sicherheitsrisiko reduziert wird. Zusammengefasst gilt für beide Verfahren folgendes:

- Beide Verfahren verwenden die Eigenschaften der asymmetrischen Kryptographie
- Es wird Überprüft ob der Besitzer des öffentlichen Schlüssels auch der vorgegebene Benutzer ist
- Es wird eine Vertrauensvolle Partei benötigt, welches den Besitz des öffentlichen Schlüssels legitimieren kann
- Kein geteiltes Geheimnis für die Authentifizierung benötigt

5.2.2.2 Unterschiede

Im Folgenden werden die Unterschiede der Client-Authentifizierung und der FIDO-Authentifizierung erklärt. Die FIDO-Authentifizierung benutzt den Metadata Service, um an den öffentlichen Schlüssel des Authenticators zu gelangen. Nach dem Erhalt des öffentlichen Schlüssels, kann der FIDO-Server diesen öffentlichen Schlüssel speichern und benötigt nicht bei jedem Authentifizierungsversuch den öffentlichen Schlüssel des Authentificators abzufragen. Besitzt der FIDO-Server den öffentlichen Schlüssel, so kann er mit diesem Schlüssel die Signierungen der öffentlichen Schlüssel, welche für die Authentifizierung benötigt werden, verifizieren. Der öffentliche Schlüssel des Authenticators wird benutzt, um die öffentlichen Schlüssel der Authentifizierung zu verifizieren. Wurde der öffentliche Schlüssel der Authentifizierung verifiziert, wird dieser für zukünftige Authentifizierungen benutzt. Bei der PKI Client-Authentifizierung wird bei jedem Authentifizierungsversuch der öffentliche Schlüssel, der Zertifizierungsstelle benötigt. Schaut man sich den Ablauf der vorgestellten PKI Client-Authentifizierung an, wird zunächst der öffentliche Schlüssel des Benutzers und anschließend das Zertifikat verifiziert. Auch wenn der öffentliche Schlüssel der Zertifizierungsstelle gespeichert wird, so wird diese jedes Mal benötigt.

PKI Client-Authentifizierung	FIDO-Authentifizierung
Benötigt für den Authentifizierungsversuch zwei öffentliche Schlüssel - 1. öffentlicher Schlüssel des Benutzers 2. öffentlicher Schlüssel der Zertifizierungsstelle	Benötigt den öffentlichen Schlüssel des Authenticators nur um öffentliche Schlüssel der Authentifizierung zu signieren. Zukünftige Authentifizierung wird mit dem öffentlichen Schlüssel der Authentifizierung vollzogen.

Tabelle 5.8: Vergleich Client-Authentifizierung mit FIDO-Authentifizierung - 1

Es besteht auch ein Unterschied bei der Verwaltung des Zertifikates bzw. des Schlüsselmaterials. Bei der FIDO-Authentifizierung ist der Authenticator für die komplette Verwaltung, Generierung sowie Signierung zuständig. Der Authenticator wird z.B. bei Mobiltelefonen bereits während der Herstellung integriert. Somit besitzt ein Benutzer schon mit dem Kauf eines Mobiltelefons eine Sicherheitskomponente, welche notwendige Aufgaben übernimmt. Es muss kein initialer Austausch zwischen dem Benutzer und dem Server stattfinden. Bei der PKI Client-Authentifizierung muss das Zertifikat ausgestellt und anschließend z.B. an einen Browser eingefügt werden. Dadurch entsteht für den Benutzer ein Aufwand.

PKI Client-Authentifizierung	FIDO-Authentifizierung
Benutzer muss technisch versiert sein oder der Server bietet ein Client-System, welcher das Zertifikat bereits beinhaltet. Es entsteht ein größerer Aufwand.	Der Authenticator übernimmt alle notwendigen Aufgaben. Benutzer wird nur nach lokaler Authentifizierung gefragt und wird mit anschließendem Vorgehen nicht konfrontiert.

Tabelle 5.9: Vergleich Client-Authentifizierung mit FIDO-Authentifizierung - 2

6 Entwicklung der FIDO-Allianz

Aufgrund der einfachen, schnellen und standardisierten Authentifizierung durch die FIDO-Spezifikationen, ist die FIDO-Allianz seit ihrer Gründung stetig am Wachsen. Im Folgenden wird die Entwicklung der FIDO-Allianz betrachtet, um einen Überblick über die Bekanntheit der FIDO-Allianz gewinnen zu können.

6.1 Mitglieder

Sechs Unternehmen waren an der Gründung der FIDO-Allianz beteiligt, inzwischen sind mehr als 250 Unternehmen Mitglied bei der FIDO-Allianz. [FIDO-FAQ] Bereits in den ersten acht Monaten, seit der Gründung der FIDO-Allianz, ist die Mitgliederanzahl auf über 50 Mitglieder angestiegen [FIDO-Alliance (2013)]. Da Board-Mitglieder eine besondere Stellung in der FIDO-Allianz besitzen, wird im Folgenden der Beitritt der Board-Mitglieder betrachtet.

6.1.1 Board-Mitglieder

Betrachtet man das Wachstum der Unternehmen, die sich in der Board-Mitgliedschaft befinden, sind seit anfänglich vier Unternehmen momentan 30 Unternehmen in der FIDO-Allianz als Board-Mitglied vertreten. Folgende Tabelle zeigt die Unternehmen, die eine Board-Mitgliedschaft besitzen sowie das jeweilige Datum der Bekanntmachung des Beitritts:

Unternehmen	Bekanntmachung	Unternehmen	Bekanntmachung
Lenovo	12.02.13	VISA	20.05.14
Synaptic	12.02.13	Qualcomm	13.09.14
NokNok	12.02.13	alibaba	23.09.14
PayPal	12.02.13	RSA	27.12.14
NXP	27.02.13	NTT Docomo	26.05.15
CrucialTec	17.04.13	Intel	02.06.15
Google	23.04.13	Ing	01.07.15
Yubico	01.05.13	USAA	01.07.15
MasterCard	10.10.13	Egis	14.07.15
Oberthur Technologies	15.10.13	American Express	14.10.15
Microsoft	12.12.13	Infineon	14.10.15
Bank of America	25.02.14	Vasco	14.10.15
ARM	22.04.14	aetna	23.02.16
Samsung	22.04.14	BC Card	08.03.16
Daon	06.05.14	Feitian	29.08.16

Tabelle 6.1: Liste der Board-Mitglieder

[Press Releases; Synaptic (2013); Google (2013); Yubico (2013); DAON (2014); Qualcomm (2014); DOCOMO (2015); Feitian (2016)]

Wie zu sehen ist, sind bereits im ersten Jahr sieben Unternehmen zu den vier Unternehmen hinzugekommen. Somit waren im ersten Jahr der Gründung der FIDO-Allianz bereits 11 Unternehmen an der Board-Mitgliedschaft beteiligt. Seit 2014 besitzen weitere acht und 2015 ebenfalls weitere acht Unternehmen die Board-Mitgliedschaft. Im Jahr 2016 sind lediglich drei Unternehmen in die Board-Mitgliedschaft beigetreten.

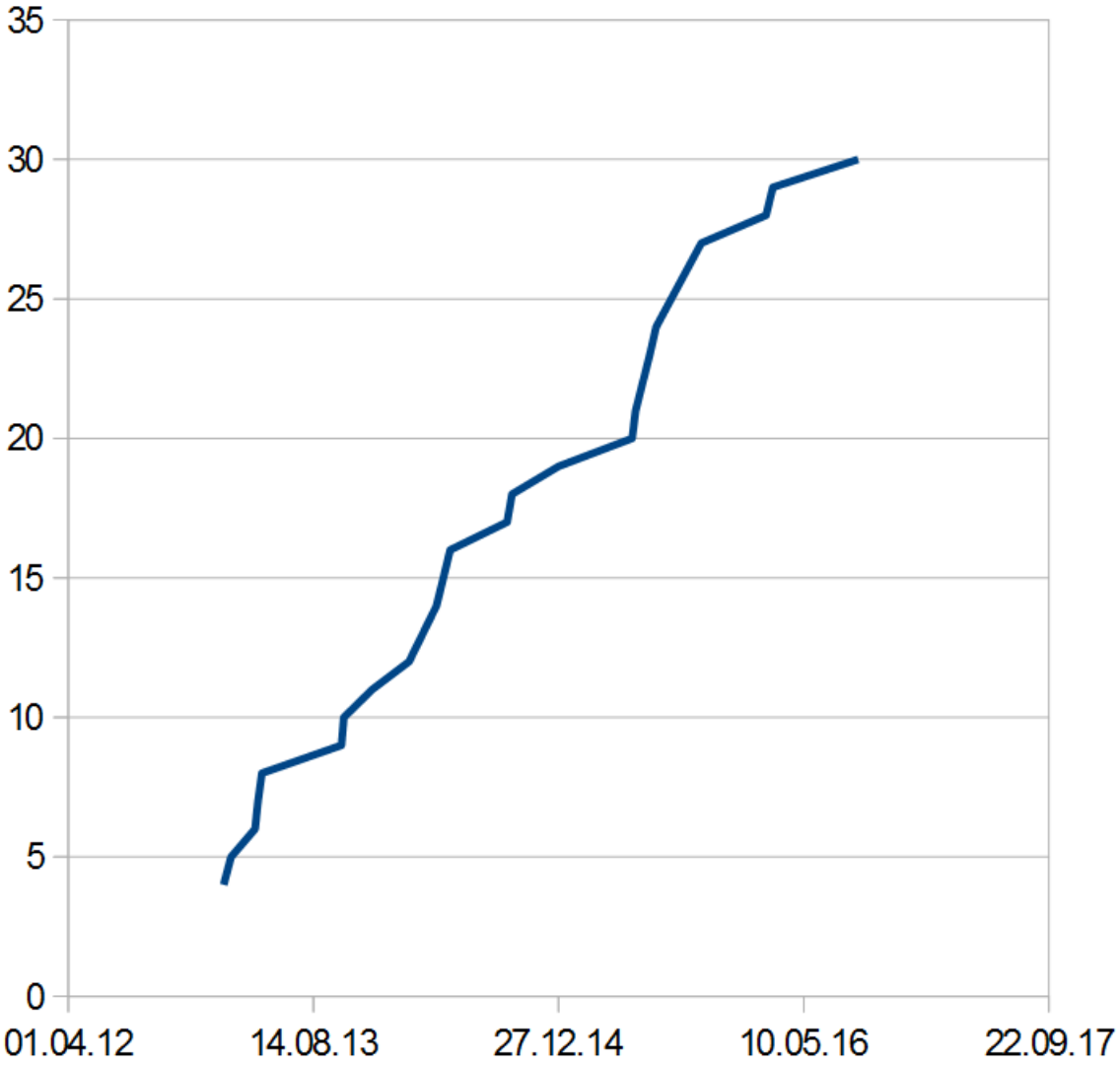

Abbildung 6.1: Beitritt der Board-Mitglieder

Betrachtet man die Bekanntmachungen anhand eines Diagramms, ist ab dem 21. Mitglied ein starker Anstieg zu beobachten. Die Mitglieder, welche 2015 beigetreten sind, sind innerhalb von ca. 6 Monaten beigetreten. Dies kann aufgrund der Veröffentlichung des “Certification Testing Program” zu Stande gekommen sein [FIDO-Cert (2015)]. Wie im Kapitel der Zertifizierung bereits erwähnt, ist ein zertifiziertes Produkt kompatibel zu anderen zertifizierten Produkten. Das “Certification Testing Program” wurde am 21.05.2015 veröffentlicht und die erste Mitgliedschaft in 2015 wurde am 26.05.2015 bekanntgegeben.

6.2 Zertifizierte Produkte

Neben dem Wachstum der Mitglieder besitzen die zertifizierten Produkte besondere Bedeutung, da durch das zertifizierte Produkt die Möglichkeit der Benutzung der FIDO-Authentifizierung besteht. Allein durch die Unterstützung der FIDO-Authentifizierung durch Facebook, erhalten 1,7 Milliarde Benutzer die Möglichkeit sich durch das FIDO-Authentifizierungsverfahren zu authentifizieren [FIDO-Available (2017)]. Folgende Liste zeigt die Anzahl der zertifizierten Produkte zu einem gegebenen Zeitpunkt an [FIDO-Prod (2017)].

Zeitpunkt	Anzahl zertifizierter Produkte
Mai 15	31
Jul 15	62
Sep 15	74
Dez 15	108
März 16	162
Mai 16	216
Aug 16	253
Jan 17	304

Tabelle 6.2: Anzahl zertifizierter Produkte zu einem bestimmten Zeitpunkt

Seit der Einführung der Zertifikation, am 21.05.2015 mit 31 zertifizierten Produkten [FIDO-Cert (2015)], ist die Anzahl der zertifizierten Produkte auf 304 angestiegen. Das folgende Diagramm soll die steigende Anzahl der zertifizierten Produkte veranschaulichen.

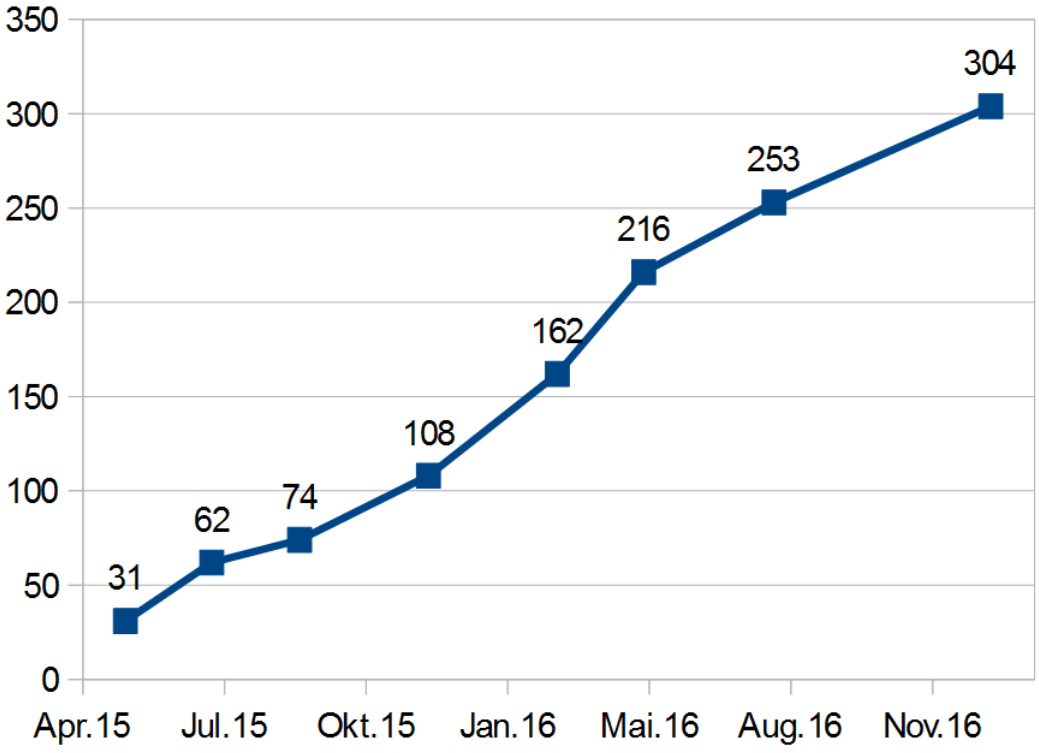

Abbildung 6.2: Zertifizierte Produkte

Wie man in dem Diagramm sieht, gab es in 2015 eine Steigerung von anfänglich 31 zertifizierten Produkten auf 108 zertifizierte Produkte. Somit wurden 108 Produkte bereits in 2015, im Jahr der Einführung der Zertifizierung, zertifiziert. Betrachtet man nun das Jahr 2016, ist eine Steigerung von ca. 200% [FIDO-Growth (2017)] zu sehen, also von 108 zertifizierten Produkten auf 304 zertifizierte Produkte.

6.3 Schlussfolgerung

Anhand der Daten ist zu sagen, dass die FIDO-Allianz besonders seit Einführung der Zertifizierung einen hohen Wachstum verzeichnen kann. Es ist zu vermuten, dass Unternehmen die Authentifizierung mittels Passwort mit einer einfachen und schnellen Authentifizierungsmethode ablösen möchten. Außerdem ist anzunehmen, dass die FIDO-Allianz mit den UAF und U2F Standards einen großen Schritt in Richtung passwortlose Authentifizierung gemacht haben und Unternehmen, wie beispielsweise Google, Microsoft und Samsung, an dieser Entwicklung teilhaben möchten.

7 Fazit und Ausblick

Die FIDO-Allianz nähert sich dem Ziel etablierte Authentifizierungsverfahren abzulösen. Durch die Ähnlichkeiten zu der Public-Key-Infrastruktur und das Verwenden des TLS-Protokolls, entsteht bei der Authentifizierung eine hohe Sicherheit. Durch das Verwenden des TLS-Protokolls, die Ähnlichkeiten zu der Public-Key-Infrastruktur und das Verwenden von Challenges werden Men-In-The-Middle und Replay Angriffe nur sehr schwer möglich. Die FIDO-Protokolle bieten auch für einzelne Transaktionen eine einfache und sichere Möglichkeit der Authentifizierung und stellen somit eine sichere Alternative zu dem TAN-Verfahren dar, welches sehr oft von Banken für Transaktionen eingesetzt wird. Benutzt ein Unternehmen bereits eine Authentifizierungsmethode und möchte diese nicht ablösen, so besteht auch die Möglichkeit diese Authentifizierungsmethode mit der FIDO-Authentifizierung zu erweitern. Da aktuell der Besitz eines Mobiltelefons üblich geworden ist und aufgrund der sicheren und einfachen Art der Authentifizierung, haben die FIDO-Standards das Potenzial etablierte Authentifizierungsmethoden mit der Zeit abzulösen. Somit gehe ich davon aus, dass Benutzer in zukunft diese Art der Authentifizierung bevorzugen werden, jedoch zurzeit im Allgemeinen die wissensbasierte Authentifizierungsmethode zur Gewohnheit geworden ist.

Besonders die Einführung der Zertifizierung scheint die Etablierung voranzubringen. Da ein Produkt durch die Zertifizierung parallel zur Sicherheit auch die Kompatibilität auszeichnet, macht die Zertifizierung diese Produkte besonders für Unternehmen interessant. Dadurch können Anwendungen mit neuen FIDO-Produkten erweitert werden, ohne bereits vorhandene FIDO-Produkte anzupassen. Somit ist möglich, dass Produkte unabhängig von Unternehmen zueinander Kompatibel sind. Die Notwendigkeit einer Datenbank, welche sensible Informationen zur Authentifizierung wie z.B. das Passwort speichern, ist durch die FIDO-Authentifizierung nicht mehr gegeben. Durch das standardisierte Verfahren muss bei der Entwicklung eines beispielsweise FIDO-Clients keine Rücksicht auf den bereits zertifizierten FIDO-Servers genommen werden. Aus den genannten Gründen geht für mich hervor, dass die Zertifizierung eines der größten Vorteile der FIDO-Authentifizierung darstellt. Durch die wachsende Anzahl an zertifizierten Produkten, wächst auch die Anzahl an Möglichkeiten. Beispielsweise besteht die Möglichkeit für Unternehmen, durch den Authenticator in Mobiltelefonen, FIDO-Clients zu entwickeln und selbst Teil der FIDO-Allianz zu werden. Außerdem denke ich, dass der Druck an Hersteller von Mobiltelefonen, die kein Mitglied der FIDO-Allianz sind, wächst.

Da die Authentifizierung mittels Benutzernamen und Passwort am weitesten verbreitet ist, besteht ein besonderer Bezug zu der FIDO-Authentifizierung. Die FIDO-Allianz besitzt viele Vorteile gegenüber der wissensbasierten Authentifizierung. In der wissensbasierten Authentifizierung wird dem Benutzer Verantwortung übertragen und der Benutzer wird, falls er eine

höhere Sicherheit haben möchte und somit für jedes Profil ein eigenes Passwort besitzt, mit einem hohen Aufwand konfrontiert. Diese Aspekte sind bei der FIDO-Authentifizierung nicht gegeben. In der FIDO-Authentifizierung wird kein Geheimnis übertragen oder gespeichert. Trotz der vielen Vorteile gehe ich davon aus, dass die wissensbasierte Authentifizierung zu sehr verbreitet und ein Ablösen in kürze nicht zu erwarten ist. Neben der weiten Verbreitung ist die Ungebundenheit der wissensbasierten Authentifizierung ein großer Vorteil. Auch wenn bereits viele Geräte einen Authenticator besitzen, ist für mich stark zu vermuten, dass die Gebundenheit eines Benutzers in der FIDO-Authentifizierung ein Ablösen erschweren wird.

Trotz der schweren Ablösung der wissensbasierten Authentifizierung wächst die FIDO-Allianz an. Waren es anfänglich lediglich sechs Unternehmen, besitzt die FIDO-Allianz aktuell mehr als 250 Mitglieder. Schaut man sich die Board-Mitglieder an, sind namenhafte Unternehmen, wie beispielsweise Google, Intel oder Samsung, an der Führung der FIDO-Allianz beteiligt. Bereits durch die Erweiterung von Facebook haben mehr als 1.7 Milliarden Menschen die Möglichkeit sich anhand der FIDO-Authentifizierung zu authentifizieren. Auch wenn nicht alle 1.7 Milliarden Menschen die FIDO-Authentifizierung nutzen werden, ist dies als ein großer Schritt für die Bekanntmachung der FIDO-Allianz anzusehen. Betrachtet man die Anzahl der zertifizierten FIDO-Produkte, sind seit Einführung der Zertifizierung mit 31 Produkten momentan über 300 Produkte zertifiziert. Ich bin der Meinung, dass Produkte von Herstellern, wie beispielsweise Samsung hier eine besondere Rolle spielen, weil durch die Zertifizierung eines Authenticators andere Unternehmen die Möglichkeit bekommen diesen Authenticator für andere FIDO-Produkte zu benutzen. Deshalb denke ich, dass die FIDO-Authentifizierung einen guten Weg verfolgt eine etablierte Authentifizierungsmethode zu werden.

Zusammengefasst gehe ich davon aus, dass nach folgenden Punkten ein Ablösen möglich ist. Die Mehrheit der Mobiltelefonhersteller müssen einen Authenticator in Mobiltelefone integrieren. Dadurch wird es für andere Unternehmen besonders Interessant die FIDO-Authentifizierung zu nutzen, weil der Besitz eines Mobiltelefon üblich geworden ist und viele Mobiltelefone einen biometrischen Scan durchführen können. Außerdem denk ich, dass die mit der FIDO-Authentifizierung verbunde Sicherheit mehr kommuniziert werden muss. Es sollten nicht nur Unternehmen und technisch versierte Benutzer den Unterschied der Sicherheit zwsichen der wissensbassierten- und FIDO-Authentifizierung verstehen können, sondern auch normale Benutzer. Ich denke, dass durch dieses Wissen und der einfachen Authentifizierung ein Ablösen in Zukunft möglich wird.

Literaturverzeichnis

[Approach-Vision] FIDO-Alliance: *Approach & Vision.* – URL https://fidoalliance.org/approach-vision/. – 23.10.2016 3, 3.2

[Certification] FIDO-Alliance: *Certification Overview.* – URL https://fidoalliance.org/certification/. – 06.12.2016 3.1, 4

[Certification-Submission] FIDO-Alliance: *Certification Submission.* – URL https://fidoalliance.org/certification/certification-submission/. – 09.12.2016 4.3

[Client-Auth 2010] Corporation, Oracle: *Client Authentication During SSL Handshake.* 2010. – URL http://docs.oracle.com/cd/E19424-01/820-4811/aakhe/index.html. – 17.01.2017 5.2.1

[Client-Server-Certificates 2012] Panday, Kaushal K.: *Client Certificates V/s Server Certificates.* 2012. – URL https://blogs.msdn.microsoft.com/kaushal/2012/02/17/client-certificates-vs-server-certificates/. – 19.01.2017 5.2

[Conformance] FIDO-Alliance: *Conformance Self Validation Testing.* – URL https://fidoalliance.org/certification/conformance/. – 09.12.2016 4.1

[DAON 2014] FIDO-Alliance: *IdentityX Joins FIDO Alliance Board.* März 2014. – URL https://www.daon.com/identityx-joins-fido-alliance-board/. – 25.01.2017 6.1.1

[DOCOMO 2015] DOCOMO: *DOCOMO Joins FIDO Alliance Board of Directors.* Mai 2015. – URL https://www.nttdocomo.co.jp/english/info/media_center/pr/2015/0526_00.html. – 25.01.2017 6.1.1

[Evoulution-of-Authentication 2010] Lindemann, Dr. R.: The Evolution of Authentication, 2010 5.1

[Feitian 2016] Feitian: *Feitian Joins FIDO Alliance Board of Directors.* August 2016. – URL https://www.ftsafe.com/article/530.html. – 25.01.2017 6.1.1

[FIDO-Alliance 2013] FIDO-Alliance: *FIDO Alliance Exceeds 50 Members in Eight Months. Industry Drives Fast toward Open Standards for Universal Strong Authentication.* Oktober 2013. – URL https://fidoalliance.org/fido-alliance-exceeds-50-members-in-eight-months-industry-drives-fast-toward-open-standards-for-universal-strong-authentication/. – 26.01.2017 6.1

[FIDO-Available 2017] FIDO-Alliance: *Facebook Makes FIDO Authentication Available to 1.7 Billion Users.* Januar 2017. – URL https://fidoalliance.org/facebook-makes-fido-authentication-available-to-1-7-billion-users/. – 30.01.2017 6.2

[FIDO-Cert 2015] FIDO-Alliance: *FIDO Alliance Unveils Certification Testing Program for 1.0 Specification and Introduces 31 FIDO Certified products.* Mai 2015. – URL https://fidoalliance.org/fido-alliance-unveils-certification-testing-program-and-introduces-fido-certified-products/. – 30.01.2017 6.1.1, 6.2

[FIDO-FAQ] FIDO-Alliance: *FAQ.* – URL https://fidoalliance.org/faqs/. – 16.02.2017 6.1

[FIDO-Growth 2017] FIDO-Alliance: *FIDO Certified Products Growth of 200Accelerating Global Support for FIDO Authentication.* Januar 2017. – URL https://fidoalliance.org/fido-certified-products-2016-growth/. – 15.02.2016 6.2

[FIDO-Prod 2017] FIDO-Alliance: *Strong Authentication Trends in Government.* Februar 2017. – URL https://www.slideshare.net/FIDOAlliance/strong-authentication-trends-in-government. – 27.02.2017 6.2

[Gigovic 2014] Gigovic, Boris: Fundamentals of the PKI Infrastructure, 2014 2.1.1

[Google 2013] Santos, Alexis: *Google joins the FIDO Alliance, supports its two-factor authentication standard.* April 2013. – URL https://www.engadget.com/2013/04/24/google-joins-fido-alliance-board-support-oepn-two-factor-authentication-standard/. – 16.02.2017 6.1.1

[History] FIDO-Alliance: *History of FIDO Alliance.* – URL https://fidoalliance.org/about/history/. – 31.10.2016 3, 3.1

[Interoperability] FIDO-Alliance: *Interoperability Testing.* – URL https://fidoalliance.org/certification/interoperability-testing/. – 10.12.2016 4.2

[Mark-Usage] FIDO-Alliance: *Certification Mark Usage.* – URL https://fidoalliance.org/certification/mark-usage/. – 10.12.2016 4.4

[Membership] FIDO-Alliance: *Member Benefits & Dues Levels.* – URL https://fidoalliance.org/participate/. – 08.11.2016 3.1

[Metadata-Service 2014] Lindemann, Rolf ; Hill, Brad ; Baghdasaryan, Davit: *FIDO UAF Authenticator Metadata Service v1.0.* Dezember 2014 3.2.1.1

[OTP 2016] Lindell, Andrew: Time versus Event Based One-Time Passwords, 2016 5.1.3

[PKI] Ultra-Electronics: Public Key Infrastructure, URL https://www.ultra-aep.com/uploads/aep/new/pdf/pkiwhitepaper.pdf. – 10.11.2016 2.1.2, 2.1.3, 2.2

[PKI-Cert-Chain] Symantec: *How certificate chains work.* – URL https://knowledge.symantec.com/support/ssl-certificates-support/index?page=content&actp=CROSSLINK&id=SO16297. – 01.04.2017 2.2.1

[Press Releases] FIDO-Alliance: *Press Releases.* – URL https://fidoalliance.org/category/news-events/press-releases/. – 16.02.2017 6.1.1

[Qualcomm 2014] Sammons, Brent: *Qualcomm announces kill switch security solution, involvement in FIDO.* September 2014. – URL https://www.qualcomm.com/news/onq/2014/09/12/qualcomm-announces-kill-switch-security-solution-involvement-fido. – 16.02.2017 6.1.1

[RFC2289 1998] Haller, N. ; Metz, C. ; Nesser, P. ; Straw, M.: A One-Time Password System / IETF. (RFC2289) February, 1998 5.1.3

[RFC5246 2008] Dierks, T. ; Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, August 2008 2.3

[Security-Advice 2009] Herley, Cormac: So Long, And No Thanks for the Externalities: The Rational Rejection of Security Advice by Users, 2009 5.1.2

[Synaptic 2013] Synaptics: *Synaptics and Nok Nok Labs work together to enable the Future of Mobile Authentication.* Februar 2013. – URL http://www.synaptics.com/company/news/future-mobile-authentication-noknok. – 16.02.2017 6.1.1

[U2F-Overview 2016] Srinivas, Sampath ; Balfanz, Dirk ; Tiffany, Eric ; Czeskis, Alexei: *Universal 2nd Factor (U2F) Overview.* September 2016 3.3, 3.3.1, 3.3.2

[U2F-YUBICO-OVERVIEW] YUBICO: *U2F Technical Overview.* – URL https://developers.yubico.com/U2F/Protocol_details/Overview.html. – 01.04.2017 3.3.2

[UAF-Overview 2014] Machani, Salah ; Philpott, Rob ; Srinivas, Sampath ; Kemp, John ; Hodges, Jeff: *FIDO UAF Architectural Overview.* Dezember 2014 3.2, 3.2.1, 3.2.4

[UAF-Specification 2014] Lindemann, Dr. R. ; Baghdasaryan, Davit ; Tiffany, Eric ; Balfanz, Dirk ; Hill, Brad ; Hodges, Jeff: FIDO UAF Protocol Specification v1.0. Dezember 2014. – Forschungsbericht 3.2, 3.2.2, 3.2.3

[UAF-Values 2013] Lindemann, Dr. R. ; Baghdasaryan, Davit ; Hill, Brad: *FIDO UAF Registry of Predefined Values.* 2013 5.1.2

[Yubico 2013] Ehrensvard, Stina: *Yubico joins FIDO Alliance.* Mai 2013. – URL https://www.yubico.com/2013/05/google-yubico-join-fido/. – 25.01.2017 6.1.1
Figureslot

Glossar

Adapter	Ein Adapter wird als eine Übersetzung einer Schnittstelle in eine andere verstanden. Inkompatible Schnittstellen können so miteinander kommunizieren.
API	Unter einer API (Application Programming Interface) versteht man eine Schnittstelle einer Anwendung, welches anderen Anwendungen zur Anbindung zur Verfügung gestellt wird.
ASCII-Codierung	Ein ASCII-Code wird verwendet um einen Text durch Zeichenkodierung bei Computern anzeigen zu lassen.
Challenge	Eine Challenge (Herausforderung) wird einem Kommunikationspartner gestellt der diese lösen muss. Falls diese Challenge verschlüsselt übertragen wurde, kann der Sender der Challenge prüfen ob der Empfänger die verwenden Schlüssel kennt. Es wird versucht Replay-Angriffe anhand von Challenges nicht zu ermöglichen.
Diffie-Hellman Schlüsselvereinbarung	Die Diffie-Hellman Schlüsselvereinbarung ermöglicht es symmetrische Schlüssel in einer abhörbaren und öffenen Leitung zu vereinbaren.
FatalError	Ein Fatal Error signalisert, dass eine dem System unbekannte Situation aufgetreten ist. Ein Beispiel einer solchen Situation ist das abweichen eines Protokolls.
Hashfunktion	Eine Hashfunktion bildet die Eingabemenge (z.B. ein Passwort) auf eine Zielmenge ab. Das bedeutet, dass beispielsweise das Passwort - password - durch die Eingabe in die SHA-1 Hashfunktion den Wert - 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8 - bildet. Außer dem SHA-1 Algorithmus existieren noch andere Hashfunktionen.
Proxy	Ein Proxy ist eine Schnittstelle in einem Netzwerk und kann Verbindung herstellen und Anfragen entgegen nehmen.

Reply Angriff	Unter einem Replay-Angriff versteht man einen Angriff, welcher Nachrichten abhört und diese Nachricht für spätere Anfragen erfolgreich nutzt. Eine Verschlüsselung von Nachrichten hilft bei dieser Art von Angriff nicht. Um vor solchen Angriffen zu schützen müssen Challenges verwendet werden, damit sich Nachrichten unterscheiden.
RSA-Algorithmus	Ein Algorithmus zum generieren von Schlüsselpaaren, welche für die asymmetrische Kryptographie verwendet werden.
Salt	Unter Salt versteht man eine zufällig gewählte Zeihenfolge. Diese Zeihenfolge wird an das Passwort angehängt und anschließend ein Hashwert gebildet.
symmetrischen Kryptosystem	In einem symmetrischen Kryptosysem wird anhand eines Algortihmus ein Schlüssel generiert. Diesen einen Schlüssel verwenden Kommunikationspartner für das ent- sowie verschlüsseln von Nachrichten.
User-Agent	Der User-Agent ist eine Schnittstelle, welcher Inhalte darstellt und Befehle des Benutzers entgegennimmt.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 4. April 2017 Murat Korkmaz