Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

Rücksetzen eines U2F-Accounts mit dem nPA

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science [B. Sc.]

eingereicht von: Samra Khan
geboren am: 12.12.1991
geboren in: Berlin

Gutachter/innen: Prof. Dr. rer. nat. Jens-Peter Redlich
Prof. Dr. Johannes Köbler

eingereicht am: .....

Zusammenfassung

Das Universal 2nd Factor Protokoll ist eine von Google gestartete Initiative. Zurzeit arbeitet die Fast IDentity Online (FIDO) Alliance (eine 2012 gegründete non-profit-Organisation) an dem Protokoll und Google ist ein Bestandteil dieser Gruppe. Der Gedanke hinter dem Protokoll ist, dass Internetnutzer Geräte (z.B. USB-Sticks) auf Webseiten registrieren können und zum Beispiel mit einem Passwort eine Zwei-Faktor-Authentifizierung (ein starker Identitätsnachweis, der aus zwei Komponenten besteht) ermöglicht wird.

Im Universal 2nd Factor Protokoll wird das Wissen des Nutzers (ein vierstelliges Passwort bestehend aus Ziffern) mit dem Besitz (eines USB-Sticks) für eine Zwei-Faktor-Authentifizierung kombiniert. Wenn der Nutzer einen Stick auf eine Webseite registriert, wird ein Schlüsselpaar für diese Webseite und diesen Account erzeugt. Der private Schlüssel kann nur von dem U2F-Stick benutzt werden[1] und das Schlüsselpaar wird später zur Authentisierung des Nutzers verwendet. Ohne das Vorhandensein des Sticks kann der Nutzer also keine Zwei-Faktor-Authentifizierung, für die der Stick verlangt wird, durchführen.

Diese Arbeit thematisiert den Fall, dass der Nutzer seinen Stick verliert oder einen Zweiten an seinen Account binden möchte. In dem Fall soll der Nutzer die Möglichkeit haben einen neuen Stick möglichst bequem auf allen Webseiten zu registrieren, auf die der alte Stick registriert war. Eine Lösung ist, dass der Nutzer über seinen neuen Personalausweis, durch einen eID-Dienstanbieter[2], authentifiziert wird, wodurch weiterhin seine Pseudonymität garantiert wird.

[1] Der private Schlüssel muss nicht zwangsläufig auf dem Stick gespeichert werden, er kann auch durch den U2F-Token verschlüsselt bei der Webseite vorhanden sein.

[2] Der eID-Dienstanbieter ist ein IT-Dienst, der in der Lage ist die eID-Funktionalität bezüglich rID des neuen Personalausweises zu nutzen.

Inhaltsverzeichnis

Abkürzungsverzeichnis

2FA Zwei-Faktor-Authentifizierung

CA Chip Authentification

eID Electronic Identity

FIDO Fast IDentity Online

MITM Man-In-The-Middle

nPA neuer Personalausweis

PACE Password Authenticated Connection Establishment

PUK Personal Unblocking Key

R Referenzwert

RFID radio-frequency Identification

rID Restricted Identification

TA Terminal Authentification

U2F Universal 2^{nd} Factor

UAF Universal Authentication Framework

Abbildungsverzeichnis

1 Einleitung

Ein Universal 2nd Factor (U2F)-Account ist ein Account bei einem Online-Dienst, der eine Authentisierungsform mit einem Passwort und einem speziell dafür hergestellten USB-Token (einem U2F-Token), den der Nutzer besitzt, anbietet. Der Nutzer authentisiert sich also mit der Kenntnis des Passwortes und der Anwesenheit des registrierten Sticks (nach dem U2F-Protokoll). Dies stellt eine Zwei-Faktor-Authentifizierung (2FA) dar. Der Stick wurde vorher für diesen Account von dem Nutzer registriert.

Dadurch wird eine sehr hohe Sicherheit erreicht. Wenn der Nutzer den Stick jedoch nicht bei sich hat, kann er sich nicht authentisieren. Der Nutzer kann den Stick verlieren, weshalb es sinnvoll ist eine sichere Methode anzubieten, einen neuen oder einen weiteren Stick an einen bereits existierenden Account zu registrieren. Die Daten des alten Sticks bei dem Online-Dienst können, wenn der Nutzer ihn verloren hat, gelöscht werden (das wird als Rücksetzen des Accounts bezeichnet). Ziel der Arbeit ist es, für die Registrierung des neuen Sticks den neuen Personalausweis (nPA) zu verwenden, da dieser eine sehr starke Authentisierungsform bietet, bei den meisten Nutzern in Deutschland ohnehin vorhanden ist und eine pseudonyme Verwendung ermöglicht.

Zuerst wird der Begriff 2FA erklärt, da sowohl das U2F-Prokoll, als auch der nPA Zwei-Faktor-Authentifizierungen ermöglichen. Dann wird der nPA thematisiert und ein günstiges und leicht umsetzbares Authentifizierungsprotokoll, welches den nPA verwendet, betrachtet. Anschließend wird das U2F-Protokoll erklärt und eine Lösung zum sicheren Rücksetzen des U2F-Accounts vorgestellt.

2 Zwei-Faktor-Authentifizierung

Eine Zwei-Faktor-Authentifizierung (2FA) ist eine starke Form der Authentifizierung. Dafür werden zwei der folgenden Methoden kombiniert. Der Nutzer authentisiert sich mit:

- seinem Wissen (Passwort, PIN, TAN usw.)
- seinem Besitz durch einen Hardwaretoken (Schlüssel, Karte usw.)
- körperlichen, untrennbaren Merkmalen (Fingerabdruck, Augen usw.)

Das IT-System, dessen Dienst der Nutzer verwenden will, verlangt zwei dieser Methoden. Nur wenn beide ausgeführt wurden und korrekt waren, ist der Nutzer authentisiert. Wenn bereits eine Methode fehlschlägt oder ein Nachweis fehlt, wird der Nutzer als nicht authentifiziert angesehen und ist nicht autorisiert den Dienst zu nutzen.

Eine Zwei-Faktor-Authentifizierung wird z.B. von Geldautomaten verwendet. Sie verlangen eine Kombination aus EC-Karte (Besitz) und einer PIN (Wissen), die der autorisierte Nutzer kennt. Die Karte kann nicht kopiert werden. Gibt man die PIN dreimal falsch ein, so wird die Karte gesperrt. Deshalb kann die PIN nicht durchprobiert werden. Durch diese Maßnahmen wird eine sehr starke Authentifizierungsform realisiert. Da die PIN nur aus vier Ziffern besteht, ist sie, im Vergleich zu herkömmlichen Passwörtern, leicht zu merken.

Die kombinierbaren Verfahren haben unterschiedliche Stärken und Schwächen. Entscheidet man sich für eine 2FA über einem Gegenstand, den der Nutzer besitzt, so muss der Token mitgeführt werden. Dadurch ist ein sehr hohes Sicherheitsniveau erreichbar, **aber wenn der Gegenstand gestohlen wird oder verloren geht, kann sich der Nutzer nicht authentisieren**. Außerdem ist eine Bindung an den berechtigten Besitzer nicht gegeben. Das heißt, ohne zusätzliche Maßnahmen kann jeder, der den Gegenstand hat, diesen benutzen. Weiterhin entstehen Kosten für die Produktion und Verwendung.

Wissensbasierte Verfahren sind ohne erhebliche Kosten umsetzbar. Jedoch können sich Menschen meist nur wenige und kurze Passwörter zuverlässig merken. Die Weitergabe des Wissens kann nicht ausgeschlossen werden. So kann

ein Angreifer z.B. während der Eingabe des Passworts zuschauen oder andere Methoden wie Social Engineering (zwischenmenschliche Beeinflussungen) verwenden, um das Passwort zu erhalten.

Im Universal 2nd Factor-Protokoll wird Wissen (ein Passwort) mit Besitz (z.B. einem USB-Token) kombiniert. Deshalb werden die Auswirkungen der genannten Nachteile (vor allem der Nachteile, die durch ein besitzbasiertes Verfahren entstehen) auf dem Protokoll besprochen und die Nachteile einer 2FA mit biometrischen Merkmalen sind nicht Bestandteil der Arbeit.

3 Der neue Personalausweis

Abbildung 1: nPA Beispielausweis. Quelle:`http://www.personalausweisportal.de/SharedDocs/Bilder/DE/Ausweis_stehend.jpg?__blob=poster&v=6` (abgerufen am 04.11.2014).

Ende 2010 wurde der neue Personalausweis (nPA) in Deutschland eingeführt (siehe Abbildung 1). Er besitzt einen radio-frequency Identification (RFID)-Chip, der verschiedene Formen der elektronischen Authentisierung ermöglicht. Neben der hoheitlichen Nutzung über die ePass-Funktion (nur durch Behörden), ist auch nicht-hoheitliche Verwendung (mit der eID-Funktion) zur elektronischen Authentisierung gegenüber Dritten möglich. Die nicht-hoheitliche Nutzung ist optional und kann abgeschaltet werden. Zusätzlich bietet der nPA dem Nutzer die Möglichkeit eine qualifizierte, elektronische Signatur zu erstellen (mit der

eSign-Funktion). Auch diese Funktion ist optional.

Das Protokoll Password Authenticated Connection Establishment (PACE) stellt sicher, dass Informationen aus dem nPA, nur bei Kenntnis der PIN, kontaktlos ausgelesen werden können. Nachdem die PIN zweimal falsch eingegeben wurde, wird die CAN verlangt. Wird die CAN richtig eingegeben, kann noch einmal die PIN eingegeben werden. Ist sie falsch, wird der Personal Unblocking Key (PUK) verlangt. So kann die PIN nicht ausprobiert werden. Der PUK selbst kann nur zehnmal verwendet werden.

Im nPA werden folgende Datengruppen gespeichert:

- Familienname
- ggf. Doktorgrad
- ggf. Ordens- bzw. Künstlername
- Vornamen
- ggf. Geburtsname
- Geburtsdatum
- Geburtsort
- Anschrift mit Postleitzahl
- biometrisches Lichtbild
- Seriennummer
- optional zwei Fingerabdrücke

Die Unterschrift wird nicht im Chip gespeichert.

3.1 Die eID-Funktion

In unserem Fall (für das Rücksetzen eines Sticks mit dem nPA) ist die nicht-hoheitliche Verwendung des nPAs interessant. Deshalb wird im Folgenden die Electronic Identity (eID)-Funktion erklärt.

Mit dieser Funktion können sich Nutzer im Internet gegenüber Dritten (ähnlich wie mit dem Personalausweis außerhalb des Internets) eindeutig und authentisch ausweisen. Möchte ein IT-System (zukünftig als Dienstanbieter bezeichnet)

Informationen aus dem nPA des Kunden auslesen, so muss dieses sich zunächst authentisieren und nachweisen, dass es berechtigt ist, diese Datengruppen zu sehen. Das geschieht in der Terminal Authentification (TA). Dazu besitzt der Dienstanbieter ein elektronisches Berechtigungszertifikat, das vom Bundesverwaltungsamt (von der Vergabestelle für Berechtigungszertifikate) ausgestellt wird. Darin sind die Datengruppen und Funktionen definiert, die dieser Dienstanbieter auslesen oder nutzen darf. Um ein Berechtigungszertifikat zu erhalten, muss der Dienst nachweisen, dass das Auslesen dieser Daten aus dem nPA für diesen Geschäftszweck notwendig ist. Die vorgelegten Zertifikate (der Dienstanbieter übergibt dem nPA des Nutzers seine Zertifikate) werden von dem nPA kryptografisch geprüft. Wenn diese Prüfung fehlschlägt, werden die Daten nicht übermittelt.

Dem Nutzer wird (in der Ausweisapp oder auf einem geeigneten Lesegerät) angezeigt, welcher Dienstanbieter welche Daten verwenden möchte. Bei Bedarf kann der Nutzer Einschränkungen vornehmen. Anschließend bestätigt der Nutzer mit der Eingabe seiner geheimen eID-PIN, dass diese Daten vom Dienstanbieter gelesen werden dürfen. Nach der Authentifizierung des Dienstanbieters (TA), muss sich der nPA auch gegenüber dem Dienstanbieter als authentisch beweisen (CA). Erst danach werden ausschließlich die Daten übertragen, die durch die Nutzereinschränkung und TA eingeschränkt wurden.

Da der Besitz des nPAs und die dazugehörige PIN verlangt werden, stellt das Szenario eine starke 2FA dar. Nur mit einem gültigen Berechtigungszertifikat und dem dazugehörigen privaten Schlüssel ist es möglich, Daten aus dem nPA zu erhalten. Ansonsten wird der Benutzer nicht erkannt, da erst nachdem sich der Dienstanbieter authentisiert hat, Daten von dem nPA übermittelt werden. Weil auch der nPA eine Authentisierung durchführt, kann der Dienstanbieter sicher sein, dass er mit einem gültigen Personalausweis spricht und die gesendeten Daten authentisch sind. Zwischen dem Lesegerät und dem nPA wird ein PACE-Kanal benutzt. Anschließend wird zwischen Dienstanbieter und dem nPA ein sicherer Kanal (Secure Messaging) aufgebaut.

Für die Kommunikation zwischen dem Dienstanbieter und dem nPA des Kunden wird ein eID-Server benötigt. Dieser verwaltet private Schlüssel, die zur

Authentisierung gegenüber dem nPA verwendet werden. Auch dafür werden sichere Kommunikationskanäle aufgebaut.

Folgende Daten kann der Nutzer freigeben:

- Vor- und Familienname, ggf. Ordens- und Künstlername oder Doktorgrad
- Ausgebender Staat: D für Bundesrepublik Deutschland
- Geburtstag und Geburtsort
- Anschrift
- Dokumententyp[3]
- **Angabe, ob der eigene Wohnort einem abgefragten Wohnort entspricht (Wohnortbestätigung)**
- **Angaben zur Über- oder Unterschreitung eines bestimmten Alters (Altersbestätigung)**
- **Pseudonyme Kennung**

3.2 Pseudonyme Nutzung des nPAs

Der nPA besitzt eine Pseudonymfunktion (z.B. für pseudonymes Login). Dadurch kann der Ausweisinhaber bei erneuter Dienstnutzung eines Accounts innerhalb eines Sektors wiedererkannt werden, ohne personenbezogene Daten preiszugeben. Der Dienst kann den Nutzer nicht identifizieren, aber der Nutzer kann beweisen, dass er zweifelsfrei der Besitzer des Accounts ist.

Für jeden Dienstanbieter generiert der nPA ein eigenes Pseudonym. Damit wird verhindert, dass zwei verschiedene Dienstanbieter die selbe Person anhand der Pseudonyme wiedererkennen und so Informationen (durch die Accounts) zusammenführen können. Da ein berechnetes Pseudonym nur für den jeweiligen Dienst gültig ist, wird es Restricted Identification (rID) genannt. Die rID ist an den nPA gebunden. Wird dem gleichen Nutzer ein neuer Personalausweis ausgestellt, so muss ein neues Pseudonym berechnet werden. Die rID ist also für ein Paar (nPA, Dienst) gültig und konstant.

[3] "Der Dokumententyp ist "ID" beim Personalausweis, "AR", "AS" oder "AF" beim Aufenthaltstitel." Quelle: Technischerichtlinien TR03127 BSI-TR-03127.

Genauer genommen ist die rID für ein Paar (nPA, Sektor) gültig. Die Dienstanbieter werden in Terminalsektoren eingeteilt. Ihre Berechtigungszertifikate enthalten Sektorschlüssel, die bei der Berechnung der rID mitwirken.

Wenn ein Ausweis gesperrt wird, werden für die Sektoren individuelle Sperrlisten erzeugt. Diese Sperrlisten werden bei der Nutzung der eID-Funktion mit dem Sperrmerkmal (das zwingend vom nPA übermittelt wird) abgeglichen. Das Sperrmerkmal ist nur für diesen Abgleich vorhanden, und darf nicht von Dienstanbietern gespeichert werden. Wenn gegen diese Policy verstoßen wird, kann die anonyme Nutzung des nPAs in eine pseudonyme Nutzung überführt werden.

Für unseren Anwendungsfall, das Rücksetzen eines U2F-Accounts, wird die rID gebraucht, da Pseudonymität bei der Authentisierung garantiert werden soll.

3.3 Anonyme Nutzung des nPAs

Über den nPA können Alters- und Wohnortabfragen durchgeführt werden, ohne das gespeicherte Geburtsdatum oder die Adresse zu übermitteln. Ein Dienst oder ein Automat kann beispielsweise feststellen, ob der Kartenbesitzer über 18 Jahre alt ist, ohne Hinweise auf dessen Identität zu erfahren. Ein Nutzer kann dabei nicht wiedererkannt werden.

4 Auth^2(nPA)

Müller, Redlich und Jeschke haben das Konzept einer sicheren und günstigen 2FA durch den nPA vorgestellt, das von mehreren Einrichtungen gemeinsam genutzt werden kann.[4]

Mit der eID-Funktion des nPAs hat jeder Bürger die Möglichkeit, einen Token zur Unterstützung komfortabler (gespeicherte Daten müssen nicht eingetippt werden), transparenter (alle übermittelten Daten werden dem Nutzer angezeigt), starker 2FA zu besitzen. Dazu werden vom Kartenbesitzer nur ein Lesegerät und

[4] Das Kapitel richtet sich nach der Quelle: Müller, Redlich, Jeschke (2011). Auth^2(nPA) Starke Authentifizierung mit nPA für jedermann. SP Gabler Verlag.

eine Endanwendersoftware (z.B. AusweisApp 2[5]) benötigt (die eID-Funktion muss gegebenenfalls aktiviert werden). Für verlorene oder gesperrte nPAs gibt es bereits Sperrmanagement-Vorkehrungen. Die weite Verbreitung des nPAs und der geringe Aufwand für den Nutzer (Kaufen eines Lesegerätes und Installieren der AusweisApp 2) sind sehr gute Voraussetzungen für eine einheitliche elektronische Authentifizierungsstruktur in Deutschland.

Für die 2FA muss der Dienstanbieter einen eID-Server selbst bereitstellen oder nutzen. Der Betrieb des eID-Services bildet die wesentlichen Kosten für die Authentifizierung. **Auth²(nPA) ist ein Konzept, welches einen einzigen eID-Service für mehrere Einrichtungen benutzt. Dadurch können die Kosten reduziert werden.** Im Folgenden wird das Konzept erläutert.

Die beteiligten Komponenten für eine 2FA im Auth²(nPA)-Konzept sind:

- Eine Einrichtung (E), die gewisse IT-Dienste erbringt. Sie verwaltet ihre Nutzer eigenständig und bietet die Möglichkeit einer Authentifizierung mit dem nPA an.
- Ein Nutzer (N), der den Dienst nutzt oder nutzen möchte. Er kann Accounts an den Einrichtungen haben und eröffnen. Der Nutzer besitzt einen nPA mit freigeschalteter eID-Funktion (ein Lesegerät, der AusweisApp 2 und einen Computer mit einem Browser).
- Ein Dienstanbieter (D), der nach Auth²(nPA) den eID-Service für mehrere Einrichtungen erbringt. Er besitzt ein Berechtigungszertifikat (ausgestellt vom Bundesverwaltungsamt) und den dazugehörigen privaten Schlüssel. Der Dienstanbieter ist in der Lage, die eID-Funktion (genauer die rID-Funktion) des nPAs zu nutzen.
- Ein anonymisierender Proxy (P), um Datenschutz zwischen der Einrichtung und dem Dienst zu garantieren.

4.1 Anforderungen

Das Zusammenführen der Informationen zu einem Nutzer soll nicht ohne dessen

[5] Seit dem 01.11.14 in Benutzung und auf der Webseite `https://www.ausweisapp.bund.de/ausweisapp2/` als Download erhältlich.

Mitwirkung möglich sein. Der Dienstanbieter soll nicht in der Lage sein zu erfahren, wie viele und welche Accounts (bei welcher Einrichtung) ein Nutzer hat. Pseudonymität des Nutzers bei dem Dienstanbieter (optional auch bei der Einrichtung), soll garantiert werden. Dafür wird das Wissen der einzelnen Komponenten (Einrichtung, Nutzer, Dienstanbieter) auf das Minimum reduziert, welches für eine sichere Authentisierung des Nutzers benötigt wird.

Weitere sicherheitsrelevante Anforderungen wie Vertraulichkeit und Authentizität der Kommunikation werden hier nicht besprochen und sollen mit Standardtechniken umgesetzt werden.

4.2 Lösungsidee

Der Nutzer möchte sich bei einer Einrichtung anmelden. Er gibt seinen Loginnamen für diese Einrichtung an. Der Nutzer muss beweisen, dass er die Person ist, die diesen Account besitzt. Bei der Einrichtung ist ein Referenzwert für den Account gespeichert. Wenn der Nutzer, zusammen mit der Einrichtung und dem Dienstanbieter, diesen Referenzwert (R) erneut berechnen kann, ist der Nachweis erbracht. Um R zu berechnen braucht man zwei Teilgeheimnisse, G_1[6] und G_2[7]. Das Geheimnis G_1 wird bei der Initialisierung eines Nutzers zufällig bei der Einrichtung erzeugt und für diesen Account gespeichert. Das zweite Geheimnis G_2 wird bei der Initialisierung eines Nutzers (durch die rID) bei dem Dienstanbieter zufällig erzeugt und für diese rID gespeichert. Durch die Verwendung eines zufälligen G_2 werden keine Daten des nPAs an Dritte weitergegeben und es wird die Möglichkeit bewahrt, auf einen anderen Ausweis mit anderer rID zu migrieren.

Für die Authentifizierung des Nutzers übergibt die Einrichtung ihr Teilgeheimnis G_1 verschlüsselt an den Nutzer. Der Nutzer reicht dieses Geheimnis zusammen mit der Authentifizierungsanfrage an den Dienstanbieter weiter. Der Nutzer authentisiert sich bei dem Dienstanbieter durch die rID und der Dienstanbieter kann dieser rID das zweite Teilgeheimnis G_2 zuordnen. Mit G_1 und G_2 kann der Dienstanbieter nun R berechnen und zum Abholen für die

[6] G_1 existiert für einen Account.
[7] G_2 existiert für eine rID.

Einrichtung verschlüsselt bereitstellen. Aus R darf kein Rückschluss auf die Teilgeheimnisse gezogen werden. Dafür nutzt man passende kryptografische Funktionen[8]. Nachdem die Einrichtung R abgeholt hat, vergleicht sie es mit ihrem für diesen Account abgespeicherten Referenzwert. Stimmen sie überein, so ist der Nutzer authentifiziert.

4.3 Kommunikation

Der Nachrichtenaustausch im Auth^2(nPA)-Protokoll wird folgendermaßen von Müller, Redlich und Jeschke beschrieben und in Abbildung 2 dargestellt:

1. „N initiiert starke Authentisierung und gibt login an E an.
2. Einrichtung generiert Anfrage an Auth^2(nPA)-Dienst. Darin sind G_1 für das entsprechende login und kryptografisches Material für vertrauliche und anonyme Abholung des durch D frisch berechneten Ergebnisses erhalten. Diese Anfrage ist authentisiert und für D verschlüsselt und wird an N zusammen mit einer Weiterleitung gesendet.
3. N leitet Anfrage an D weiter.
4. Nachweis der rID von N mit der eID-Funktion des nPAs gegenüber D.
5. Durch P anonymisierte Abfrage des von D frisch berechneten und signierten Ergebnis R‘.
6. Auslieferung des von D frisch berechneten und signierten Ergebnis R‘, verschlüsselt mit einem Schlüssel, den E in Schritt 2 gewählt hat.“[9]

4.3.1 Initialisierung des Nutzers bei einer Einrichtung

Entscheidet sich ein Nutzer erstmalig dazu, den Auth^2(nPA)-Dienst an einer Einrichtung für einen Account zu nutzen, so muss erst der Referenzwert für

[8] Hierfür wurde ein HMAC mit G_1 als Nachricht und G_2 als Schlüssel vorgeschlagen
[9] Müller, Redlich, Jeschke. (2011) Auth^2(nPA) Starke Authentifizierung mit nPA für jedermann

diesen Account berechnet und bei der Einrichtung abgelegt werden. G_1 wird bei der Einrichtung zufällig erzeugt und für den Account gespeichert. Dann erfolgen die Schritte 2-6 der Kommunikation wie in Abschnitt 4.3 beschrieben. Im 6. Schritt erhält die Einrichtung den Referenzwert R‘ und speichert diesen für den Nutzeraccount. Für spätere Authentifizierungen des Nutzers ist dies der R.

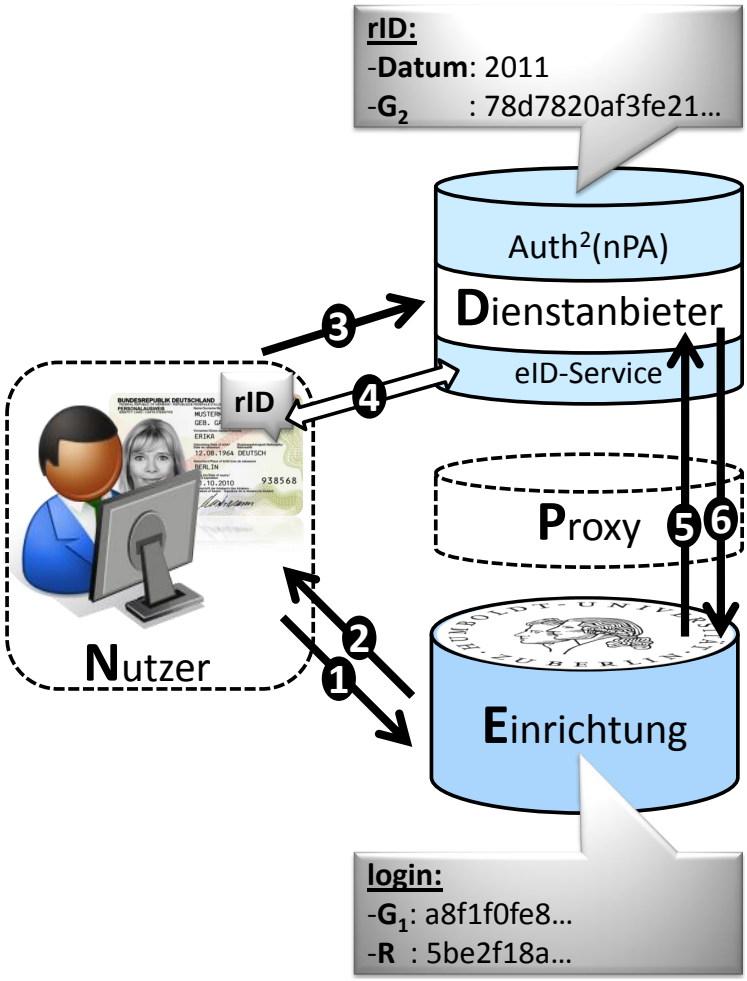

Abbildung 2: Kommunikation im Auth²(nPA)-Konzept. Quelle: Müller, Redlich, Jeschke (2011). Auth²(nPA) Starke Authentifizierung mit nPA für jedermann.

4.3.2 Authentisierung des Nutzers bei einer Einrichtung

Für die Nutzerauthentifizierung werden die Kommunikationsschritte 1 bis 6 durchlaufen. Für diesen Account existiert G_1 bereits bei der Einrichtung und für die passende rID des Nutzers existiert G_2 beim Dienstanbieter. Im

6. Schritt erhält die Einrichtung R' und vergleicht es anschließend mit seinen abgespeicherten Referenzwert für diesen Account. Sind beide identisch so ist der Nutzer authentifiziert.

4.4 Weitere Aspekte

Einrichtungen, also IT-Dienste, die den Auth^2(nPA)-Dienst nutzen wollen, müssen sich beim Dienstanbieter registrieren. Dort werden unter anderem Name, Anschrift und Public-Key bzw. Zertifikat hinterlegt.

Die Anfragen beim Dienstanbieter sollen nicht zu Einrichtungen zugeordnet werden können. Deshalb kann kein einrichtungsindividuelles Schlüsselmaterial (die Schlüssel werden verwendet um die Referenzwerte abzuholen) verwendet werden. Die Lösung ist eine gemeinsame Nutzung von symmetrischen Schlüsseln für alle Einrichtungen. In kurzen Zeitperioden werden neue Schlüssel generiert, die sich jede Einrichtung (mit ihrem Zugangskonto) abholen kann. Gesperrte Einrichtungen haben damit keinen Zugang zum Schlüsselmaterial.

Ist ein Nutzer nicht mehr bei einer Einrichtung angemeldet, sperrt die Einrichtung diesen Account. Das Löschen bei dem Dienstanbieter findet nicht statt, da dasselbe G_2 von anderen Einrichtungen genutzt werden kann. Wurde ein Ausweis gestohlen oder hat der Nutzer ihn verloren, so steht er auf der Sperrliste des Auth^2(nPA)-Dienstanbieters und kann dort nicht benutzt werden. Weiterhin löscht der Dienstanbieter alle Einträge, die älter als 10 Jahre sind, da dafür kein gültiger Ausweis existieren kann.

Für das Rücksetzen des U2F-Accounts soll in dieser Arbeit der Auth^2(nPA)-Dienst benutzt werden, da so eine sehr starke 2FA verwendet wird, die Pseudonymität erlaubt und möglichst günstig ist. In diesem Zusammenhang bietet der eigentliche IT-Dienst, dessen Dienst der Nutzer verwenden möchte, eine 2FA[10] durch den Auth^2(nPA)-Dienst an.

[10] Der Besitz des nPAs und das Wissen der dazugehörigen eID-PIN bilden eine 2FA.

5 Universal 2nd Factor

Das U2F Protocol ist ein Teil einer von Google gestarteten Initiative. Zurzeit arbeitet die FIDO Alliance (eine 2012 gegründete non-profit-Organisation) an diesem Protokoll.[11]

Diese hat es sich zur Aufgabe gemacht, die Natur von starker online Authentifizierung zu verändern[12]. Dafür wollen sie globale, offene und skalierbare Standards definieren, durch die die Abhängigkeit von Passwörtern reduziert wird. Da durch die wachsende Anzahl an Online-Dienste, der Nutzer immer mehr und immer komplexere[13] Passworte bräuchte. Die FIDO Alliance legt dabei Schwerpunkt auf Benutzerfreundlichkeit, Sicherheit und Datenschutz. Für diesen Zweck wird neben dem U2F-Protokoll auch das Universal Authentication Framework Protokoll entwickelt. Der Vollständigkeit wegen wird im folgenden Abschnitt das Universal Authentication Framework Protokoll erläutert.

5.1 Universal Authentication Framework Protokoll

Das Universal Authentication Framework (UAF) Protokoll (von FIDO entwickelt) ermöglicht es IT-Diensten (Einrichtungen) multi-Faktor Authentifizierungen ohne Passwörter anzubieten. Der Nutzer registriert ein Authentisierungsgerät bei der Einrichtung, indem er eine lokale Authentifizierungsmethode (die ihm zur Verfügung steht) auswählt (z.B. Fingerabdruck einscannen, in eine Kamera blicken, in ein Mikrofon sprechen, PIN eingeben). Die Einrichtung kann hierbei entscheiden, welche Methoden sie zulässt. Nachdem ein Gerät registriert wurde, kann der Nutzer zukünftig über ein wiederholtes Ausführen dieser Methode authentifiziert werden.

Der Mehrwert dabei ist, dass die Nutzer sich keine[14] Passwörter mehr merken müssen und dennoch sicher authentifiziert werden können. Das UAF-Protokoll bietet auch das Kombinieren mehrerer Authentifizierungsmethoden an, wie z.B. Fingerabdruck und PIN Eingabe.

[11] Das Kapitel richtet sich nach der Quelle: S. Srinivas, D. Balfaz, E. Tiffany (2014): FIDO alliance.Univrsal 2nd Factor (U2F) Overview.
[12] Quelle: `https://fidoalliance.org/about` [abgerufen am 30.11.14].
[13] Um Brute-Force, trotz steigender Rechenleistung, zu erschweren.
[14] PINs werden dabei nicht als Passwörter betrachtet.

5.2 Konzept

Das U2F-Protokoll erlaubt es IT-Diensten die Möglichkeit einer 2FA mit einem Passwort (wie gewohnt) und zusätzlich einem Stick anzubieten. Dadurch kann die Sicherheit beim Login gesteigert werden. Der Nutzer gibt wie gewohnt seinen Nutzernamen und das dazugehörige Passwort an. Die Einrichtung kann den Nutzer jederzeit dazu auffordern, die Anwesenheit des Sticks nachzuweisen. **Durch den zweiten Faktor (den Stick) kann das Passwort auf vier Ziffern reduziert werden, ohne, im Vergleich zu der alleinigen Verwendung eines Passworts, an Sicherheit zu verlieren.**[15]

Während der Registrierung oder Authentifizierung kann der Nutzer den Stick benutzen, indem er bequem auf einen Knopf (siehe Abbildung 3) auf dem USB-Stick drückt. Für verschiedene Dienste, die das Protokoll nutzen, kann ein einziger Stick verwendet werden.

Alternativ können Einrichtungen auch Authentifizierungen ausschließlich mit dem Stick zulassen.

5.3 U2F-Geräte

In dieser Arbeit werden USB-Sticks als Repräsentation für U2F-Geräte benutzt. In der Praxis können U2F-Geräte auf verschiede Arten (z.B. als ein eigenständiges NFC-fähiges Gerät, ein eigenständiges 'Bluetooth LE'-fähiges Gerät, reine Softwareimplementierung[16] auf dem Computer oder Smartphone des Nutzers oder eingebaute kryptografisch sichere Hardwarekomponenten) realisiert werden. Es wird empfohlen Hardwarekomponenten (eingebaut oder alleinstehend) zu benutzen, dies ist jedoch nicht zwingend erforderlich um das Protokoll zu realisieren. Abhängig von der Wahl des U2F-Geräts, wird die Kommunikation mit dem Gerät angepasst.

Zur Vereinfachung werden wir uns nicht mit gerätespezifischen Eigenschaften beschäftigen und davon ausgehen, dass USB-Token verwendet werden.

[15] Quelle: `https://fidoalliance.org/specifications` (abgerufen am 16.11.14).

[16] Reine Softwareimplementierungen bieten nur eine schwache Implementierung des U2F-Protokolls, da z.B. das Aktivieren des Tokens nicht die Anwesenheit des Nutzers voraussetzt und durch Malware simuliert werden kann.

Abbildung 3: Beispiel U2F-Stick Quelle:`https://www.yubico.com/press/images/` (abgerufen am 16.11.14).

5.4 Datenschutz und Sicherheit

Bei der Umsetzung des U2F-Protokolls und dem Entwurf der U2F-Geräte wurde sehr viel Wert auf Datenschutz und Datensicherheit gelegt.

Registriert sich ein Nutzer bei der Webseite einer Einrichtung, mit einem Stick, für einen Account, so erzeugt der Stick ein neues Schlüsselpaar und ein Key-Handle. Das Schlüsselpaar kann nur für diese Einrichtung verwendet werden. Die Einrichtung erhält den öffentlichen Schlüssel zusammen mit dem Key-Handle vom Stick und speichert beide für den benutzten Account ab. Der private Schlüssel kann nur von dem Stick verwendet werden.

Wenn der Nutzer eine Authentisierung durchführt, kann die Einrichtung prüfen, ob der Nutzer den Stick besitzt, indem sie eine Signatur einer Challenge vom Stick verlangt. Diese Signatur kann die Einrichtung mit dem passendem öffentlichem Schlüssel (der Schlüssel, der für diesen Account abgespeichert ist) verifizieren. Dazu sendet die Einrichtung den Key-Handle über den Browser an den Stick. Der Stick nutzt den Key-Handle, um den privaten Schlüssel zu

identifizieren[17], der zu dieser Einrichtung und diesem Account gehört, und erstellt die Signatur.

Da für jede Einrichtung ein eigenes Schlüsselpaar erzeugt wird, ist der Stick vergleichbar mit einem Schlüsselbund (ein privater Schlüssel repräsentiert dann einen physikalischen Schlüssel). Genau wie einen echten Schlüsselbund, kann der Nutzer den Stick verlieren oder zu Hause vergessen und er kann gestohlen werden. Für die Authentisierung muss der Nutzer Zugriff auf den Stick haben.

Damit ein Angreifer den Key-Handle nicht erraten kann, wird ein Hash zur Erkennung der berechtigten Einrichtung benutzt. Während der Registrierung mit einem Stick sendet der Browser dem Token einen Hash von dienstspezifischen Daten der Einrichtung (verwendetes Protokoll, Hostname, Port). Der Stick codiert diese Informationen in den Key-Handle. Wenn der Nutzer eine Authentisierung vornehmen möchte, sendet der die Einrichtung den Key-Handle zum Browser. Der Browser leitet den Key-Handle zusammen mit dem frisch[18] erstelltem Hash über die dienstspezifischen Daten der Einrichtung weiter. Der Stick prüft, ob der Key-Handle zu dem frischem Hash (basierend auf dem SSL-Zertifikat und den Namen der Einrichtung) gehört. Nur im positiven Fall, wird eine Signatur erstellt.

Dadurch wird (wenn der Browser nach dem Protokoll handelt) sichergestellt, dass ein öffentlicher Schlüssel und ein Key-Handle, die für eine Einrichtung erzeugt wurden, nicht von einer anderen Einrichtung oder einer anderen Webseite benutzt werden können. Ohne den Hash könnten ausgegebene öffentliche Schlüssel mit den dazugehörigen Key-Handles von nicht berechtigten Angreifern benutzt werden, um Nutzerinformationen zusammenzuführen.

5.4.1 Aktivierung des Sticks durch den Benutzer

Der Stick kann nur benutzt werden, wenn er explizit (zum Beispiel mit einem Knopfdruck) aktiviert wird. Damit soll die Anwesenheit des Nutzers sichergestellt werden. Hier ist zu empfehlen, dass die U2F-Komponente nicht als

[17] Diese Identifikation kann auf unterschiedliche Arten erfolgen. Wenn z.B. die privaten Schlüssel in einer Tabelle auf dem Stick gespeichert werden, kann ein Key-Handle als Index in der Tabelle benutzt werden.

[18] Der Browser berechnet erneut wie bei der Registrierung ein Hash.

reine Softwareimplementierung, die eine Tastatureingabe für die Aktivierung verlangt, umgesetzt werden sollte, da sonst Malware das Aktivieren vortäuschen kann.

Sowohl bei der Registrierung (um das Schlüsselpaar zu erstellen) als auch bei der Authentisierung (um die Signatur zu autorisieren), wird der Nutzer durch den Browser dazu aufgefordert den Stick zu aktivieren. Die Aufforderung durch den Browser (in Form einer Infobar) kann abgeschaltet werden, der Knopfdruck muss dennoch stattfinden. Dies soll die Generierung eines Schlüsselpaares oder einer Signatur, durch z.B. Malware, unterbinden.

5.4.2 Man-In-The-Middle

Ein Man-In-The-Middle (MITM)-Angriff während der Authentifizierung kann in den meisten Fällen erkannt werden.

Wenn ein Nutzer seinen Token korrekt bei einer Einrichtung registriert hat und danach eine andere Einrichtung einen MITM-Angriff bei der Authentifizierung durchführen will, so wird der Stick (wie oben beschrieben) nicht antworten. Der Key-Handle der richtigen Einrichtung wird nicht zu den Daten des Angreifers passen.

Das U2F-Protokoll kann auch raffiniertere MITM-Angriffe erkennen. Wird der Stick zu einer Signatur aufgefordert (um einen Account bei einer Einrichtung zu authentisieren), so schickt der Browser einen Hash über die Daten, die er von der anfragenden Einrichtung sieht. Diese Daten bezeichnen wir als Client-Data.

Zur Client-Data gehören[19]

- eine zufällige, von der Einrichtung gesendete Challenge
- der Hostname der Webseite, die die Signatur verlangt
- (optional falls TLS ChannelID verwendet wurde) der öffentliche Schlüssel für diese Verbindung

[19] Quelle: S. Srinivas, D. Balfaz, E. Tiffany: FIDO alliance.Univrsal 2nd Factor (U2F) Overview. 2014.

Der Stick erhält also einen Hash über diese Daten von dem Browser. Zusätzlich werden ihm, wie oben beschrieben, ein Hash über die dienstspezifischen Daten der Einrichtung und der, im besten Fall, dazugehörige Key-Handle übergeben. Der Stick prüft, ob der Key-Handle zu dem Hash der dienstspezifischen Daten gehört. Falls ja, erstellt er mit dem privaten Schlüssel für den Account eine Signatur über den Hash der Client-Data. Diese Signatur und die Client-Data werden, als Rückgabe der Signaturaufforderung, an die Webseite der Einrichtung geliefert. Zuerst prüft die Einrichtung ob die Signatur zur übermittelten Client-Data gehört, indem sie den abgespeicherten öffentlichen Schlüssel für den Account zur Verifikation der Signatur benutzt. Im positiven Fall, kann die Einrichtung die Client-Data nach einem MITM-Angriff wie folgt untersuchen:

- Wenn der Name der Einrichtung in der Client-Data nicht zu der rechtmäßigen Einrichtung gehört, ist ein Angreifer vorhanden. Selbst wenn der schlaue Man-In-The-Middle bei der Registrierung des Sticks bei einer berechtigten Einrichtung, anwesend war und er erreicht hat, dass der Key-Handle zu seinem eigenen Namen passt, wird er auf diese Weise erkannt. (Für einen Angreifer, der nur bei der Authentifizierung, jedoch nicht bei der Registrierung anwesend ist, wird, wie oben beschrieben, der U2F-Stick keine Signatur erstellen).
- Wenn in der Client-Data eine ChannelID vorhanden ist, die nicht mit der ChannelID der Einrichtung übereinstimmt, ist ein Man-In-The-Middle vorhanden. Selbst wenn der Angreifer ein aktuell gültiges SSL-Zertifikat für den IT-Dienst besitzt und nicht durch den Namen unterscheidbar ist, wird er erkannt.

Es ist dennoch möglich einen erfolgreichen MITM-Angriff durchzuführen, wenn

- der Angreifer in der Lage ist ein Zertifikat für den Namen einer zulässigen Einrichtung (bei der sich der Nutzer registrieren wird), ausgestellt von einer gültigen Zertifizierungsstelle, zu erhalten und
- ChannelIDs nicht vom benutzten Browser unterstützt werden.
- der Nutzer einen kompromittierten Browser verwendet.

Dies ist jedoch schwer zu erreichen.

Ein MITM-Angriff gegen den das U2F-Protokoll nicht schützt, existiert weiterhin. Angenommen, eine Einrichtung bietet die Möglichkeit einer Authentifizierung mit einem Passwort. Der Nutzer kann (nach der eigentlichen Registrierung) nachträglich einen Stick registrieren und zukünftig mit dem U2F-Stick authentifiziert werden. Aber das Registrieren eines Sticks wird nicht zwingend von der Einrichtung verlangt. Wenn ein Man-In-The-Middle von der eigentlichen Registrierung an vorhanden ist, kann er den Wunsch des Nutzers einen zweiten Faktor zu registrieren abfangen und den Stick bei sich selbst registrieren. Der rechtmäßige IT-Dienst sieht immer noch, dass der Nutzer nur über das Passwort authentifizierbar ist. Bei der Authentifizierung kann der Man-In-The-Middle den Stick akzeptieren und eine Authentisierung des Nutzers durch das Passwort bei der eigentlichen Einrichtung vornehmen. Der Nutzer bemerkt den Angreifer nicht. Er würde davon ausgehen, dass er sich mit einer 2FA authentisiert. Diesem Angriff kann die Einrichtung entgegenwirken, indem sie den Nutzer sicher (zum Beispiel über einen anderen Kanal oder durch eine Signatur) mitteilt, ob ein Stick erfolgreich registriert wurde.

5.4.3 Echtheit des U2F-Geräts

Damit das U2F-Protokoll sicher ist, müssen die U2F-Sticks nach bestimmten Sicherheitsstandards hergestellt werden. Wenn zum Beispiel der Key-Handle den privaten Schlüssel beinhaltet[20], sollte eine Zertifizierungsstelle (wie z.B. die FIDO-Alliance[21]) zertifizieren, dass die Umsetzung sicher ist. Außerdem sollte der kryptografisch sichere Bereich auf dem Stick starke Sicherheitseigenschaften besitzen.

Deshalb muss eine Einrichtung überprüfen können, ob ein Token für sie akzeptabel ist. Für diesen Zweck besitzt die Einrichtung eine Datenbank und vergleicht den Typ des Tokens mit den Datenbankeinträgen. Zum Beispiel kann eine Einrichtung festlegen, dass sie nur U2F-Geräte mit Hardwarekomponenten zulässt (zur Erinnerung: U2F-Geräte können auch als reine Softwareimplemen-

[20] Der private Schlüssel kann im Key-Handle bei der Einrichtung verschlüsselt gespeichert werden. Dafür verschlüsselt der U2FToken den privaten Schlüssel.
[21] Wird von der FIDO-Alliance vorgeschlagen.

tierung existieren).

Damit ein Token beweisen kann welchen Typ er besitzt, hat er ein Bezeugungsschlüsselpaar. Diesen Schlüssel teilt er mit vielen weiteren Tokens, die vom gleichen Hersteller sind und den gleichen Typ besitzen. Dadurch kann ein Token nicht eindeutig identifiziert werden. Wenn der Token bei einer Einrichtung für einen Account registriert wird, wird der öffentliche Schlüssel für diesen Account, mit dem privaten Bezeugungsschlüssel signiert, übergeben. Die Idee ist, dass die öffentlichen Bezeugungsschlüssel für jeden Token-Hersteller, z.B. in einer Liste, öffentlich zugänglich sind. Damit kann die Einrichtung prüfen von welchem Hersteller der Token ist und ob sie ihm vertraut. Wie genau diese Schlüssel veröffentlicht werden wird noch entschieden. Der private Bezeugungsschlüssel wird im sicheren Bereich des Sticks gespeichert[22].

Durch den Bezeugungsschlüssel kann ein Gerät sicher nachweisen, wer sein Hersteller ist. Die Einrichtung kann dann entscheiden, ob sie dem Hersteller vertraut. Dadurch wird aber nicht bewiesen, dass der Stick tatsächlich den Standards genügt. Ein Bezeugungsschlüssel ist also eine Zertifizierung durch den Hersteller. Sticks können auch ohne einen Bezeugungsschlüssel existieren (oder der öffentliche Bezeugungsschlüssel wurde nicht freigegeben). Eine Einrichtung kann den Stick dennoch (mit dem Wissen, dass der Hersteller des Sticks unbekannt ist) akzeptieren.

Geklonte U2F-Geräte erkennen

Hersteller die keine Bezeugungsschlüssel nutzen wollen oder können, die Sticks ohne kryptografisch sichere Elemente herstellen oder reine Softwareimplementierungen anbieten, sollen auch die Möglichkeit haben das U2F-Protokoll zu verwenden. Das Schwierige dabei ist, dass solche U2F-Geräte kompromittiert und kopiert werden können.

Darum wurde entschieden, dass U2F-Geräte einen oder mehrere Zähler beinhalten. Ein Zähler zählt die durch den Stick getätigten Signaturen. Entweder existiert ein Zähler für jeden Schlüssel (das benötigt mehr Speicherplatz) oder

[22] Quelle: S. Srinivas, D. Balfaz, E. Tiffany: FIDO alliance.Univrsal 2nd Factor (U2F) Overview. 2014.

einer für alle Schlüssel gemeinsam (dadurch können aber Informationen zusammengeführt werden). Gemeinsam genutzte Zähler sind, durch die Einteilung der Schlüssel in Gruppen, auch denkbar.

Der Wert des Zählers wird für jede Signatur vom Stick zum Browser gesendet. Der Browser leitet den Wert weiter zu der Einrichtung, die die Signatur erhalten soll. Außerdem konkateniert der Stick den Zähler an den Hash von der Client-Data, vor der Signierung. So kann die Einrichtung sicherstellen, dass der Zähler nicht manipuliert wurde.

Die Einrichtung vergleicht einen frisch erhaltenen Zähler, mit dem Zähler, den sie früher vom gleichen Stick bekommen hatte. Wenn der frische Wert kleiner ist als der alte, wurde der Stick kopiert (mindestens ein weiterer besitzt den privaten Schlüssel für diese Signatur). Trotz des Zählers können Sticks kopiert werden, ohne erkannt zu werden. Zum Beispiel wenn ein Stick kopiert und das Original danach nie mehr verwendet wird, wird dies nicht bemerkt.

5.4.4 Zusammenfassung

1. Ein U2F-Gerät hat keine globale Kennzeichnung, die es über Webseiten oder Einrichtungen hinweg identifizierbar macht.
2. Innerhalb einer Einrichtung oder einer Webseite hat das U2F-Gerät keine globale (unabhängig von den Nutzeraccounts) Kennung.
    - Wenn zwei Nutzer ein Gerät teilen und beide bei der selben Webseite (mit diesem Stick) angemeldet sind, hat die Webseite nicht die Möglichkeit diese Accounts über eine Kennung des Tokens in Verbindung zu bringen.[23]
    - Wenn ein Nutzer sein U2F-Gerät verliert, kann derjenige, der es

[23] Die Webseite kann durch ausprobieren herausfinden ob zwei Accounts den selben Stick benutzen. Seien die Accounts, die für die selbe Einrichtung mit selben Stick erstellt wurden, A1 und A2. Wenn ein Nutzer sich für den Account A1 authentisieren möchte, sendet die Einrichtung den Key-Handle für A2 (diesen Key-Handle muss sie unter ihren gespeicherten Key-Handles raten, da keine Verbindung zwischen den Key-Handles für A1 und A2 besteht). Der Stick hat keine Zuordnung von Key-Handles zu Accounts. Der Stick gibt eine gültige Signatur zurück und die Einrichtung weiß damit, dass dieser Stick für A1 und A2 benutzt wird. Nutzer, die das verhindern wollen, müssen für mehrere Accounts bei der selben Einrichtung verschiedene Sticks nutzen. Nach S. Srinivas, D. Balfaz, E. Tiffany: FIDO alliance.Univrsal 2nd Factor (U2F) Overview. 2014.

findet, das Gerät nicht nutzen, um die Accounts des Nutzers auf einer Webseite zu sehen.

3. Wenn ein Schlüssel von einem U2F-Gerät für eine Einrichtung ausgestellt wurde, kann nur sie den Schlüssel benutzen.
    - Da für jeden Account ein eigenes Schlüsselpaar erzeugt wird, können Einrichtungen keine Informationen zu gemeinsamen Nutzern über ihre gespeicherten Schlüssel zusammenführen.
4. Damit ein Stick auf eine Authentisierungs- oder Registrierungsanfrage reagiert, muss er erst aktiviert werden (z.B. mit einem Knopfdruck). Damit kann ein Authentisierungsvorgang nicht allein von einem Programm auf dem Rechner durchgeführt werden.

5.5 Produktionskosten reduzieren

Einer der Nachteile bei der Nutzung einer 2FA mit einem Gegenstand, den der Nutzer besitzt, sind die Produktionskosten für diese Token. Deshalb wurde bei der Entwicklung des U2F-Protokolls die Minimierung der Produktionskosten als ein Ziel gesetzt[24].

Damit ein physikalischer Token möglichst günstig hergestellt werden kann, ist es zwingend notwendig keinen oder nur minimalen sicheren Speicher zu erlauben. Mit dem U2F-Protokoll ist dies möglich. Für einen Account, bei dem der Stick registriert wird, muss auf den ersten Blick nur ein privater Schlüssel sicher auf dem Stick gespeichert werden. Der, an die Einrichtung übermittelte Key-Handle für diesen Account kann als Index verwendet werden, um den dazugehörigen privaten Schlüssel auf dem Stick zu finden. Um weniger internen Speicher auf Sticks zu benutzen hat die FIDO-Alliance vorgegeben, dass der Key-Handle den privaten Schlüssel und den Hash der dienstspezifischen Daten der Einrichtung beinhalten darf. Dafür muss er mit einem verpackendem Schlüssel[25] verschlüsselt

[24] Quelle: S. Srinivas, D. Balfaz, E. Tiffany: FIDO alliance.Univrsal 2nd Factor (U2F) Overview. 2014.

[25] Hier soll das Verschlüsselungsverfahren nach der verwendeten Umsetzung des U2F-Tokens ausgewählt werden. Für manche Hardware kann es sinnvoller sein "schwächere" Verfahren (z.B. 3DES) zu nutzen als Starke (z.B. AES).
Quelle: D. Balfaz: FIDO alliance. U2F Implementation Considerations. 2014.

worden sein, den nur der Stick (der den privaten Schlüssel erstellt hatte) kennt. Wenn der Stick für eine Authentifizierung einen Key-Handle erhält, kann er den privaten Schlüssel und den Hash auspacken. Bei dieser Lösung muss nur ein einziger Schlüssel für alle Accounts sicher abgespeichert werden.

Alternativ können Informationen (privater Schlüssel + Hash oder nur der private Schlüssel) verschlüsselt auf einem unsicherem Bereich des Sticks, in einer Tabelle, gespeichert werden. Dann dient der Key-Handle als Index in der Tabelle. Daraus folgt aber, dass die Anzahl der Accounts, bei denen der Stick registriert werden kann, durch den Speicher begrenzt wird.

5.6 Kommunikation

Die Kommunikation mit dem Stick kann in zwei Anwendungsfälle (ein Schlüsselpaar für eine Registrierung erzeugen oder eine Signatur für eine Authentisierung generieren) eingeteilt werden. In dem Browser, den der Nutzer verwendet, sind diese beiden Funktionen eingebaut (als JavaScript-Funktionen). Eine Einrichtung oder eine Webseite ruft diese Funktionen auf, um mit dem Stick zu kommunizieren. Im Folgenden werden die Kommunikationsfälle ohne sicherheitsrelevante Aspekte wie Client-Data und dienstspezifische Daten beschrieben und in Abbildung 4 dargestellt.

5.6.1 Registrierung

1. Um einen Stick bei einer Einrichtung für einen Account zu registrieren, muss der Nutzer sich erst authentisieren (z.B. mit dem Usernamen und dem dazugehörigen Passwort).
2. Anschließend ruft die Webseite der Einrichtung die JavaScript-Funktion zur Erzeugung des Schlüsselpaars auf.
3. Der Browser zeigt dem Nutzer eine Warnung, dass eine Kommunikation mit seinen angeschlossenen U2F-Geräten gestartet wird (dieser Schritt kann für weitere Registrierungen durch den Nutzer abgestellt werden). Der Nutzer bestätigt diese Warnung.

4. Alle angeschlossenen U2F-Geräte werden aufgefordert ein Schlüsselpaar zu erzeugen.
5. Der erste Stick, der aktiviert wird (durch einen Knopfdruck) und damit die Anwesenheit des Nutzers bestätigt, antwortet auf diese Aufforderung (öffentlicher Schlüssel und Key-Handle werden übergeben).
6. Der Browser verpackt die Antwort und gibt sie der Webseite der Einrichtung als Rückgabewert des Funktionsaufrufs zurück.
7. Die Einrichtung speichert die Antwort (Key-Handle + öffentlicher Schlüssel) für diesen Account.

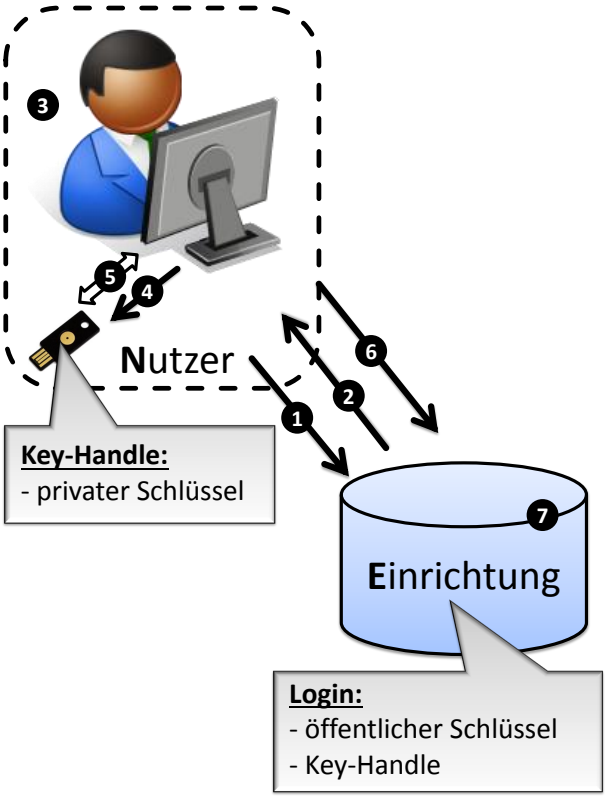

Abbildung 4: U2F Kommunikation

5.6.2 Authentisierung

1. Der Nutzer startet den Authentisierungsprozess mit seinem Nutzernamen und sein Passwort (das Passwort kann auch weggelassen werden, wenn die Webseite nur eine Authentisierung mit dem Stick verlangt).

2. Die Webseite der Einrichtung ruft mit dem Key-Handle für den Account und einer Nonce die JavaScript Funktion auf, die im erfolgreichen Fall eine Signatur vom Stick zurückgibt.
3. Der Browser zeigt, wie auch bei der Registrierung, eine Warnung an, dass eine Kommunikation mit den U2F-Geräten gestartet wird. Diese Warnung kann abgeschaltet werden. Der Nutzer bestätigt die Warnung.
4. Der Browser fordert alle angeschlossenen U2F-Geräte zu einer Signatur auf.
5. Der Nutzer aktiviert einen Stick, per Knopfdruck und dieser Stick erzeugt die Signatur. Wenn der Nutzer bereits mehrere Sticks bei dem Dienstanbieter registriert hat, können mehrere Sticks aktiviert werden, ansonsten wird nur die erste Antwort gesendet.
6. Die JavaScript Funktion übergibt der Webseite der Einrichtung die Client-Data und die Antwort des Sticks/der Sticks als Rückgabewert.
7. Die Einrichtung prüft, ob die Signatur passt (falls mehrere Sticks aktiviert wurden, wird überprüft, ob eine der Signaturen der erwarteten Signatur entspricht).

6 Rücksetzen eines U2F-Accounts

Einer der Nachteile einer besitzbasierten 2FA ist, dass sich der Nutzer nicht authentisieren kann, wenn der geforderte Gegenstand nicht präsent oder defekt ist. In unserem Fall also wenn der U2F-Stick gestohlen wird oder schlichtweg verloren geht (die in diesem Kapitel vorgestellte Lösung kann auch benutzt werden, wenn der Nutzer einen weiteren Stick an seinen Account binden möchte, obwohl der alte Stick noch vorhanden ist). Dem Nutzer soll die Möglichkeit angeboten werden, einen neuen Stick möglichst angenehm, pseudonym und sicher bei all jenen Einrichtungen zu registrieren, bei denen der alte Stick auch registriert war. Für die Umsetzung wird das Auth^2(nPA)-Konzept verwendet.

Um einen neuen Stick an einen bereits vorhandenen Account sicher anzubinden sind folgende Komponenten beteiligt:

- Der Nutzer, der bereits mit einem U2F-Stick bei der Einrichtung registriert ist. Er besitzt einen nPA mit freigeschalteter eID-Funktion und einen zweiten U2F-Stick, den er bei der Einrichtung registrieren möchte. Außerdem hat er die nötige Hard- und Software (z.B. Computer mit der Ausweisapp 2 und Kartenleser), um die eID-Funktion seines nPAs zu nutzen.
- Eine Einrichtung, die einen IT-Dienst erbringt und bei der der Nutzer registriert ist. Sie bietet die Möglichkeit einer 2FA mit einem U2F-Stick an und erlaubt das Rücksetzen eines Sticks mit dem nPA.
- Ein Auth^2(nPA)-Dienst (Dienstanbieter im Sinne des Auth^2(nPA)-Konzepts), der den eID-Service übernimmt und die Berechtigung (ein Berechtigungszertifikat und den privaten Schlüssel dazu) dafür besitzt.

6.1 Anforderungen und Lösungsidee

Der Nutzer hat einen USB-Stick (im Folgenden als T_1 bezeichnet), der für seinen Account bei einer Einrichtung registriert ist, verloren oder möchte nun einen zweiten Stick T_2 an den gleichen Account anbinden. Dafür muss er sich gegenüber der Einrichtung authentisieren. Dabei erwartet man eine höhere Sicherheit als für gewöhnliche Authentisierungen wie das Einloggen. Kein Anderer soll die Möglichkeit haben, einen weiteren Stick, ohne Einverständnis des Nutzers, für seine Accounts zu registrieren oder den alten zu entfernen. Um diese Sicherheit zu erreichen, muss der Nutzer zweifelsfrei beweisen, dass er derjenige ist, dem der Account gehört, ohne weitere personenbezogene Daten (Daten, die nicht bereits für den Account gespeichert sind) preiszugeben. Da der Account prinzipiell auf ein Pseudonym des Nutzers basieren kann, muss die Authentisierung auch Pseudonymität garantieren.

Der Auth^2(nPA)-Dienst, der für die Authentisierung verwendet wird, darf nicht erfahren, bei welchen Einrichtungen ein Nutzer registriert ist oder war und soll nicht in der Lage sein, Informationen zwischen verschiedenen Diensten oder Accounts zusammenzuführen.

Der Nutzer registriert T_2, wie in Abschnitt 5.6.1, und führt dabei eine Au-

thentisierung, ähnlich wie in 4.3.2, durch. Um das zu erlauben musste bei der Registrierung von T_1 ein Referenzwert (R' in 4.3.1) bei der Einrichtung, mit Hilfe des Nutzers und dem Auth^2(nPA)-Dienst (der Nutzer ist N nach 4.3, die Einrichtung ist E und das Auth^2(nPA)-Dienst ist D), gespeichert werden. Dabei generierte und speicherte die Einrichtung G_1 für den Nutzeraccount und der Auth^2(nPA)-Dienst speicherte das zweite Teilgeheimnis G_2 für die rID des Nutzers. Mit diesen zwei Geheimnissen kann, der Referenzwert für spätere Authentisierungen (z.B. für das Anbinden von T_2), erneut berechnet werden.

Wenn der Nutzer T_2 wie gewohnt registriert, muss R frisch berechnet und mit dem gespeicherten Referenzwert verglichen werden. Sind diese beiden gleich, so ist der Nutzer authentifiziert. Nach Wunsch des Nutzers kann T_1 (der Key-Handle und der öffentliche Schlüssel zu T_1[26]) aus der Datenbank der Einrichtung entfernt oder beibehalten werden. **Wurden der Key-Handle und der öffentliche Schlüssel zu T_1 bei der Einrichtung entfernt (das ist sinnvoll, wenn der Nutzer den Stick verloren hat), so kann derjenige, der den Stick findet, sich, selbst wenn er die Login-Daten kennt, nicht mit dem Stick für den Nutzeraccount authentisieren, weil die Einrichtung den Stick nicht weiter akzeptiert.** Wurden die Daten zu T_1 nicht bei der Einrichtung entfernt, können beide Sticks jeweils für Authentisierungen verwendet werden.

So kann die Pseudonymität des Nutzers bei der Einrichtung gewahrt werden und auch der Auth^2(nPA)-Dienst kann keine Informationen über die Accounts eines Nutzers zusammenführen.

6.2 Kommunikation

Damit die Accounts auf einen zweiten Stick übertragen werden können, muss bei der ersten Registrierung eines U2F-Sticks der Referenzwert für die späteren Authentifzierungen durch den Auth^2(nPA)-Dienst von der Einrichtung gespeichert werden. Die folgenden Kommunikationsfälle werden in Abbildung 5 dargestellt.

[26] Der Referenzwert R wird nicht gelöscht, weil er für weitere Authentifizierungen genutzt werden kann und unabhängig von T_1 ist.

6.2.1 Erste Registrierung (späteres Rücksetzen möglich)

1. N authentisiert sich mit dem Nutzernamen und dem Passwort für seinen Account (ein Passwort muss nicht von der Einrichtung verlangt werden) bei E.
2. Die Webseite der Einrichtung ruft die JavaScript Funktion zur Erzeugung eines Schlüsselpaars durch den U2F-Stick des Nutzers auf. Die Einrichtung generiert zusätzlich eine Authentisierungsanfrage (für das Login des Nutzers) für D, erzeugt dafür das Teilgeheimnis G_1 und übergibt es dem Nutzer. Diese Anfrage ist für D verschlüsselt.
3. N leitet die Authentisierungsanfrage an D weiter.
4. Der Nutzer sieht eine Warnung (dass die Kommunikation mit den angeschlossenen Sticks gestartet wird) des Browsers und gibt sein Einverständnis für die Ausführung. Dieser Schritt kann durch den Nutzer abgestellt werden und wird dann bei weiteren Registrierungen ausgelassen.
5. Alle angeschlossenen U2F-Geräte werden aufgefordert ein Schlüsselpaar zu erzeugen.
6. Der erste Stick, der durch den Nutzer aktiviert wird, antwortet auf die Aufforderung (generiert ein Schlüsselpaar und einen Key-Handle).
7. Der Nutzer authentisiert sich mit der rID über die eID-Funktion seines nPAs bei D.
8. D übergibt dem Nutzer den frisch berechneten, signierten und für E verschlüsselten Referenzwert.
9. Der Browser des Nutzers verpackt die Antwort des U2F-Sticks und gibt sie der Webseite des Dienstanbieters als Rückgabewert des Funktionsaufrufs. N leitet den Referenzwert an E weiter.
10. E speichert den Referenzwert und die Daten des Sticks für den Account.

Die Schritte 4-6 sind unabhängig von den Schritten 3,7 und 8 und müssen nicht in dieser Reihenfolge (innerhalb dieser zwei Gruppen muss die Reihenfolge eingehalten werden, untereinander kann die Reihenfolge jedoch beliebig sein) ausgeführt werden. Im Schritt 9 werden die beiden Ergebnisse (Antwort des

U2F-Sticks und Referenzwert), die in Schritt 2 verlangt wurden, an E übergeben. In dieser Kommunikation wird der Referenzwert nicht, anonymisiert durch einen Proxy, vom Dienstanbieter (E) abgeholt, sondern über den Nutzer weitergeleitet[27].

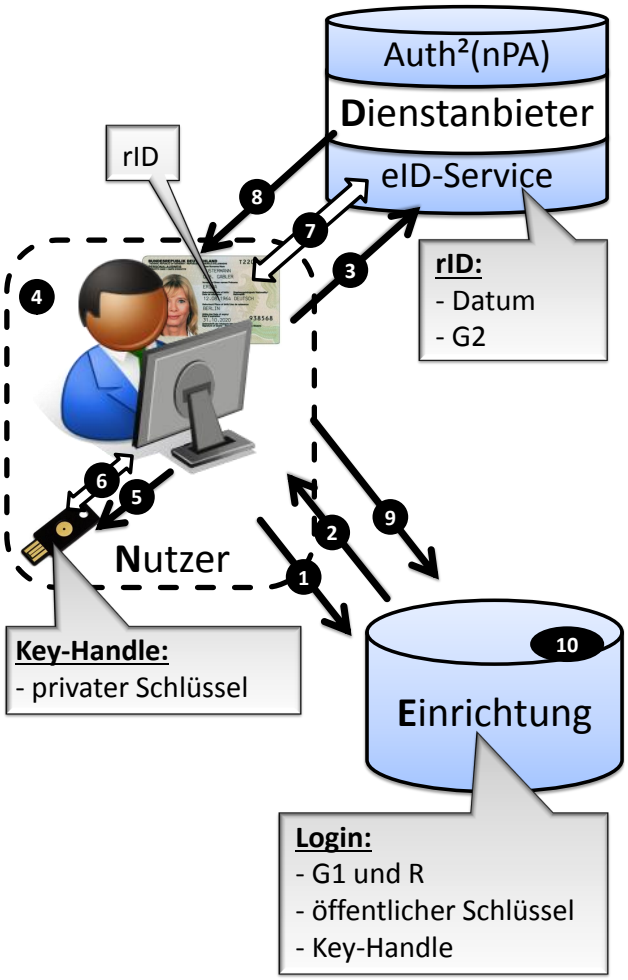

Abbildung 5: Kommunikation beim Rücksetzen eines U2F-Tokens mit dem nPA

6.2.2 Registrierung eines weiteren Sticks

Um einen zweiten Stick für einen Account zu registrieren, bei dem bereits ein Stick registriert ist, muss wie oben beschrieben der Referenzwert bei der Einrichtung existieren.

[27] Der Referenzwert ist für E verschlüsselt und kann vom Nutzer nicht eingesehen werden. Durch Frische wird erreicht, dass der Nutzer den Referenzwert nicht abspeichern und später verwenden kann. Damit kann die Einrichtung sicher sein, dass der Referenzwert für sie bestimmt ist und frisch berechnet wurde. Der Auth²(nPA)-Dienst gewinnt keine Informationen über die Accounts des Nutzers.

1. N wählt aus, dass er einen weiteren Stick für einen Account registrieren möchte. Er gibt zusätzlich an, ob der bereits registrierte Stick weiter verwendet werden soll und authentisiert sich mit seinem Nutzernamen und sein Passwort (falls ein Passwort verlangt wird) für den Account.
2. Die Webseite der Einrichtung ruft die JavaScript Funktion zur Erzeugung eines Schlüsselpaars durch den neuen U2F-Stick auf. Außerdem übergibt die Einrichtung eine Authentisierungsanfrage (für das Anbinden eines neuen Sticks durch den Nutzer) und damit das Teilgeheimnis G_1, das bereits für diesen Account abgespeichert ist, für den Auth^2(nPA)-Dienst verschlüsselt an den Nutzer.
3. N leitet die Authentisierungsanfrage an D weiter.

Die Schritte 4-9 erfolgen analog zu 6.2.1.

10. E vergleicht den frisch erhaltenen Referenzwert mit dem gespeicherten Referenzwert und speichert die Daten des neuen Sticks für den Account. Falls N im ersten Schritt ausgewählt hatte, dass der alte Stick nicht mehr verwendet werden soll, werden die Daten des alten Sticks gelöscht.

7 Proof of Concept

Für die Implementierung wurde ein Softwareclient[28] verwendet. Der Client ersetzt den Browser und bietet eine Softwareimplementierung des U2F-Tokens an[29]. Die Einrichtung ist ein U2F-Server[30]. Das heißt, sie bietet die Möglichkeit einer Authentifizierung mit einem U2F-Stick an und erlaubt zusätzlich das Anbinden eines zweiten Sticks durch den nPA. Um den Auth^2(nPA)-Dienst zu realisieren, wurde ein Mock-up-Server geschrieben, der eine Authentisierung mit Nutzernamen und Passwort statt einem nPA akzeptiert[31]. Einen

[28] Die verwendete Softwareimplementierung wurde von dieser Seite entnommen (inzwischen wurde die Software weiterentwickelt): `https://pypi.python.org/pypi/python-u2flib-host/1.1.0` (abgerufen am 21.11.14)
[29] Der Client ersetzt also den Chrome-Broswer mit Extension und den Token.
[30] Die Version von der Webseite `https://pypi.python.org/pypi/python-u2flib-server/1.0.0` (abgerufen am 21.11.14) wurde übernommen und weiterentwickelt.
[31] Für die reale Umsetzung sollten die Technischerichtlinien BSI TR-03130-1 und BSI TR-03130-2 beachtet werden.

richtliniengetreuen eID-Server zu schreiben wäre zeitlich zu aufwändig für eine Bachelorarbeit, die vier Monate benötigt. Das Teilgeheimnis G_2 wird also durch Nutzername und Passwort ermittelt und nicht durch die rID. Der Server besitzt kein Berechtigungszertifikat.

7.1 Kommunikation im Proof of Concept

In Abbildung 6 wird die Kommunikation zwischen den Komponenten dargestellt. Die Nachrichten sind in abgerundete Rechtecke eingetragen und werden entlang der Pfeile, von Absender zum Empfänger, transportiert.

Der Nutzer hat bereits, wie in 6.2.1 beschrieben, einen Stick registriert und will einen weiteren Stick mit dem nPA registrieren. Dafür schickt er der Einrichtung eine Nachricht, die für die Registrierung den Nutzer- und Sticknamen[32] enthält. Daraufhin erzeugt die Einrichtung eine Challenge für den neuen Stick. Das Teilgeheimnis G_1 existiert bereits (wurde bei der Registrierung des ersten Sticks erstellt). Die Einrichtung erstellt die Authentifzierungsanfrage[33]. Für die Antwort wird ein ephemerer AES-Schlüssel erzeugt. Die Anfrage beinhaltet G_1, einen Zeitstempel, der als Frische dient, eine Session-ID, den AES-Schlüssel und einen Hash über diese Daten. Die, für den Auth^2(nPA)-Dienst verschlüsselte, Authentifizierungsanfrage wird mit der Challenge an den Nutzer gesendet.

Der Nutzer leitet die Challenge an den neuen Stick weiter und aktiviert den Stick. Der Stick antwortet mit der Response. Der Nutzer sendet außerdem die Authentifizierungsanfrage mit Nutzernamen und Passwort[34] an den Auth^2(nPA)-Dienst. Der Dienst bestimmt G_2, durch den Nutzernamen und das Passwort. Mit Hilfe von G_2 und dem erhaltenen G_1 berechnet er R, wenn der Hash korrekt ist und

[32] Der Stickname wurde verwendet, um die Daten des alten Sticks zu löschen. Die Einrichtung speichert nach dem U2F-Protokoll, wenn ein Stick registriert wird, den Key-Handle und den öffentlichen Schlüssel. Diese Daten können nicht verwendet werden, um den Stick zu identifizieren. Sie müssen gelöscht werden, damit die Einrichtung den Stick nicht mehr akzeptiert. Im Proof of Concept wurden einfachheitshalber Sticknamen verwendet, um den alten Stick zu sperren. Im realen Fall muss die Einrichtung alle Key-Handles für den Account des Nutzers schicken. Der Nutzer aktiviert die Sticks, die er besitzt. Die Key-Handles, für die der Nutzer keine Signaturen liefert, werden von der Einrichtung gelöscht. Die öffentlichen Schlüssel zu den Key-Handles werden ebenfalls gelöscht.

[33] Der Nutzer soll mit Hilfe des Auth^2(nPA)-Dienstes authentifiziert werden.

[34] Dieser Nutzername und das Passwort dienen dazu G_2 zu bestimmen. Sie sind also ein Ersatz für die rID, die im realen Einsatz verwendet werden soll.

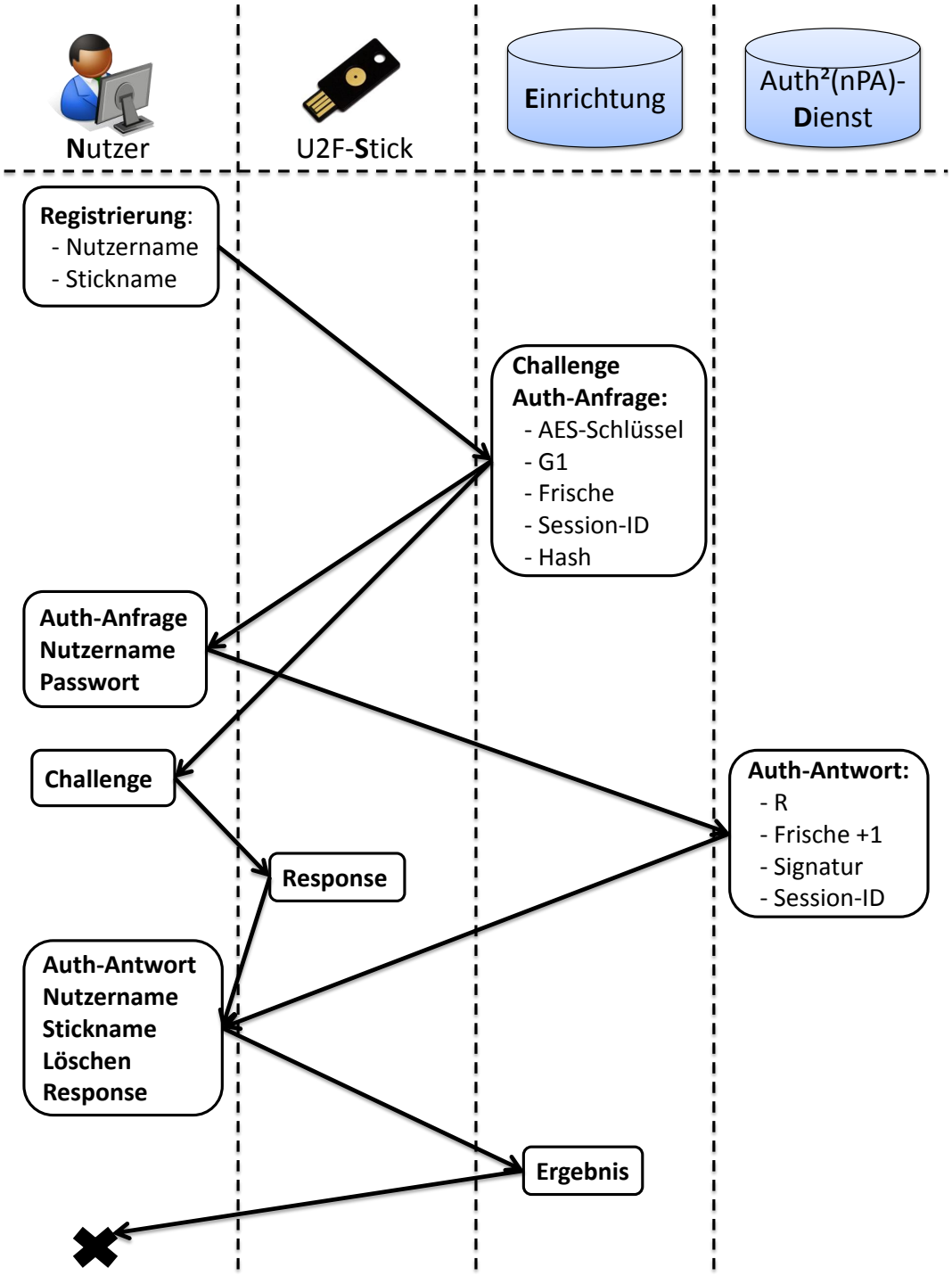

Abbildung 6: Kommunikation im Proof of Concept

der Zeitstempel nicht zu alt ist. Eine Differenz von 15 Sekunden zur aktuellen Zeit wird in dieser Implementierung maximal toleriert. Um den Hash zu prüfen, bildet der Dienst einen Hash über das erhaltene G_1, den Zitstempel und der Session-ID und vergleicht diesen Hash mit dem Erhaltenen. Der Dienst erstellt dann die Authentifizierungsantwort. Dazu gehören R, der erhaltene Zeitstempel +1, eine Signatur (mit dem RSA-Schlüssel des Auth^2(nPA)-Dienstes) über diese beiden Daten und die Session-ID. Der Auth^2(nPA)-Dienst sendet diese Antwort an den Nutzer.

Der Nutzer leitet die Antwort mit der Response vom Stick, dem Nutzernamen, dem Sticknamen und einem boolschen Wert (in der Abbildung Löschen genannt), der angibt, ob der alte Stick gesperrt werden soll, an die Einrichtung weiter. Durch die Session-ID weiß die Einrichtung welcher AES-Schlüssel zu der Antwort gehört. Sie prüft, ob zum Zeitstempel eins addiert wurde und ob die Signatur passt. Zur Verifikation der Signatur wird der öffentliche Schlüssel des Auth^2(nPA)-Dienstes benutzt. Anschließend vergleicht die Einrichtung das erhaltene R mit dem gespeicherten Referenzwert. Falls beide gleich sind, ist der Nutzer authentifiziert. Falls die Response des Sticks korrekt ist, ist der Stick nun registriert[35] und die Einrichtung antwortet mit „True“ (in der Abbildung Ergebnis genannt). Ansonsten antwortet die Einrichtung mit „False“. Wenn der Nutzer die Antwort erhalten hat, endet der Prozess.

7.2 Sicherheit

Der Nachrichtenaustausch zwischen der Einrichtung und dem Auth^2(nPA)-Dienst findet über den Nutzer statt. Weder der Nutzer noch Dritte sollen diese Nachrichten lesen oder wiederverwenden können.

7.2.1 Authentifizierungsanfrage

Für die Antwort einer Authentifizierungsanfrage erzeugt die Einrichtung einen ephemeren AES-Schlüssel. Der AES-Schlüssel wird mit dem öffentlichen RSA-

[35] Falls der boolsche Wert für Löschen True war, werden in der Implementierung die Daten des alten Sticks gelöscht. Im realen Fall sollen hier alle Key-Handles für diese Antwort übertragen werden und der Nutzer benutzt die Sticks, die er besitzt.

Schlüssel[36] des Auth^2(nPA)-Dienstes verschlüsselt. G_1, der Zeitstempel, die Session-ID und ein Hash (SHA-256) über diese Daten werden durch AES (im Cipher Block Chaining Mode) verschlüsselt.

Wenn der Auth^2(nPA)-Dienst die Anfrage erhält, nutzt er seinen privaten RSA-Schlüssel, um den AES-Schlüssel für diese Session zu entschlüsseln. Mit dem AES-Schlüssel entschlüsselt er dann das Teilgeheimnis G_1, den Zeitstempel, die Session-ID und den Hash. Durch den Hash wird eine unerkannte Manipulation der Nachricht verhindert. Durch den Zeitstempel wird eine Wiederverwendung 15 Sekunden nach der Erstellung der Anfrage ausgeschlossen. Die Session-ID wird nur für die Authentifizierungsantwort gebraucht.

7.2.2 Authentifizierungsantwort

Mit dem erhaltenen AES-Schlüssel (im Cipher Block Chaining Mode) werden R und der erhaltene Zeitstempel+1 verschlüsselt. Im Folgenden wird das Ergebnis als R_{AES} bezeichnet. Von R_{AES} erstellt der Auth^2(nPA)-Dienst eine RSA-Signatur (mit SHA-256). Zusätzlich wird die Session-ID unverschlüsselt übertragen.

Wenn die Einrichtung die Nachricht erhält, nutzt sie die Session-ID um den AES-Schlüssel zu identifizieren. Ist die Session-ID unbekannt, wird die Nachricht verworfen. Die Einrichtung prüft die Signatur, indem sie den SHA-256 Hash von R_{AES} bildet, den öffentlichen Schlüssel des Auth^2(nPA)-Dienstes für die Signatur benutzt und das Ergebnis mit dem selbst berechneten Hash vergleicht. Wenn sie gleich sind, entschlüsselt die Einrichtung R_{AES} und prüft ob zum Zeitstempel eins addiert wurde. Dann vergleicht sie R mit dem gespeicherten Referenzwert. Im positiven Fall, ist der Nutzer authentifiziert.

7.2.3 Vertraulichkeit und Integrität

Um G_1 zu entschlüsseln wird der AES-Schlüssel benötigt, der wiederum mit dem RSA-Schlüssel des Auth^2(nPA)-Dienstes verschlüsselt wurde. Damit können

[36] Der Auth^2(nPA)-Dienst besitzt, wie in 4.4 beschrieben, ein RSA Schlüsselpaar. Wir nutzen einen asymmetrischen Schlüssel, damit andere Einrichtungen die Authentifizierungsantwort nicht entschlüsseln können.

der Nutzer und Dritte nicht an G_1 gelangen. Da der ephemere AES-Schlüssel jedes mal neu erstellt wird, kann der Auth^2(nPA)-Dienst die Einrichtung nicht durch den verwendeten Schlüssel identifizieren. Der Hash wird verwendet damit eine Anfrage nicht unerkannt verändert werden kann. Durch die Frische wird erreicht, dass die gleiche Authentifizierungsanfrage nicht von Dritten abgespeichert und wiederverwendet werden kann. In der Authentifizierungsantwort wird zu der erhaltenen Frische eins addiert. Damit zeigt der Auth^2(nPA)-Dienst, dass die Antwort zur Anfrage gehört[37]. Das Ergebnis und der frisch berechnete Referenzwert R werden mit dem AES-Schlüssel verschlüsselt übertragen. Deshalb können der Nutzer und Dritte nicht den Referenzwert entschlüsseln und abspeichern. Der Hash[38] der beiden Daten wird vom Auth^2(nPA)-Dienst signiert. So kann die Einrichtung prüfen, ob R vom Auth^2(nPA)-Dienst stammt und eine Manipulation der Nachricht wird erkannt. Die Einrichtung und der Auth^2(nPA)-Dienst kommunizieren nie direkt miteinander.

Durch die Kenntnis des öffentlichen RSA-Schlüssels zeigt die Einrichtung, dass sie berechtigt ist den Auth^2(nPA)-Dienst zu nutzen. Der Schlüssel wird periodisch erneuert und kann von den Einrichtungen abgeholt werden. Dafür benutzen sie ihr Zugangskonto beim Auth^2(nPA)-Dienst.[39]

7.2.4 Angriff

Für die Authentifizierung des Nutzers, wird vom Auth^2(nPA)-Dienst G_1 und die dazugehörige rID benötigt. Erst dann kann der Dienst G_2 ermitteln und damit R berechnen. Ein Angreifer, der einen zweiten Stick ohne Einverständnis des Nutzers registrieren möchte, muss:

1. dafür sorgen, dass die Einrichtung eine Authentifizierungsanfrage erzeugt. Dafür muss der Angreifer mit dem Konto des berechtigten Nutzers eingeloggt sein oder zumindest den Loginnamen kennen.

2. über den nPA des Nutzers verfügen und die eID-PIN kennen.

[37] Wenn die Einrichtung eine Authentifizierungsanfrage erstellt, speichert sie den Zeitstempel zur Session-ID. Ist der Zeitstempel älter als eine Stunde, wird die Session-ID gelöscht.
[38] Für den Hash wird SHA-256 benutzt.
[39] Quelle: Müller, Redlich, Jeschke. Auth^2(nPA) Starke Authentifizierung mit nPA für jedermann. SP Gabler Verlag. 2011.

Es ist sehr schwer den ersten Schritt zu umgehen. Selbst wenn der Angreifer das Teilgeheimnis G_1 und den öffentlichen RSA-Schlüssel des Auth^2(nPA)-Dienstes kennt und eine Authentifizierungsanfrage generiert, wird die Einrichtung die Antwort nicht akzeptieren. Damit die Einrichtung eine Antwort akzeptiert, muss dazu eine Session-ID existieren. Um eine gültige Session-ID, den passenden AES-Schlüssel und Zeitstempel durch einen MITM-Angriff zu erhalten, muss ein Angreifer den privaten RSA-Schlüssel des Auth^2(nPA)-Dienstes kennen. Dies sollte unter keinen Umständen möglich sein. Der Angreifer kann dennoch zu den Daten gelangen, wenn die Einrichtung ihre Daten unsicher aufbewahrt. Durch die Session-ID ist keine Bindung zum Nutzeraccount gegeben. Ein Angreifer kann durch einen eigenen Account eine Authentifizierungsanfrage erstellen lassen und diese Session-ID benutzen, falls er sie kennt.

Wenn der Angreifer nicht den nPA des Nutzers besitzt oder nicht die eID-PIN kennt, kann er eine Authentifizierungsantwort abspeichern. Für das Wiederverwenden müsste er die Nachricht so verändern, dass sie zum Zeitstempel und AES-Schlüssel der neuen Anfrage passt und außerdem muss die Signatur angepasst werden. Der Angreifer muss dann auch die Session-ID der neuen Anfrage kennen, damit die Einrichtung die Authentifizierungsantwort akzeptiert.

Die Einrichtungen sind selbst dafür verantwortlich, ihre Teilgeheimnisse G_1, Session-IDs, Zeitstempel und AES-Schlüssel zu schützen. Der Nutzer sollte seinen nPA unverzüglich sperren lassen, falls dieser geklaut wurde oder verloren gegangen ist. Außerdem sollte er wissen, dass die eID-PIN ein besonders schützenswertes Geheimnis ist.

7.3 Erweiterungsmöglichkeit

Wenn der Nutzer einen Stick verloren hat, möchte er selbstverständlich einen neuen Stick an allen Accounts, bei denen der Stick registriert war, registrieren. Damit er sich nicht für jeden Account sicher authentisieren (die eID-PIN eingeben) muss, ist ein Protokoll denkbar, das eine einmalige Authentisierung für alle Accounts oder für einen Stapel von Accounts zulässt. Damit der Auth^2(nPA)-Dienst keine Informationen zusammenführen kann, sollte eine Einrichtung

genügend Fakereferenzwerte bei dem Dienst besitzen. Diese Referenzwerte, die zu keinen echten rIDs gehören, werden zu echten Nutzeraccounts gemischt. Damit können Stapel gleicher Größe erzeugt werden.

Dazu kann der Browser die für die Authentisierung benötigten Daten (Webadresse der Einrichtung und Nutzername[40]) zu den Accounts eines Nutzers in einer Liste speichern. Wenn der Nutzer einen zweiten Stick registrieren möchte, wählt er im Browser aus, bei welchen Accounts der neue Stick registriert werden soll. Der Nutzer authentisiert sich bei dem Auth^2(nPA)-Dienst mit seinem nPA. Anschließend werden die Referenzwerte für die von dem Nutzer ausgewählten Accounts frisch berechnet. Dazu leitet der Browser die Authentifizierungsanweisungen (siehe Schritt 2 in 6.2.2) von den Einrichtungen gesammelt mit Fakeanfragen an den Auth^2(nPA)-Dienst weiter und der Auth^2(nPA)-Dienst gibt die Referenzwerte zusammen in der gleichen Reihenfolge zurück. So kann der Browser den Einrichtungen ihre verlangten Referenzwerte zuordnen und weiterleiten.

Das Weiterleiten der Nachrichten kann generell vom Browser übernommen werden, damit der Nutzer keine zusätzliche Software benötigt.

[40] Bei Zustimmung des Nutzers kann auch sein Passwort für die Accounts gespeichert werden, wenn es bei der Registrierung eines Sticks gebraucht wird.

8 Fazit

Bei der Umsetzung des U2F-Protokolls wurde auf Datensicherheit, geringe Produktionskosten und Benutzerfreundlichkeit geachtet. Der Nutzer muss keine zusätzliche Software downloaden oder Bedienungsanleitungen lesen. Da die Passwörter auf vierstellige Ziffern reduziert werden können, ohne an Sicherheit gegenüber einer Authentisierung mit komplexen und langen Passwörtern zu verlieren, ist es deutlich bequemer für den Nutzer. Der Stick eignet sich zusätzlich für alle Betriebssysteme und kann damit eine weite Verbreitung finden.

Dadurch, dass der Stick bei der Authentisierung vorhanden sein muss, wird eine hohe Sicherheit erreicht. Hat der Nutzer ihn verloren, kann er mit seinem nPA einen Neuen anbinden und den Alten sperren. Wenn ein Angreifer den verlorenen Stick findet bevor er gesperrt wurde, muss er sowohl den Nutzernamen als auch das Passwort für einen Account und die passende Einrichtung, um den Stick zu nutzen. Umgekehrt braucht ein Angreifer, der den Nutzernamen und das Passwort zum Beispiel durch Social Engineering kennt, den Stick für eine Authentisierung. Das U2F-Protokoll gleicht die Nachteile der einzelnen Authentisierungsverfahren (Anwesenheit des Sticks und Wissen des Passwortes) sehr gut aus und erreicht damit eine sehr hohe Sicherheit.

Das Ziel der Arbeit wurde erreicht, denn der nPA bietet eine sichere und günstige Möglichkeit für das Rücksetzen eines U2F-Accounts in Deutschland. In der beschriebenen Kommunikation (6.2.2) wurde das Authentisieren mit dem nPA im Registrierungsprozess des U2F-Tokens eingebettet. Der somit erzeugte Referenzwert kann benutzt werden, um weitere Sticks zu registrieren oder den Alten zu sperren. Durch die Nutzung der rID wird Pseudonymität beim Registrieren eines U2F-Tokens garantiert.

Literaturverzeichnis

[1] Personalausweis Beispielbild (n.d.) [Online]. Verfügbar unter: http://www.personalausweisportal.de/SharedDocs/Bilder/DE/Ausweis_stehend.jpg?__blob=poster&v=6 [04.11.2014]

[2] Personalausweis (n.d.) [Online]. Verfügbar unter: http://www.personalausweisportal.de/DE/Home/home_node.html [30.11.14]

[3] W. Müller, J. P. Redlich, M. Jeschke (2011). *Auth²(nPA) Starke Authentifizierung mit nPA für jedermann.*: SP Gabler Verlag. Verfügbar unter: http://link.springer.com/article/10.1007%2Fs11623-011-0116-9 [26.11.14]

[4] *AusweisApp 2* (n.d.) [Online] Verfügbar unter: https://www.ausweisapp.bund.de/ausweisapp2/ [04.12.14]

[5] Yubico (n.d.) [Online]. Verfügbar unter: https://www.yubico.com/press/images/ [16.11.14]

[6] FIDO-Alliance (n.d.) [Online]. Verfügbar unter: http://fidoalliance.org/about [30.11.14]

[7] S. Srinivas, D. Balfaz, E. Tiffany (2014). *Univrsal 2nd Factor (U2F) Overview*. FIDO-Alliance. Verfügbar unter: http://fidoalliance.org/specs/fido-u2f-overview-v1.0-rd-20140209.pdf [29.11.14]

[8] D. Balfaz (2014). *U2F Implementation Considerations*. FIDO-Alliance. Verfügbar unter: http://fidoalliance.org/specs/fido-u2f-implementation-considerations-v1.0-rd-20141008.pdf [21.11.14]

[9] D. Balfanz (2014). *FIDO U2F Application Isolation through Facet Identification*. FIDO-Alliance. Verfügbar unter: http://fidoalliance.org/specs/fido-u2f-application-isolation-through-facet-identification-v1.0-rd-20140209.pdf [30.11.14]

[10] D. Balfanz, J. Ehrensvard (2014). *FIDO U2F Raw Message Formats* .FIDO-Alliance. Verfügbar unter: https://fidoalliance.org/specs/fido-u2f-raw-message-formats-v1.0-rd-20141008.pdf [30.11.14]

[11] Sepecifications (n.d.) [Online]. Verfügbar unter: https://fidoalliance.org/specifications [16.11.14]

[12] BSI Technischerichtlinie TR03127 (2012) *Architektur elektronischer Personalausweis und elektronischer Aufenthaltstitel* [Online]. Verfügbar unter:

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03127/BSI-TR-03127_pdf.pdf?__blob=publicationFile [29.11.14]

[13] BSI Technischerichtlinie TR-03130-1 (2014) *Technical Guideline eID-Server* [Online]. Verfügbar unter: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03130/TR-03130_TR-eID-Server_Part1_pdf.pdf?__blob=publicationFile [08.12.14]

[14] BSI Technischerichtlinie TR-03130-2 (2014) *Technical Guideline eID-Server* [Online]. Verfügbar unter: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03130/TR-03130_TR-eID-Server_Part2_pdf.pdf?__blob=publicationFile [08.12.14]

[15] U2F-Host (n.d.) [Online]. Verfügbar unter: https://pypi.python.org/pypi/python-u2flib-host/1.1.0 [21.11.14]

[16] U2F-Server (n.d.) [Online]. Verfügbar unter: https://pypi.python.org/pypi/python-u2flib-server/1.0.0 [21.11.14]

[17] Wikipedia: *Zwei-Faktor-Authentifizierung* (n.d.) [Online]. Verfügbar unter: http://de.wikipedia.org/wiki/Zwei-Faktor-Authentifizierung [30.11.14]

[18] Wikipedia: *Personalausweis (Deutschland)* (n.d.) [Online]. Verfügbar unter: http://de.wikipedia.org/wiki/Personalausweis_(Deutschland)#Der_elektronische_Personalausweis_.28nPA.29 [30.11.14]

[19] Latexvorlage (n.d.) Online. Verfügbar unter: https://www.informatik.hu-berlin.de/studium/formulare/vorlagen [30.11.14]

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den 9. Dezember 2014 .....