Lufthansa Systems

DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Duale Hochschule Baden-Württemberg
Mannheim

Bachelorarbeit

Analyse und Integration einer passwortlosen Authentifizierung im Unternehmenskontext

Studiengang Cyber Security

Verfasser:	Luka Tsipitsoudis
Matrikelnummer:	4110112
Kurs:	TINF20CS1
Bearbeitungszeitraum:	06.06.2023 – 29.08.2023
Abgabedatum:	29.08.2023
Betreuer:	Stefan Köster

Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: „*Analyse und Integration einer passwortlosen Authentifizierung im Unternehmenskontext*“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

Abstract

Deutsch

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

LSY Lufthansa Systems GmbH & Co. KG
FIDO Fast Identity Online
W3C World Wide Web Consortium
SFA Single-Factor Authentication
MFA Multi-Factor Authentication
CTAP2 Client-to-Authenticator Protocol 2
ITU-T International Telecommunication Union Telecommunication Standardization Sector
MITM Man In The Middle
PAKE Password Authenticated Key Exchange
EUF-CMA Existential Unforgeability under a Chosen Message Attack
PQ Post-Quantum
KEM Key Encapsulation Mechanism
TLS Transport Layer Security
puvProtocol PIN/UV Auth Protocol
SUF Strongly Unforgeable
UF Unforgeable

1 Einführung

Diese Arbeit beschäftigt sich mit passwortlosen Authentifizierungsverfahren. Im Folgenden werden zunächst die Problemstellung und das Ziel der Arbeit erläutert. Anschließend wird der Aufbau der Arbeit beschrieben und auf verwandte Arbeiten eingegangen:

1.1 Problemstellung & Ziel der Arbeit

Die Problemstellung dieser Arbeit bezieht sich auf den aktuellen, passwortlosen Ansatz der Authentifizierung im Unternehmenskontext der Lufthansa Systems GmbH & Co. KG (LSY). Trotz ihrer hohen Etablierung und Verbreitung bieten passwortlose Authentifizierungsverfahren nicht nur Vorteile, sondern auch eine hohe Anzahl an Angriffsvektoren.

Ziel dieser Arbeit ist es daher passwortlose Authentifizierungsverfahren genauer zu betrachten. Verschiedene passwortlose Verfahren werden vorgestellt und ihre Vor- und Nachteile aufgezeigt. Dabei soll ein besonderes Augenmerk auf den Vergleich der Angriffsvektoren von passwortlosen und passwortbasierten Verfahren gelegt werden. Einer der passwortlosen Verfahren wird begründet ausgewählt und detailiierter betrachtet. Dabei wird analysiert, ob das Verafahren für die LSY geeignet ist und welche Anpassungen vorgenommen werden müssen. Betrachtet werden insbesondere die Aspekte der Sicherheit und der Benutzerfreundlichkeit. Der Fokus liegt auf der Frage, ob passwortlose Verfahren eine Alternative darstellen, welche Passwörter gänzlich ersetzen.

1.2 Aufbau der Arbeit

1.3 Referenzierte Arbeiten

2 Grundlagen

2.1 Einführung in cFront

2.2 CIA-Triade

2.3 Arten der Authentifizierung

- Die Authentifizierung dient häufig als erste Verteidigungslinie von Systemen. **boonkrong2012security**
- Faktor Something you know. Diese Methode nutzt Informationen, welche nur dem Nutzer bekannt sind, um seine Identität zu bestätigen **boonkrong2012security**.
- Faktor Something you have. Diese Methode nutzt physische Objekte, welche sich im Besitz des Nutzers befinden, um seine Identität zu bestätigen. Dazu gehören u.a. Smartcards und Hardware-Token **boonkrong2012security**.
- Faktor Something you are. Diese Methode nutzt biometrische Daten des Nutzers, um seine Identität zu bestätigen. Dazu gehören u.a. Fingerabdrücke, Iris-Scans und Gesichtserkennung **boonkrong2012security**.
- Ein Problem dieser Methode ist, dass sich menschliche Eigenschaften im Laufe der Zeit verändern können. Auch Verletzungen oder Krankheiten können die biometrischen Daten verändern **boonkrong2012security**.
- Nicht standardmäßig, aber weiterer Faktor ist something you perform or produce. Diese Methode nutzt beispielsweise die Stimme oder die (digitale) Unterschrift des Nutzers, um seine Identität zu bestätigen **boonkrong2012security**.

2.4 Passwortbasierte Authentifizierung

- Die heutzutage am häufigsten genutzte Methode zur Authentifizierung ist die passwortbasierte Authentifizierung **chanda2016password boonkrong2012security yildirim2019encouraging**.

- Die Sicherheit von Systemen basiert somit auf der Sicherheit der Passwörter **boonkrong2012security**.
- Passwörter gelten als eins der größten Risiken für Systeme, da sie viele Angriffsvektoren bieten **yildirim2019encouraging farke2020you**.
- 81% der Hackerangriffe basierten auf der Kompromittierung von Passwörtern **barbosa2021provable**.
- 2017 waren Phishing E-Mails die häufigste Angriffsmethode **barbosa2021provable**.
- Obwohl es bereits alternative Ansätze gibt, werden Passwörter weiterhin genutzt. Das liegt an der Einfachheit und dem geringen Aufwand, welche die Nutzung von Passwörtern mit sich bringt **yildirim2019encouraging**.
- Eine Vielzahl an großen Unternehmen wurden bereits Opfer von der Veröffentlichung von Passwörtern, obwohl ein hoher Aufwand betrieben wird diese zu schützen. Da sich die Enthüllung der Passwörter allerdings als Angriffsziel bei Angreifern etabliert hat, ist selbst ein hoher Aufwand nicht ausreichend **boonkrong2012security**.
- Dabei handelt es sich am häufigsten um alphanumerische Passwörter, welche aus einer Kombination von Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen bestehen **chanda2016password**.
- Passwörter können durch verschiedene Angriffe kompromittiert werden. Angreifer können Zugriff auf die Datenbank erhalten, in welcher die Passwörter gespeichert sind. Aber auch auf persönlicher Ebene können Passwörter erlangt werden. Aufgeschriebene Passwörter können in fremde Hände geraten. Auch Social Engineering kann genutzt werden, um Passörter mit Hilfe von Phishing oder Keyloggern zu erlangen. Häufig lassen sich Passwörter allerdings auch mit Hilfe von Brute-Force- oder Dictionary-Attacken kompromittieren **chanda2016password morii2017research**.
- Brute-Force-Attacken versuchen alle möglichen Kombinationen von Zeichen, welche ein Passwort enthalten kann, auszuprobieren. Je höher dabei die Anzahl an möglichen Kombinationen ist, desto aufwändiger wird es ein Passwort zu erraten.
- Je länger ein Passwort, desto schwieriger zu knacken. Länge auch wichtiger als Zeichenraum **chanda2016password**.
- Hier auch kurz auf die Mathematik dahinter eingehen.

- Studien zeigen, dass Nutzer dazu neigen gleiche oder ähnliche Passwörter für verschiedene Zugänge zu nutzen **chanda2016password ives2004domino**.
- Verfügen Angreifer über ein Passwort eines Nutzers, können häufig auch andere Zugänge übernommen werden **chanda2016password morii2017research**.
- Obwohl die Angriffsvektoren und Schwachstellen von Passwörtern schon lange bekannt sind, bleiben diese unverändert bestehen. **ives2004domino**.

2.4.1 Speicherung

- Viele Angreifer versuchen Passwörter zu kompromittieren, indem sie Zugriff auf die Datenbank erhalten, in welcher die Passwörter gespeichert sind. Mit Hilfe von Passwörtern erhoffen sich die Angreifer Zugriff auf Systeme und Netzwerke **boonkrong2012security**.
- Passwörter können auf verschiedene Arten gespeichert werden. Dadurch können verschiedene Angriffsvektoren entstehen **chanda2016password**.
- Plaintext am schlechtesten. Werden die Passwörter in lesbarer Form gespeichert, können Angreifer alle Passwörter auslesen, sobald sie Zugriff auf die Datenbank haben. Dabei muss kein weiterer Aufwand betrieben werden **chanda2016password**.
- Verschlüssellung besser, aber nicht optimal. Verschlüssellung ist zurückführbar. Gelangen Angreifer an den benötigten Schlüssel, können sie alle Passwörter entschlüsseln und auslesen **chanda2016password**.
- AM besten Hashing mit Salt. Sobald ein Passwort gehasht wurde, kann es nicht mehr zurückgerechnet werden. Durch einen individuellen Salt kann ebenfalls verhindert werden, dass Angreifer die Passwörter mit Hilfe von Rainbow-Tables entschlüsseln können **chanda2016password**.
- auch noch zwei salts möglich - einer public einer private. schützt vor offline angriffen **chanda2016password**.
- Vielleicht hier noch ganz kurz auf Hash Funktionen eingehen?

2.4.2 Faktor Mensch

- Die Sicherheit ist nicht nur von den technischen Aspekten abhängig **ives2004domino**.
- Ein Großteil der Angriffsfläche von Passwörtern entsteht durch den Faktor Mensch **yildirim2019encouraging**.
- Von Menschen erstellte Passwörter sind keine echten Zufallswerte. Das liegt insbesondere daran, dass Nutzer sich Passwörter merken können müssen. Daher beinhalten Passwörter häufig Informationen, welche einen Bezug zum Nutzer haben. Dazu gehören beispielsweise Namen, Geburtsdaten, Adressen oder andere persönliche Informationen. Auch Passwörter, welche einfache Tastaturmuster beinhalten sind sehr beliebt. Dazu zählen beispielsweise „qwertz“ oder „123456“ **chanda2016password boonkrong2012security yildirim2019encouraging**.
- Das Hauptproblem entsteht dabei durch die benötigte Einprägsamkeit der Passwörter **yildirim2019encouraging**.
- Es ist sehr schwierig für Nutzer sich verschiedene komplexe Passwörter zu merken. Daher neigen Nutzer dazu, einfache Passwörter zu nutzen oder Passwörter für verschiedene Zugänge zu wiederholen **chanda2016password**.
- Das ist der Hauptgrund dafür, dass Nutzer dazu neigen, einfache Passwörter zu nutzen oder Passwörter für verschiedene Zugänge zu wiederholen **yildirim2019encouraging**.
- Diese Faktoren führen dazu dass die Anzahl an genutzten Passwörtern deutlich geringer ist als die gesamte Menge an möglichen Passwörtern **boonkrong2012security**.
- Ebenfalls ist häufig die Motivation der Nutzer gering komplexe Passwörter zu erstellen. Dies liegt häufig daran, dass die Nutzer nicht die Gefahr erkennen und nicht überzeugt von Guidelines und Richtlinien zur Erstellung von Passwörtern sind **yildirim2019encouraging**.
- Nutzer tendieren dazu bewusst schwache Passwörter zu erstellen, die den Anforderungen der Richtlinien entsprechen. Das führt zu einem kontraproduktiven Effekt, da die Sicherheit geringer wird **yildirim2019encouraging**.
- Sehr komplexe Richtlinien führen demnach nicht zwangsmäßig zu einer höheren Sicherheit. Vielmehr kann das Gegenteil erreicht werden **yildirim2019encouraging morii2017research**.

- Aktive Internet-Nutzer verwalten durchschnittlich 15 Passwörter pro Tag **ives2004domino**.
- Eine der größten Schwachstellen ist also die Wahl des Passwortes durch den Nutzer **boonkrong2012security**.
- Ein Domino Effekt kann entstehen, wenn mit Hilfe eines Passwortes weitere Passwörter kompromittiert werden. So können mehrere Systeme indirekt davon betroffen sein, sobald ein Passwort kompromittiert wurde **ives2004domino**.
- Das macht von Menschen erstellte Passwörter anfälliger für Angriffe, da diese einfacher zu erraten sind **chanda2016password**.
-

2.5 Passwortlose Authentifizierung

- Die Fast Identity Online (FIDO) Allianz nutzt die Bezeichnung passwortlos, um eine Single-Factor Authentication (SFA) und Multi-Factor Authentication (MFA) mit Hilfe eines Authentifizierungsgerätes zu beschreiben **farke2020you**.

2.5.1 Magic Link

2.5.2 One Time Password (OTP)

- Passwörter die sich mit jedem Login ändern. Dadurch wird das Risiko verringert, dass das Passwort erraten werden kann **boonkrong2012security**.

2.5.3 Biometrische Daten

2.5.4 Public Key Cryptography

2.6 YubiKey

- Ein Security Key ist eine Hardware, welche es ermöglicht einen Nutzer zu authentifizieren, indem dieser mit dem Security Key interagiert (beispielsweise durch einen Knopfdruck) **reynolds2018tale**.
- Häufig werden Security Keys so designed, dass sie per USB an einen Computer angeschlossen werden können **reynolds2018tale**.
- Die YubiKeys 5 ermöglichen drei Arten der Authentifizierung: 1. SFA Ersetzt Passwörter durch ein passwortloses *tap-n-go* Verfahren. 2. **2FA!** (**2FA!**) Sichert ein Passwort zusätzlich mit einem *tap-n-go* Faktor ab. Der Security ist somit der zweite Faktor (*something you have*). 3. MFA Verbindet die passwortlose *tap-n-go* Authentifizierung mit einer PIN. (EIG AUCH SFA ODER NICHT)

2.6.1 Usability

- Vorteile:
- Ergebnisse zeigen, dass Nutzer grundsätzlich bereit sind, Passwörter durch passwortlose Verfahren zu ersetzen **lyastani2020fido2**.
- Passwortlose Verfahren mit Yubikey wurden mehr akzeptiert als tradionelle passwortbasierte Verfahren **lyastani2020fido2**.
- Implizite Garantie, dass sich lediglich Nutzer authentifizieren können, welche auch im Besitz des Authentifizierungsgerätes sind. **lyastani2020fido2**.
- Durch die Nutzung von FIDO2 kann die Usability verbessert werden, da Nutzer sich keine Passwörter mehr merken müssen. Häufig wird das Verwalten der immer höher werdenden Anzahl an Passwörtern als Problem angesehen **lyastani2020fido2 farke2020you**.
- Es wird ein deutlich geringerer kognitiver Aufwand benötigt, da Nutzer keine neuen Passwörter mehr erstellen und merken müssen **lyastani2020fido2**.

- Zum aktuellen Zeitpunkt wird FIDO2 bereits von einer Vielzahl an Browsern unterstützt. Zusätzlich bieten immer mehr Online-Dienste die Möglichkeit an sich mit Hilfe von FIDO2 zu authentifizieren **lyastani2020fido2 farke2020you**.
- Es handelt sich um offene und standardisierte Protokolle **farke2020you**.
- Nachteile:
- Im Falle einer SFA wird der Verlust des Authentifizierungsgerätes als größtes Problem angesehen. Bei Verlust hat auch der Nutzer keinen Zugriff mehr und aktuell gibt es noch keine sichere und effiziente Möglichkeiten, um den Zugriff wiederherzustellen (vor allem ohne Pause) **lyastani2020fido2**.
- Da es sich um zusätzliche Hardware handelt kann diese ebenfalls kaputt gehen **farke2020you**.
- Im Unternehmenskontext, kann die Verwaltung und Verteilung der Authentifizierungsgeräte zu einem Problem werden **farke2020you**.
- Da es sich um Hardware handelt, können Zugänge nicht an vertraute Personen weitergegeben werden, da der Zugang nur mit dem Authentifizierungsgerät möglich ist **lyastani2020fido2**.
- Ohne das Authentifizierungsgerät sind keine spontanen Logins möglich **lyastani2020fido2**.
- Es wird ein physischer Aufwand benötigt, da das Authentifizierungsgerät mitgeführt werden muss **lyastani2020fido2**.
- Bereits das aus der Tasche holen des Authentifizierungsgerätes ist für manche Nutzer bereits eine Hürde **farke2020you**.
- Authentifizierungsgeräte sind häufig mit Kosten verbunden, welche vom Nutzer getragen werden müssen **lyastani2020fido2**.
- Nutzer haben Probleme ein neues Verfahren für die Authentifizierung zu nutzen, da sie sich an das alte Verfahren gewöhnt haben. Das führt dazu, dass Nutzer das neue Verfahren als kompliziert und ungewohnt empfinden. Sie verfügen häufig nicht über das nötige Wissen, um die Funktion und Sicherheit des Verafahrens zu verstehen **lyastani2020fido2**.
- Selbst Nutzern, welchen das Konzept der passwortlosen Authentifizierung gefällt, nutzen häufig weiterhin Passwörter **farke2020you**.

- Nutzer wollen keine Angewohnheiten verändern, wenn die nicht dazu gezwungen sind **farke2020you**.
- Nutzer verwenden lieber Passwörter, da sie das Konzept und die Technologie besser verstehen **lyastani2020fido2**.
- Nicht zwangsweise schneller als die Nutzung von Passwortmanagern **farke2020you**.
- Allgemein fällt das Feeback von Nutzern weniger positiv aus, wenn diese vorher bereits Passwortmanager genutzt haben **farke2020you**.
- Fazit:
- Insgesamt lassen sich noch nicht alle Szenarien mit FIDO2 abdecken. Es gibt noch spezielle Fälle, in welchen die Nutzung von Passwörtern weiterhin notwendig ist **lyastani2020fido2**.
-

2.7 Fido2

- FIDO2 wird von der FIDO und dem World Wide Web Consortium (W3C) entwickelt und bereitgestellt **lyastani2020fido2 farke2020you**.
- Die FIDO Allianz ist eine Organisation mit weltweit über 250 Mitgliedern. Darunter befinden sich Unternehmen wie Google, Microsoft, Apple Amazon, Facebook, Visa und viele mehr **lyastani2020fido2 farke2020you**.
- Ziel ist es Nutzer zu authentifizieren, ohne, dass diese eine Passwort nutzen müssen **morii2017research barbosa2021provable**.
- Basiert auf der Nutzung eines internen oder externen Authentifizierungsgerätes **morii2017research barbosa2021provable**.
- Dabei können Authentifizierungsgeräte, ebenfalls mit einer PIN oder einem biometrischen Merkmal, geschützt werden **farke2020you**.
- Hierbei ist ein PIN allerdings nicht gleichzusetzen mit einem Passwort. Der PIN wird lediglich für das Authentifizierungsgerät genutzt und wird auch nur auf diesem gespeichert **farke2020you barbosa2021provable**.

- Es handelt sich dabei also auch nicht um eine MFA, sondern, um einen einzelnen Faktor, welcher lediglich den Zugriff das Gerät selbst authentifiziert **barbosa2021provable**.
- FIDO2 unterstützt sowohl MFA als auch SFA **lyastani2020fido2 farke2020you**.
- Viele Alternativen zur passwortbasierten AUthentifizierung existieren bereits. Diese werden allerdings nur in einem sehr geringen Ausmaß genutzt **farke2020you**.
- Stellt Zugangsdaten bereit, welche nicht gephisht oder von Datenlecks betroffen sein können **lyastani2020fido2**.
- Das liegt daran, dass keine geteilten Geheimnisse zwischen Nutzer und Dienst existieren, welche auf einem Server gespeichert werden **morii2017research**.
- Wird von fast allen Browsern standardmäßig unterstützt **lyastani2020fido2**.
- Viele verfügbare Authentifizierungsgeräte. Z.B. Security Keys oder auch Smartphones. Beispielsweise Apples Touch ID oder Face ID **lyastani2020fido2**.
- Besteht aus zwei Komponenten: CTAP2 für die Kommunikation zwischen Client und Authentifizierungsgerät und WebAuthn für die Kommunikation zwischen Client und Server **farke2020you**.
- Dabei wird WebAuthn von der W3C spezifiziert und CTAP2 von der FIDO Allianz **farke2020you**.

2.7.1 Webauthn

- WebAuthn ist ein Standard, welcher von dem W3C entwickelt wird. Das Protokoll erlaubt es Webanwendungen Nutzer zu authentifizieren. Dies kann dabei auch über Client-to-Authenticator Protocol 2 (CTAP2) erfolgen **lyastani2020fido2**. ?
- Wurde 2019 ein offizieller Webstandard **farke2020you**.
- Spezifiziert eine standardiserte, vom Browser unabhängige JavaScript API zur Authentifizierung von Nutzern für Webanwendungen. So können Webanwendungen eine Authentifizierung integrieren, welche resistent gegenüber Phishing, Datenlecks und Passwortdiebstahl ist. Anstelle von geteilten Geheimnissen nutzt WebAuthn public-key Kryptographie, um einzigartige

Zugangsdaten für jede Webanwendung zu erstellen, welche nur auf dem Gerät des Nutzers gespeichert werden **farke2020you**.

- Passwortloses Challenge-Response-Verfahren zwischen Client und Server **barbosa2021provable**.
- WebAuthn unterstützt zwei Opertationen: Registrierung und Anmeldung **barbosa2021provable**.
- In der Registrierungsphase sendet der Server dem Authentifizierungsgerät über den CLient eine zufällige Challenge. In dieser Phase signiert das Authentifizierungsgerät mit Hilfe seines privaten Schlüssels die CHallenge und sendet zusätzlich öffentliche Anmeldedaten für zukünftige Anmeldungen an den Server. Meldet sich ein bereits registrierter Nutzer an, wird die Challenge des Servers erneut von dem Authentifizierungsgerät signiert zurück an den Server gesendet. Der Server kann die Signatur mit Hilfe des öffentlichen Schlüssels verifizieren und den Nutzer authentifizieren **barbosa2021provable**.
- Registrierungsphase: Der Server S sendet eine challenge message m_{rch} über den Client C an den Security Key. Diese Challenge beinhaltet eine randomisierte Nonce, Parameter (beispielsweise, ob eine Nutzerverifizierung notwendig ist) und optional einen wert tb, welcher den zugrunde liegenden Kanal eindeutig identifiziert (typischerweise eine Transport Layer Security (TLS) Verbindung). Der Client C erhält die challenge message m_{rch} und wandelt diese in eine command message m_{rcom} und eine client mesasage m_{rcl} um. die command message m_{rcom} wird an den Security Key T übermittelt. Der Security Key T erzeugt öffentlich-privates Schlüsselpaar, welches an den Server S gebunden ist und diesem ermöglicht eine Verifizierung, während der folgenden AUthentifizierungsphase durchzuführen. Zudem gibt der Security Key T eine response message m_{rrsp} aus. Der Client übergibt diese und die client message m_{rcl} an den Server S. Die response message m_{rrsp} beinhaltet einen *attestation type*, welcher es dem Server S ermöglicht eine Verifizierung während der Registrierungsphase durchzuführen und beinhaltet den öffentlichen Schlüssel. WebAuthN 2 unterstützt fünf *attestation types*. Häufig werden die types *None* und *Basic* verwendet. Die restlichen types sind *Self*, *AttCA* und *AnonCA*. **bindel2022fido2**
- Authentifizierungsphase: Der Client empfängt die challenge message m_{ach} von Server S und wandelt diese in eine command message m_{acom} und eine client message m_{acl} um. Die command message m_{acom} wird an den Security

Key T übermittelt. Der Security Key T erzeugt eine response message m_{arsp}, welche mit dem privaten Schlüssel signiert wird und sendet diese an den Server S (über den Client C). Der Server S akzeptiert die response message m_{arsp} und die client message m_{acl} nur, wenn sie sich mit dem dazugehörigen öffentlichen Schlüssel verifizieren lassen. **bindel2022fido2**

- Die Sicherheit von WebAuthn basiert auf dem Beweis, dass RSASSA PKCS1-v1_5 und RSASSA-PSS als Existential Unforgeability under a Chosen Message Attack (EUF-CMA) gelten und der Annahme, dass SHA-256 kollisionsresistent ist **barbosa2021provable**.

2.7.2 CTAP2

- 2018 wurde CTAP2 als internationaler Standard der International Telecommunication Union Telecommunication Standardization Sector (ITU-T) anerkannt **barbosa2021provable**.
- CTAP2 ist ein Protokoll auf der Anwendungsebene, welches für die Kommunikation zwischen eines WebAuthn Clients und eines konformen Authentifizierungsgerätes genutzt wird. Das Authentifizierungsgerät kann dabei ein externes Gerät sein wie beispielsweise ein Security Key, welches über USB, Bluetooth oder NFC eine Verbindung mit dem Client aufbaut. Aber auch ein internes Gerät wie beispielsweise ein Fingerabdruckscanner oder ein Trusted Platform Module können als Authentifizierungsgerät genutzt werden **lyastani2020fido2**.
- CTAP2 spezifiziert, die Kommunikation zwischen einem Authentifizierungsgerät und einem Client. Der Client ist dabei üblicherweise ein Webbrowser. Das Ziel ist es zu garantieren, dass der CLient das AUthentifizierungsgerät nur nutzen darf, wenn der Nutzer dies erlaubt. Dafür muss der Nutzer beispielsweise einen Knopf am Authentifizierungsgerät drücken und/oder sich mit Hilfe eines PINs oder eines biometrischen Merkmals beim AUthentifizierungsgerät authentifizieren **barbosa2021provable**.
- Das Ziel ist es somit einen Client an das AUthentifizierungsgerät zu binden. Ist ein CLient nicht an das authentifizerungsgerät gebunden, kann dieser sich nicht authentifizieren **barbosa2021provable**.
- besteht aus mehreren Phasen. 1. In der Setup Phase initialisiert ein Client C' einen PIN, welcher vom User übergeben wird an den Security Key T.

2. In der Binding Phase tauschen ein Client C (nicht zwangsweise C') und der Security Key T einen gemeinsamen Verbindungsstatus aus, wenn der Client C in der Lage ist, Informationen über die auf dem Security Key T gespeicherte PIN zu liefern. So soll eine einzigartige Verbindung zwischen dem Client C und dem Security Key T hergestellt werden. Schlägt der Client C drei mal in Folge fehl die PIN zu liefern, wird der Security Key T neu gestartet und der Verbindungsstatus wird zurückgesetzt. Schlägt der Client C insgesamt acht mal fehl, wird der Security Key T gesperrt. 3. Ist diese Phase erfolgreich, autorisiert der Client C jeden Befehl, indem eer einen Tag t ausgibt, welcher mit der command message an den Security Key T übermittelt wird. Der Security Key T fährt lediglich fort, wenn eine *positive decision* d des Users vorliegt (beispielsweise einem Knopfdruck) und validiert darauf hin die command message und den Tag t. **bindel2022fido2**

- CTAP2 nutzt unauthentifizierten Diffie-Hellman Schlüsselaustausch **barbosa2021provable**
- Dieser kann von Man In The Middle (MITM) Angriffen betroffen sein **barbosa2021provable**.
- In der Binding Phase sendet das Authentifizierungsgerät dem CLient ein pinToken, welcher beim hochfahren des Authentifizierungsgerätes generiert wird. Dieser pinToken wird lokal auf dem AUthentifizierungsgerät gespeichert und wird von dem verbundenen Client in der Access Channel Phase genutzt, um die nachfolgenden Nachrichten des Clients zu autorisieren **barbosa2021provable**.
- Jedem AUthentifizierungsgerät wird ein pinToken pro hochfahren zugeordnet. Das bedeutet mehrere CLients erezugen mehrere Access Channels mit dem selbem Authentifizierungsgerät und dem selben pinToken **barbosa2021provable**.
- Dadurch wird die Sicherheit von CTAP2 limitiert ??

2.7.2.1 CTAP2.1

- Gilt in Verbindung mit WebAuthn 2 als Post-Quantum (PQ) bereit, da ein Operationsmodus ermöglicht wird, der nur symmetrisch kryptographische Primitive, digitale Signaturen und Key Encapsulation Mechanism (KEM) verwendet **bindel2022fido2**.

- Im Gegensatz zu CTAP2 basiert CTAP2.1 nicht auf unauthentifizierten Diffie-Hellman Schlüsselaustausch, sondern auf einem sogeannten PIN/UV Auth Protocol (puvProtocol), wodurch die PQ-Sicherheit ermöglicht wird **bindel2022fido2**.
- In CTAP2 wird der Verbindungszustand als *pinToken* definiert, welcher aus mehreren 128 Bit-Blöcken besteht und keine maximale Begrenzung der Länge beseitzt. In CTAP2.1 wird der Verbindungszustand als *pinUvAuthToken* definiert welcher eine feste Länge von 128 oder 256 Bit besitzt **bindel2022fido2**.
- Der pinToken von CTAP2 wird bis zum nächsten Neustart wiederverwendet. Der pinUvAuthToken von CTAP2.1 wird nach jeder erfolgreichen Authentifizierung neu generiert. Das führt dazu, dass CTAP2.1 eine Strongly Unforgeable (SUF)-t' Sicherheit aufweist und CTAP2 lediglich eine Unforgeable (UF)-t' Sicherheit **bindel2022fido2**.
- CTAP2 erlaubt es Security Keys und Clients nur den pinUvAuthToken zu teilen, wenn der Nutzer den korrekten PIN eingegeben hat. CTAP2.1 ermöglicht zusätzlich, dass der Nutzer sich mit Hilfe eines biometrischen Merkmals authentifiziert **bindel2022fido2**.

2.7.3 Sicherheit

- FIDO2 ist eine Erweiterung des FIDO U2F Protokolls und bietet die selbe Sicherheit wie public key Kryptographie **lyastani2020fido2**.
- Es handelt sich um geprüfte assymetrische Kryptographie **farke2020you**.
- Es handelt sich dabei um ein Challenge-Response-Verfahren mittels Hardware basierten Authentifizierungsgeräten. Dies bietet einige Vorteile gegenüber passwortbasierten Verfahren. Es gibt keine geteilten Geheimnisse zwischen Usern und Diensten, welche durch Phishing oder Datenlecks kompromittiert werden können. Dabei ist das selbe Authentifizierungsgerät für mehrere Dienste nutzbar, ohne, dass sich dabei eine Verknüpfung zurückführen lässt **lyastani2020fido2 farke2020you**.
- lediglich die Session kann kompromittiert werden **morii2017research**.

- Authentifizierungsgeräte lassen sich mit zusätzlichen PINs oder biometrischen Merkmalen absichern, um sich ebenfalls vor Diebstahl schützen **barbosa2021provable**.
- Unauthentifizierter Diffie-Hellman Schlüsselaustausch könnte durch ein Password Authenticated Key Exchange (PAKE) Verfahren ersetzt werden **barbosa2021provable**.
- Das Paper gibt an, dass dieses sicherer und effizienter sein soll **barbosa2021provable**.
- In folgendem Szenario: 1. Der Nutzer besitzt einen Security Key, welcher mit einem drückbaren Knopf oder ähnlichen ausgestattet ist. 2. Der Security Key ist mit einem geheimen PIN geschützt. 3. Der Nutzer autorisiert vertrauten Clients auf den Security Key zuzugreifen. 4. Der Nutzer verbindet seinen Security Key mit mehreren Clients und nutzt diese um sich bei mehreren Webdiensten zu registrieren/anzumelden. Dann ist versichert, dass: 1. Die Authentifizierung von dem Security Key durchgeführt wurde, welcher die genutzen Zugangsdaten bei dem Webdienst registriert hat. 2. ein autorisierter Befehl auf den Security Key zugegriffen hat. 3. und dieser autorisierte Befehl von einem autorisierten Client beauftragt wurde (sollte der Nutzer den korrekten PIN eingegeben haben). Dies setzt voraus, dass: 1. Der Security Key nicht gestohlen wurde. 2. Der PIN des Security Keys nicht kompromittiert wurde. 3. Der autorisierte Client nicht kompromittiert wurde (korrekte Ausführung von CTAP2 und CLient ist nicht von böswilliger Software betroffen). **barbosa2021provable**.
- Wird ein Security Key gestohlen, kann dieser nur genutzt werden, wenn ebenfalls der PIN bekannt ist **barbosa2021provable**.

3 Umsetzung

3.1 Aktueller Stand der LSY

3.2 Integration eines Yubikeys in die LSY

3.3 User Feedback

- Fragebogen:
- How old are you?
- What is your role within the team?
- Have you ever used a security key before? Yes, and I still do - Yes, but I stopped using it - No - I don't know
- If Yes in which context? (private - work - both)
- Do you currently use a password manager at work? (Yes, I use it all/most of the time - Yes, but I only use it sometimes - No)
- Would you prefer using a security key at work? (Yes - No - I don't know)
- why?
- Do you know how the Fido2 protocol works? (Yes - No - I don't know)
- Would you pay for a security key (about 50€)? (Yes - No - I don't know)
- Do you think security keys are more secure than passwords? (Yes - No - I don't know)
- Comments:

3.4 Zeitmessung

3.5 Nutzung des passwortlosen Verfahrens im privaten Kontext