UNIVERSITY
OF OSLO

Andre Büttner

# Security of Evolving Authentication Technologies

## Multi-Factor Authentication, Passwordless Authentication, and Self-Sovereign Identity

**Thesis submitted for the degree of Philosophiae Doctor**

Department of Informatics
Faculty of Mathematics and Natural Sciences

**2024**

UNIVERSITAS OSLOENSIS
MDCCCXI

*Series of dissertations submitted to the*
*Faculty of Mathematics and Natural Sciences, University of Oslo*
*No. 2798*

ISSN 1501-7710

Cover: UiO.
Print production: Graphic center, University of Oslo.

# Summary

Online services today play a crucial role in people's personal and professional lives. In particular, they handle valuable assets like personal data, financial information, and much more. Thus, access by malicious users can have severe consequences in digital and real life. With identity and access management, online services can restrict access to legitimate users. This entails implementing robust authentication methods to verify the user's identity before providing authorized access. However, online users find themselves exposed to a landscape of many different authentication technologies. While the password is purportedly still the predominant method, new authentication methods are evolving. Therefore, further investigation is required to analyze how these technologies can be used and implemented most effectively in order to protect online users from attackers.

This thesis contributes towards making digital identities more secure by addressing several challenges of some of the most relevant authentication technologies to date, including multi-factor authentication, passwordless authentication, and self-sovereign identity (SSI). It comprises five peer-reviewed and published papers researching three different aspects of user authentication. Firstly, it was examined how users are actually authenticated by online services. For this, a study was conducted analyzing how users configure their authentication settings, what consequences this has on their account security, and their risk of being locked out of their accounts. The findings indicate that the account configurations of many users are not secure and that several users might be locked out of their accounts if they lose their smartphones. In another study, it was observed that account recovery by online services sometimes involves risk-based decision-making, which can increase the difficulty of exploiting recovery for account takeover attacks. The second aspect analyzed is the security of FIDO2 authentication. As the industry standard for security keys and passwordless authentication, there are high expectations for the security of FIDO2. This work addresses potential vulnerabilities of FIDO2 extensions that are relevant in specific contexts, such as online transactions. Furthermore, resilient countermeasures for these extensions that prove effective against manipulation or eavesdropping are proposed. Finally, authentication in the context of data subject rights was analyzed. The General Data Protection Regulation (GDPR) requires data controllers to offer certain rights to the respective data subjects, including access, modification, or deletion of their data. Consequently, service providers must implement appropriate measures for authenticating data subjects to let them exercise their rights. At the same time, they need to ensure that nobody else gains illegitimate access. It turned out that this is particularly challenging for third-party services, where this is implemented in somewhat insecure ways. As this threatens the data subjects' security and privacy, an architecture taking advantage of SSI was proposed to mitigate the risk of attacks exploiting data subject rights implementations.

Beyond the main findings, a trend toward federated credential management is indicated. This suggests that even with seemingly passwordless or SSI-based authentication, the ecosystem of online accounts may become more complex. Eventually, the accounts might even depend again on a weak password. Also, the research on authentication configurations reveals that it is not guaranteed that users can actually exercise their data subject rights, especially in those cases where a high risk for account lockout was discovered. Furthermore, scenarios where users are authenticated implicitly have generally been identified as cases that require more investigation.

# Sammendrag

Nettbaserte tjenester har i dag en viktig rolle i folks private og profesjonelle liv. Slike tjenester håndterer verdifulle verdier som personopplysninger, finansiell informasjon og mye annet. Dermed kan tilgang fra ondsinnede brukere få alvorlige konsekvenser i det digitale og det virkelige liv. Med identitets- og tilgangsstyring kan nettbaserte tjenester begrense tilgangen slik at bare legitime brukere har tilgang. Dette innebærer bruk av robuste autentiseringsmetoder for å verifisere brukerens identitet før autorisert tilgang gis. Det finnes derimot mange ulike autentiseringsteknologier for brukere på internett. Selv om passord fortsatt er den dominerende metoden kommer det stadig nye autentiseringsløsninger. Det er derfor nødvendig med ytterligere undersøkelser for å analysere hvordan disse teknologiene kan brukes og implementeres mest mulig effektivt for å beskytte brukere på internett mot angripere.

Denne avhandlingen bidrar til å gjøre digitale identiteter sikrere gjennom å adressere utfordringer knyttet til noen av de mest relevante autentiseringsteknologiene per dags dato. Dette inkluderer flerfaktorautentisering, passordløs autentisering og selvsuveren identitet (SSI). Avhandlingen består av fem fagfellevurderte og publiserte artikler som undersøker tre ulike aspekter ved brukerautentisering. For det første undersøkes det hvordan brukere faktisk autentiseres av nettbaserte tjenester. I den forbindelse ble det gjennomført en studie som analyserte hvordan brukere konfigurerer autentiseringsinnstillingene sine, hvilke konsekvenser dette har for sikkerheten til brukerkonto og risikoen for å bli utestengt fra kontoene sine. Funnene tyder på at kontokonfigurasjoner for mange brukere ikke er nok sikre, og at flere brukere kan bli låst ute fra kontoene sine hvis de mister smarttelefonen sin. I en annen studie ble det observert at gjenoppretting av kontoer via nettbaserte tjenester noen ganger innebærer risikobaserte beslutninger. Dette kan gjøre det vanskeligere å utnytte gjenopprettingen til kontoovertakelsesangrep. Det andre aspektet som analyseres, er sikkerheten ved FIDO2-autentisering. Som bransjestandard for sikkerhetsnøkler og passordfri autentisering er det høye forventninger til sikkerheten til FIDO2. Dette arbeidet utforsker derfor potensielle sårbarheter ved FIDO2-utvidelser som er relevante i spesifikke sammenhenger. Et eksempel på dette er for digitale transaksjoner. Videre foreslås det robuste mottiltak for disse utvidelsene som viser seg å være effektive mot manipulering eller avlytting. Til slutt analyseres autentisering i sammenheng med de registrertes rettigheter. Personvernforordningen (GDPR) krever at behandlingsansvarlige gir de registrerte visse rettigheter. Dette inkluderer tilgang til, endring av eller sletting av opplysninger. Tjenesteleverandørene må derfor iverksette egnede tiltak for å autentisere de registrerte slik at de kan utøve rettighetene sine. Samtidig må de sørge for at ingen andre får uberettiget tilgang. Det viser seg at dette er spesielt utfordrende for tredjepartstjenester, der dette implementeres gjennom til dels usikre måter. Siden dette truer de registrertes sikkerhet og personvern, foreslås det en arkitektur som utnytter SSI for å redusere risikoen for angrep som utnytter implementeringen av de registrertes rettigheter.

I tillegg til de viktigste funnene observeres en trend mot føderert legitimasjonshåndtering. Dette tyder på at selv med tilsynelatende passordfri eller SSI-basert autentisering kan økosystemet av internettbaserte kontoer bli mer komplekst. Over tid kan kontoene til og med bli avhengige av svake passord. Forskning på autentiseringskonfigurasjoner viser også at det ikke er garantert at brukerne faktisk kan utøve sine registrerte rettigheter. Dette gjelder spesielt i de tilfellene der det ble oppdaget en høy risiko for låsing av konto. Videre har scenarier der brukere autentiseres implisitt, generelt sett blitt identifisert som tilfeller som krever videre forskning.

# Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo. The research presented here was conducted at the University of Oslo under the supervision of Professor Nils Gruschka and Associate Professor Johan Ivar Sæbø. This work was financed as a University scholarship.

The thesis is a collection of five research papers investigating crucial aspects of online user authentication. In particular, evolving authentication technologies are analyzed regarding their usage and implementation, potential security vulnerabilities, and a specific use case. The papers are preceded by four introductory chapters that provide the motivation for this work as well as the technical fundamentals of the different authentication technologies. Furthermore, the papers are related to each other, and their conclusions are summarized.

## Acknowledgements

First of all, I want to express my deepest gratitude to Professor Nils Gruschka. As my principal supervisor, he has always supported me and offered his valuable time to make my journey as a PhD student as pleasant and inspiring as possible. Also, I want to thank my co-supervisor, Associate Professor Johan Ivar Sæbø, for all his support and the insights I gained into research fields even beyond the main scope of this thesis. In addition, I would like to thank all co-authors for their contributions to the research papers of this thesis.

The whole experience would not have been as enjoyable without the excellent work environment within and around the Digital Security Group at the University of Oslo. I especially want to thank all my fellow PhD students, colleagues, and the dear friends I made beyond this for their support and inspiration.

Last but not least, without the love and support of my parents and my sister and their understanding of letting me go to another country, I would have had neither the opportunity nor the stamina to do a PhD.

**Andre Büttner**
Oslo, June 2024

# List of Papers

## Paper I

Andre Büttner and Nils Gruschka. “Evaluating the Influence of Multi-Factor Authentication and Recovery Settings on the Security and Accessibility of User Accounts”. In: *Proceedings of the 10th International Conference on Information Systems Security and Privacy - ICISSP*; ISBN 978-989-758-683-5; pages 691–700; INSTICC, SciTePress (2024). DOI: 10.5220/0012319000003648.

## Paper II

Andre Büttner, Andreas Thue Pedersen, Stephan Wiefling, Nils Gruschka, and Luigi Lo Iacono. “Is it Really You Who Forgot the Password? When Account Recovery Meets Risk-Based Authentication”. In: *Ubiquitous Security. UbiSec 2023*; ISBN 978-981-97-1274-8; pages 401–419; Springer Nature Singapore (2024). DOI: 10.1007/978-981-97-1274-8_26.

## Paper III

Andre Büttner and Nils Gruschka. “Enhancing FIDO Transaction Confirmation with Structured Data Formats”. In: *NISK Norsk informasjonssikkerhetskonferanse*; Issue No. 3 (2021). https://www.ntnu.no/ojs/index.php/nikt/article/view/5506.

## Paper IV

Andre Büttner and Nils Gruschka. “Protecting FIDO Extensions against Man-in-the-Middle Attacks”. In: *Emerging Technologies for Authorization and Authentication. ETAA 2022*; ISBN 978-3-031-25467-3; pages 70–87; Springer Nature Switzerland, Cham (2023). DOI: 10.1007/978-3-031-25467-3_5.

## Paper V

Malte Hansen and Andre Büttner. “Secure and Privacy-Preserving Authentication for Data Subject Rights Enforcement”. In: *Privacy and Identity Management. Sharing in a Digital World*; ISBN 978-3-031-57978-3; pages 175–191; Springer Nature Switzerland, Cham (2024). DOI: 10.1007/978-3-031-57978-3_12.

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

> "Anyone telling you that they have something that can't be hacked is lying or naïve." [41]
>
> R. A. Grimes, *Hacking Multifactor Authentication*

## 1.1 Motivation

In the digitized world that people live in today, it is a matter of course that they have to handle digital identities on a daily basis. They need them to identify themselves when using commercial services, such as online banking, social media, governmental services, and services in enterprise environments. Notably, the COVID-19 pandemic, which coincided with the PhD project reported on here, has undoubtedly boosted the use of online services due to the sudden need for remote work and teaching. As a consequence, people are now even more dependent on their online identities than before the pandemic.

While people enjoy more possibilities through online services, they also increase the attack surface from the physical to the digital world. If a user account is compromised, this can have severe consequences. There are various threat actors—some even backed by their governments and therefore respectively powerful. Taking over a user account is often the first step in enabling the attackers' malicious intentions. Their purposes can range from financial gain over revenge to political interests. Once an attacker gains privileged access, they can cause damage to individuals, organizations, or even entire nations. Therefore, it is crucial that service providers implement reliable and trustworthy systems so account takeovers are prevented in the first place. Before accessing their account, users must authenticate, i.e., prove their identity to the respective service. This makes authentication the most important aspect of account security. Inevitably, the most widespread authentication method is the password. The supposedly first documented use of passwords on a computer was in the 1960s by professor Fernando J. Corbató at the Massachusetts Institute of Technology [21, 59]. Ironically, he himself later raised concerns about the security of passwords after a case where all passwords were accidentally leaked to other users [13]. Further problems like the predictability of user-chosen passwords have been addressed early on, and different countermeasures were proposed [63]. However, this was long before their use in globally distributed networks and the World Wide Web, where passwords suffer from even more vulnerabilities. In particular, phishing attacks have become popular since they require low effort and cost while offering a high reward. In recent years, authentication has undergone a significant evolution, enhancing passwords with modern technologies, such as multi-factor authentication (MFA), passwordless authentication, and the concept of self-sovereign identity.

Despite the efforts to improve user authentication, account takeovers still remain an issue. For example, according to the Identity Theft Resource Center, there has been an increase in hijacked social media accounts by more than 1000% in 2021 [46]. This is concerning because if attackers gain access to social media accounts, they can impersonate their owners, spread false information, or conduct social engineering attacks on their connections. In another case from the end of 2023, several municipalities in Germany have been hit severely by a cyber attack carried out by the hacker group *Akira*. They exploited virtual private network accounts that did not have MFA enabled and were thus easy to compromise [5]. This attack has left the municipalities unable to operate properly for several months, affecting critical public services like issuing ID documents and passports or registering new cars [50]. This shows again the importance of using robust authentication methods and the damage that can be caused otherwise. Also, a hack against email accounts by Microsoft leadership and staff reported in January 2024 has drawn a lot of attention [47]. Given the potential influence of a global player and leading provider of digital infrastructures, one should assume that these accounts are protected with the best effort. In contrast, this incident proves that virtually anybody can be affected by an attack.

The cases mentioned above are just a few examples where improper account protection led to fatal consequences. The fact that they are very recent shows the relevance and urgency of this topic. The development of new authentication technologies gives hope for more secure account protection in the future. This thesis investigates this matter by examining the authentication landscape with a particular focus on evolving authentication technologies and their adoption, which will improve the protection of online users.

## 1.2 Challenges of User Authentication

User authentication is a demanding task for both service providers and users. Although there is a diverse landscape of authentication methods, finding the ideal solution is challenging. One reason for this is that security requirements differ depending on the service. For example, online banking or governmental services are required to provide secure services for their users. This is also true for other services that process particularly sensitive data, such as health data. Commercial services, on the other hand, are more focused on revenue and an increasing number of customers. Therefore, they prioritize convenience over security. Nonetheless, they must satisfy privacy regulations such as the EU General Data Protection Regulation (GDPR) [20] or the U.S. Health Insurance Portability and Accountability Act (HIPAA) [89]. Hence, most online services are legally obliged to protect their users' data appropriately.

Another important aspect to consider is account recovery. Recovery methods allow users to regain access to their accounts if the main authentication credentials are unavailable, e.g., because the password has been forgotten. Yet, account recovery can be a double-edged sword. If the recovery methods are less secure than the authentication methods, attackers can abuse them to bypass authentication [53]. Also, if a recovery factor is lost, recovery may become impossible, which is undesirable if the account is crucial for the user [4].

Furthermore, one has to take into account the user's perspective. Secure authentication methods are only effective if users actually adopt them. Most online services nowadays offer MFA. However, when MFA is optional, users tend to leave it disabled and protect their accounts with a password only. The reluctant use of MFA is possibly linked to convenience as well as the fear of being locked out from the account [14, 15]. It is thus essential to consider both which authentication methods are offered by a service and which are ultimately activated by the users.

Social engineering is another significant concern as it exploits people's trust and emotions. It is a vast field that allows attackers to persuade users to share sensitive information or credentials on-site or remotely through malicious emails, phone calls, or text messages—better known as phishing, vishing, or smishing, respectively [43]. Nobody, including security professionals, is exempt from social engineering threats, as shown in a study from 2018 [73], where even security staff fell for phishing attempts. Technological developments can mitigate this to a certain degree, e.g., with FIDO2 authentication, which is assumed to be phishing resistant (see Section 2.3). Nevertheless, social engineering attacks cannot be entirely prevented by technical means. The human factor must always be taken into account, and users should be educated about the risks of social engineering [6].

Equally important is the implementation behind each authentication method. As with any technology, there may be vulnerabilities in the underlying hardware, communication channels, protocols, or implementations. Therefore, it is essential to test authentication methods properly for any vulnerability and implement countermeasures where necessary. Also, developing clear standards can help provide proper implementation guidelines in order to mitigate potential vulnerabilities.

In summary, account security depends on various factors, including the choice of authentication methods offered by a service, their adoption by the users, and the proper implementation and validation of these methods.

## 1.3 Thesis Goal

As shown in the section above, user authentication is a challenging field with many compelling and highly relevant research opportunities. This thesis aims to investigate the use and impact of authentication technologies on the users' security and contribute to their improvement. In order to achieve this, authentication is investigated from three different angles.

First, it is examined how authentication is actually implemented and used in online services. There has been relatively little research focusing on the adoption of authentication methods. However, this is important, as it determines the risk of a successful account takeover as well as the likelihood of users losing access to their accounts. On the one hand, it is the responsibility of a service provider to implement measures to protect their users. When they offer only weak methods and do not even implement any kind of MFA, their users have only a few possibilities to strengthen their account security. On the other hand, strictly enforcing strong authentication can discourage users from using the service at all. Therefore, online services tend to provide users with several options, thereby pushing the responsibility to them. Furthermore, if several options are enabled, it becomes less trivial to comprehend the consequences of the security and lockout risk of an account. It is thus of particular interest to investigate what authentication methods service providers offer and how users adopt them.

The second angle from which the topic is approached focuses on a specific emerging authentication technology. New authentication technologies must be tested carefully and, if necessary, be improved to ensure they can effectively protect users. Service providers will only implement such technologies if they have been proven to be reliable. Also, users need to be able to trust that technology in order to accept it, especially when it comes to security. This, in turn, also has an impact on whether service providers are willing to offer this method. Therefore, this thesis takes a closer look at FIDO2 authentication and examines if there are any possible weaknesses that need to be addressed.

The third angle is the point of view from a specific use case. The implementation of data subject rights, as required by the GDPR [20], has been identified as non-trivial. No single authentication method or set of authentication methods can be applied in all cases. Third-party data collectors must allow data subjects to access, modify, or remove their data. However, in certain cases, data subjects have not actively registered for this service, so they can not identify themselves as the rightful owner. Traditional authentication methods like passwords can not be applied under these circumstances. Therefore, adequate authentication solutions must be found so data subjects can exercise their rights without the risk of any privacy violation or harm.

## 1.4 Research Questions

The three different angles described in the section above define the overall scope addressed by this thesis. From this, the following research questions are derived:

- **RQ1** How are online users authenticated in practice, and what implications does it have on the security and accessibility of their accounts?

- **RQ2** To what extent is FIDO2 authentication vulnerable to Man-in-the-Middle attacks, and how can this be mitigated?

- **RQ3** How can data subjects be securely authenticated when exercising their data subject rights as required by GDPR?

The first research question (RQ1) examines user authentication on online services from a holistic point of view. One objective is to observe what authentication methods are offered by different online services. Importantly, this includes not only primary and secondary authentication but also account recovery methods. The combination of authentication and recovery methods affects security significantly. If insecure authentication methods are configured, an online account is vulnerable to account takeover attacks. The same applies if an account has weak recovery methods independent of the security of the main authentication. Furthermore, it is crucial that users are not locked out of their online accounts. If a user is not able to provide the requested authentication factors, they will not be able to sign in. The term *accessibility* is used as a metric for the number of options a user has to access an account. In this regard, low accessibility indicates a high risk, and high accessibility a low risk of being locked out of an account. The second objective in this context is to consider what authentication methods are potentially requested when a user attempts to sign in. This can likewise affect the users' security and accessibility. Earlier research has shown that some services implement risk-based authentication (RBA), where different authentication methods are requested depending on client features like the IP address [90]. While this is usually not clearly documented, it is important to be aware of it so it can be applied in the most advantageous way for the users.

The second research question (RQ2) aims to analyze the security of FIDO2 authentication. At the time of writing, FIDO2 is still a young authentication technology. FIDO2 authenticators are expected to be phishing-resistant and are usually offered as a secondary authentication factor in conjunction with passwords. However, the long-term mission of the FIDO Alliance [24], led by big-tech companies like Apple, Google, and Microsoft, is to replace passwords in order to overcome their well-known security weaknesses (see Section 2.2.2). The underlying protocols are somewhat complex and may be prone to possibly unknown attack vectors. Even though the protocol makes use of well-established cryptographic primitives, it is still to be verified if any harm can be done to a user by an active or passive Man-in-the-Middle (MitM) attack.

The third research question (RQ3) addresses a use case where online service providers collecting personal data may have to authenticate data subjects, even if they do not offer user accounts. According to the GDPR, data subjects have several rights regarding their data [20]. However, this can be challenging if a service provider – especially in the case of a third-party data collector – does not implement an identity management system, and the data subject has to be identified solely based on the data. Therefore, the security of current state-of-the-art data subject authentication should be analyzed, and new methods will be proposed.

## 1.5 Approach and Research Methods

Research on user authentication covers many different technical aspects as well as human factors. The research questions formulated in the previous section have been approached using different research methods. Figure 1.1 illustrates how the authentication technologies, research questions, and papers are connected and which research methods were used, respectively. The remainder of this section describes the research methods and how they addressed the research questions. Note that the classification of research methods is not normative and serves only to get an overview of the different applied approaches.

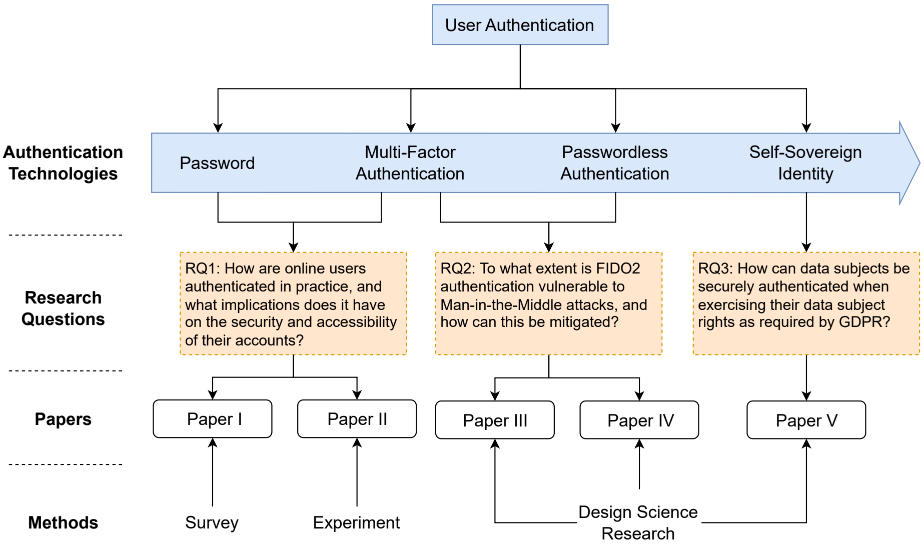

User Authentication
Authentication Technologies
Password
Multi-Factor Authentication
Passwordless Authentication
Self-Sovereign Identity
Research Questions
RQ1: How are online users authenticated in practice, and what implications does it have on the security and accessibility of their accounts?
RQ2: To what extent is FIDO2 authentication vulnerable to Man-in-the-Middle attacks, and how can this be mitigated?
RQ3: How can data subjects be securely authenticated when exercising their data subject rights as required by GDPR?
Papers
Paper I
Paper II
Paper III
Paper IV
Paper V
Methods
Survey
Experiment
Design Science Research

Figure 1.1: Overview of authentication technologies, research questions, papers, and methods.

**Survey** A survey is a methodology that originates in social studies and can be defined as "systematic data collection about a sample drawn from a specified larger population" [79]. The goal is to draw conclusions on the behavior or effects on a specific population. A population can be specified by demographics such as age or country of residence, case-specific criteria, or the entire world population. Typically, a survey is conducted using a questionnaire or interview based on multiple-choice or open questions. When conducting surveys with a specific sample, the sample size is one important factor to determine if the results can be generalized to the entire population. Also, when it comes to hypotheses based on subjective answers, biases resulting from the environment of the study or the phrasing of the survey questions must be considered carefully [77, 85].

While surveys are commonly used in psychology or market studies, they have also become popular in computer science. Generally, technological developments are only useful if they are accepted by their target group. Therefore, survey-based usability studies are conducted to determine how users perceive new technologies and whether they are willing to use them. Specifically in information security, surveys can also help to test awareness about security risks and whether the perceived risk justifies specific security measures from the user's perspective. The survey approach was used to address RQ1. In the study described in Paper I, an online survey was conducted to learn how users configure their online accounts and what devices they need to access them. This was done to learn about the diversity of account configurations that may occur, how this affects the users' security, and the risk of account lockout.

**Experiment** Experiments are an essential approach applied in most scientific fields. An experiment is usually conducted to test the effect of an independent variable on a test group by measuring a dependent variable [49]. For instance, a hypothesis could be as follows: *There is a difference in people's life expectancy depending on how much alcohol they consume.* The independent variable is the amount of alcohol consumption, and the dependent variable is the life expectancy. Its influence can be measured quantitatively using statistical tests or qualitatively by other criteria [36]. Furthermore, experiments can be divided into between-group experiments, where the test subjects are exposed to different conditions, and within-group experiments, where all test subjects undergo the same conditions. It is also possible to combine both approaches in a mixed study design [82].

A within-group experiment is conducted in Paper II to determine if online services apply risk-based decision-making in their account recovery procedures. Essentially, this experiment tested the following hypothesis: *There is a difference in the account recovery procedure depending on specific client features.* This contributed to answering RQ1 by examining whether user account recovery procedures, as implemented by online services, rely not only on the users' account configurations but also on additional client features. The measurement occurred in a qualitative manner by distinguishing whether the account recovery procedure was different for identical account configurations but different client features. The results indicate that some service providers make an effort to make account recovery more challenging for illegitimate users.

**Design Science Research** A method widely established in connection with information systems is design science research [70]. Computer science often investigates a specific software or technology development problem by creating and validating a new concept. Depending on the goal, an architecture, protocol, or a concrete algorithm is developed. Validation can be based on qualitative methods, such as a proof-of-concept implementation or discussion of relevant use cases. Alternatively, quantitative approaches like formal evaluation methods can be applied [80]. Design science research is also prevalent in the cyber security domain as this field deals with software vulnerabilities and mitigation strategies. A significant part of the research is therefore aimed at developing new concepts to improve the security of software systems. Formal methods are very useful, especially when developing new security protocols, as manual analyses are often unable to fully capture the complexity of systems [42].

This methodology was used in particular to address RQ2 and RQ3. Regarding RQ2, Paper III and Paper IV identify extensions as a vulnerable part of the FIDO2 protocols and, thus, propose changes to prevent these extensions from being exploited by an attacker. The former paper includes a theoretical discussion of a proposed change in the format of a specific extension. The latter suggests a cryptographic protocol, which is evaluated using a formal protocol verification tool. In Paper V, which investigates RQ3, an architecture for an authentication scheme is proposed and thoroughly discussed.

## 1.6 Structure of the Thesis

This thesis is written as a cumulative dissertation. Chapter 1 introduces the overall topic, research goals, and methods. In Chapter 2, the technical foundation on identity management and authentication is provided. Chapter 3 summarizes each research paper's most important findings. The results are then concluded in Chapter 4. Finally, the corresponding peer-reviewed and published research papers I to V are attached.

Chapter 2

# Background

## 2.1 Identity and Access Management

People use information systems in their everyday lives, either privately or in the context of organizations or enterprises. It is essential that only legitimate users can access these systems and especially underlying resources to prevent misuse by others and to guarantee people's privacy. For this purpose, *Identity and Access Management* (IAM) plays a crucial role. Specifically, IAM denotes the concept of enabling "the right people or machines to access the right assets at the right time for the right reasons" [35]. To achieve this, users get assigned digital identities that can be associated with specific resources. IAM thereby entails any aspect related to the use and protection of such digital identities.

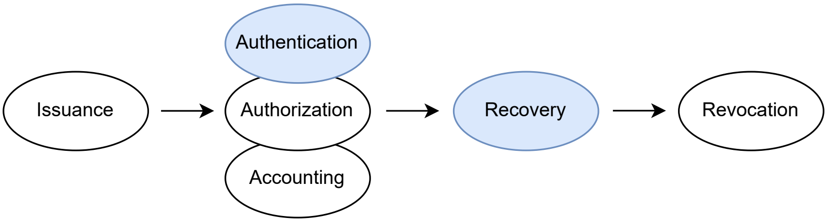


Figure 2.1: Identity lifecycle. Stages relevant to this thesis are highlighted.

An essential part of IAM is the identity lifecycle illustrated in Figure 2.1. It consists of different stages of a digital identity, starting from its creation until its deletion. The first stage in the lifecycle is usually the issuance—sometimes also referred to as registration, provisioning, or enrollment. It denotes the creation of a digital identity for a user and the procedure of binding it to one or several authentication factors. In addition, this may involve the verification of real-world identity attributes, such as name and address, when the digital identity is supposed to represent a mapping onto a real identity [40]. This applies, e.g., to online banking or governmental services.

The step of verifying a user's identity is called authentication. Authentication factors bound to a digital identity allow users to prove the ownership of this identity when interacting with the system. This step aims to protect a system from unwanted intruders, thereby constituting a critical entry point for attackers. After authentication, a service verifies if the user has the right to access the desired resources and grants the corresponding access, which is called authorization. By this, important security principles, such as *separation of privilege* and *least privilege* [78], can be implemented, thereby preventing users from accessing assets of others and minimizing the potential damage of a compromised digital identity. This can be realized per user or, e.g., using role-based access control (RBAC) [22] or attribute-based access control (ABAC) [91] models, where access is based on a user's role or attributes, respectively. Another essential property is accounting, which aims to keep track of whether resources are accessed by legitimate users and to the intended extent. It is used for allocating budgets as well as for auditing, which is crucial for investigating errors and security incidents. Authentication, authorization, and accounting are closely linked as they form the actions performed on an identity during its lifetime. Thus, they are sometimes collectively referred to as the AAA framework [60].

A further important part of the lifecycle is account recovery (or fallback authentication). Users may lose one or several authentication factors and can no longer access their resources. This should be avoided, as the loss of access to their resources can cause financial or other damage to users. Therefore, IAM involves mechanisms for users to regain access to their digital identity.

At the end of the identity lifecycle, a digital identity may expire, or the users may request its deletion, which is referred to as revocation (or deprovisioning). This is relevant when a user does not need the digital identity anymore, but also in case there is a suspicion of compromise.

## 2.2 Authentication

This thesis is particularly concerned with the protection of online users and their digital identities, which requires robust authentication and recovery methods. As explained above, authentication is a subdiscipline of IAM and an essential part of the identity lifecycle. It aims to prevent unallowed access to digital resources and to guarantee the authenticity of a user's actions performed on such resources. As an illustrating example, Figure 2.2 shows a typical scenario where a user is authenticated through username and password. The remainder of this section describes technical details related to online user authentication.

### 2.2.1 Authentication factors

Users can authenticate themselves through a variety of factors. These can be associated with one of the following modalities. *Knowledge-based* factors include anything a user knows or remembers, whereas *ownership-based* factors rely on something a user possesses. *Inherence-based* factors involve anything a user is, particularly biometric features. Some of the most common authentication factors used for online user authentication are described below.

**Password** A password is a text-based secret the user chooses during registration, which later must be entered during every authentication instance. They can also occur in the form of a longer passphrase or a numerical PIN. While the concept is straightforward and well-established, passwords on their own only provide low security. It is, therefore, important that they are chosen wisely

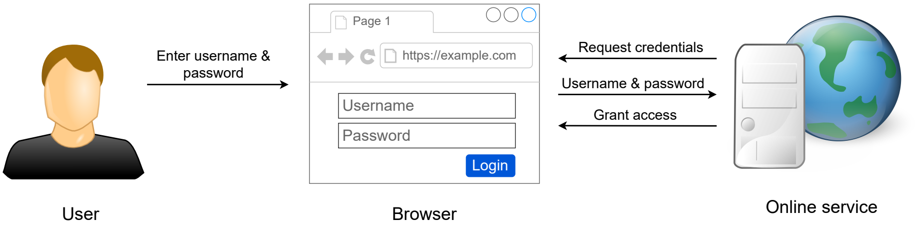


Figure 2.2: Password-based user authentication.

and hard to predict. As passwords are still the most prominent authentication method, their security concerns are elaborated in Section 2.2.2.

**One-Time Password** A one-time password (OTP) is a special kind of password or code that is intended only for a single authentication session. It can be implemented in different ways. For instance, an online service might generate a random code every time a user attempts to sign in. This code is then sent to the user over a trusted channel, e.g., via email or text message on the phone, and subsequently entered by the user for verification. Backup codes are another common type, where a list of codes is stored or printed out by the user. These codes can typically be used as a fallback method in case other authentication factors become inaccessible. Furthermore, OTPs can be based on a shared secret between the online service and a user device used as a seed to generate new codes. This is implemented in authenticator apps like Google Authenticator, Microsoft Authenticator, or Authy. Another typical example is physical bank tokens, devices that generate OTPs for online transactions. There are well-known standards to implement this, such as the HMAC-Based One-Time Password (HOTP) [56], which is based on a shared secret and a counter to generate new OTPs, or the Time-Based One-Time Password (TOTP) [57], which uses the current time instead of a counter.

**Security Key** Security keys are specific devices that need to be plugged in or connected to the client device during the authentication ceremony. When setting up a security key, it shares a public key with the online service. During authentication, the online service generates a random value signed by the security key using the corresponding private key. If the service can verify the signature, the authentication is successful, and the user is logged in. Moreover, security keys often include local authentication mechanisms, such as a PIN or fingerprint, to mitigate the risk in case a device gets stolen. The most relevant standards for this are described in more detail in Section 2.3.

**Biometrics** Biometric authentication takes advantage of human features that differ among individuals. These can include physical features like fingerprints, iris, or the shape of the face, but also behavioral characteristics of users, such as gait, mouse movement, or keystroke dynamics [7, 68]. Biometrics are often used, especially in local contexts, e.g., to unlock a smartphone or laptop. While they can be difficult to forge in the first place, a downside is that some features, like fingerprints, do not change significantly over a whole lifespan. Therefore, they can not easily be revoked. Also, biometrics are globally identifiable and may thus be used to expose a user in other arbitrary contexts. For those reasons, biometric data is considered very sensitive in terms of privacy and should be protected from being leaked [8]. Another challenge is the accuracy of biometric scanners. For high reliability in identifying individuals, they should be highly sensitive. However, such scanners require a certain error tolerance due to noise introduced by the sensor and natural causes affecting a feature, such as a minor injury on the fingerprint. This makes them vulnerable to presentation attacks where an attacker uses a decoy with sufficient similarity [88].

### 2.2.2 Password Security

For several reasons, passwords can provide only low security guarantees in practice. A common problem is the users' choice of passwords. Some of them use a password that is based on anything related to their personal life, like the name of their spouse, child, or pet. Such information can usually be obtained easily through a web search or social media. Furthermore, recent statistics examining the most common passwords suggest that many people still use something that is easy to remember or default passwords, such as *123456* or *admin* [58, 65]. Also, people tend to reuse the same password for different services. Every once in a while, online services get breached by attackers, and password databases are leaked. It gets challenging for attackers when the passwords are stored using hash functions[1]. But even in this case, they can recover many of the hashed passwords using brute-forcing, dictionary attacks, or rainbow tables. As a consequence, attackers manage to breach into accounts of other services where the same username and password combination is used [48]. To make this more challenging for attackers, online services often implement password policies, such as requiring a minimal length and the use of special characters, to make a user choose more secure passwords [86]. Moreover, password managers have been introduced as a tool for end users. They can store their passwords on one or multiple devices, encouraging them to use less memorizable passwords and different passwords for each service. Furthermore, they can generate random passwords with high entropy and provide autofill functionality. This can significantly increase the difficulty of those attacks mentioned above.

One of the biggest concerns nowadays is the use of phishing attacks. By this, attackers lure their victims into opening a website that pretends to be a specific

[1] Hash functions are one-way functions that transform input into a seemingly random output of a fixed size. A key characteristic is that they are meant to be irreversible. Therefore, the original input can only be obtained by testing different input values and comparing if the output matches. A random value called *salt* is usually concatenated to the input in order to avoid the same output value for equal inputs in different contexts, e.g., if several users use the same password. Therefore, the stored value is always different, thus increasing the required effort for an attacker [63].

online service. This is usually done by sending emails or text messages to victims where they are asked to click on a link that leads to a malicious website. An attacker can thereby easily target millions of users with low effort and costs. If only a few of those users end up clicking on the link without any suspicion, the attack will likely be successful. The affected victims end up on a malicious website, entering their password, which is subsequently stored in cleartext by the server and thus disclosed to the attacker. This finally allows the attacker to sign in to the actual accounts of their victims. The problem with phishing attacks is that they are challenging to prevent entirely. It is thus vital that users are aware of this threat so they do not click on suspicious links [2].

### 2.2.3 Multi-Factor Authentication

Due to the security problems with passwords, many online services offer their users to configure one or more additional authentication factors. If only one factor is requested, this is called single-factor authentication (SFA). Otherwise, if several factors are required, this is usually referred to as multi-factor authentication (MFA). Note that it can also be called two-factor authentication (2FA) if exactly two authentication methods are required. Most online services offer the password as the primary authentication factor and OTP-based approaches like text messages or an authenticator app as the secondary factor. Using MFA increases the reliability of the user's identity but at the cost of usability. This is because users need to enroll and sign in with multiple authentication factors, which requires more effort from the user. Furthermore, they need to avoid losing any of the configured authentication factors, as the user would, in many cases, no longer be able to access their account [4, 75]. Some services relax the requirements for the MFA method and allow users to deactivate it for the current browser through a checkbox on the login page. If this option is selected and after a successful login (with the MFA factor), they will only need their password for future logins from the same browser.

### 2.2.4 Risk-Based Authentication

There are also online services that collect additional information about a user's client setup, like the IP address, device type, or user agent. This information can be compared with the user's login history to determine if a login attempt originates from an unusual client. For that purpose, a risk score can be calculated using traditional statistical approaches [33] or machine learning [61, 71]. If the score is above a threshold, this indicates a *suspicious* client, and therefore, additional authentication factors can be requested. In high-risk cases, the online service may even reject the login attempt entirely. This methodology is called risk-based authentication (RBA) [90].

If used properly, this can be advantageous for usability and security. The authentication procedure may require fewer or easier factors by legitimate users. In contrast, other users are exposed to a more challenging set of authentication factors. Yet, there are also situations where even legitimate users want to access their accounts from a different device or location. Typical scenarios are when a user is traveling or switching devices. Also, many client features can easily be spoofed and may thus not present a significant barrier for an attacker. Online service providers need to consider this when selecting the client features and risk score algorithm in their RBA implementation.

### 2.2.5 Single Sign-On

In a federated identity architecture, an online service, or relying party (RP), outsources identity management to a trusted third party, the identity provider (IdP). This IdP authenticates users and provides an RP with respective identity attributes. Online services may use such an IdP in addition to or instead of implementing their own IAM. This feature is called *single sign-on* (SSO) [69] and essentially allows users to reuse their account at one service to identify themselves to several other services. SSO can be implemented using standard protocols such as SAML [66], OAuth2.0 [45], or OpenID Connect [76]. Enterprises and organizations often use this to allow employees to access their work-related services through one corporate account.

### 2.2.6 Account Recovery

Another vital part of user authentication is account recovery. While this is often regarded separately, it is closely related to the main authentication. From a user's perspective, account recovery is a mechanism to be used in case it is not possible to access an account, e.g., because the password or another required authentication factor was lost. Account recovery can usually be initiated by clicking on a link saying something like "Password forgotten". This is often followed by entering an email address, phone number, or username. Finally, the user receives a link or OTP code to regain access to the account.

Recent research has investigated account recovery specifically for accounts configured with MFA. It was found that recovery can be more challenging for the user, and there is a higher chance of losing access to an account. Many services offer recovery codes that need to be stored or printed out by the user. However, the user might ignore that, lose the codes, or forget where they were stored. Also, there is a risk of weak account recovery mechanisms being exploited to bypass strong main authentication factors [4, 37].

## 2.3 FIDO Protocols

The *Fast IDentity Online* (FIDO) alliance has developed industry standards for authentication with security keys. Their goal is to address the weaknesses of passwords by offering passwordless and, in particular, phishing-resistant authentication solutions. This section describes the history of the FIDO protocols as well as some details on the most recent FIDO2 standard and passkeys.

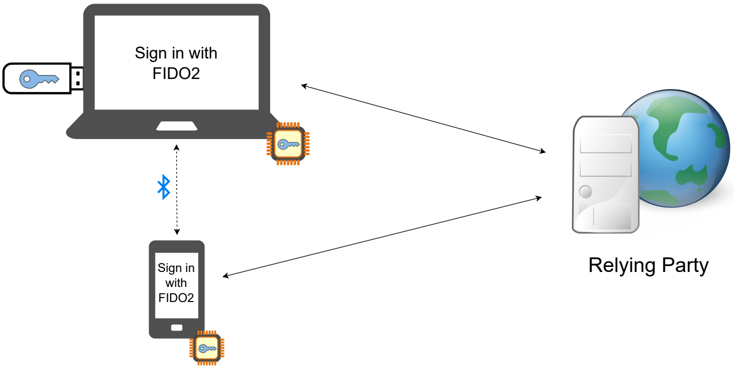


Figure 2.3: FIDO2 authentication with platform and roaming authenticators.

### 2.3.1 History of FIDO

The Fast IDentity Online (FIDO) Alliance was founded in 2012 by six companies, including well-known vendors like PayPal and Lenovo. From the beginning, their goal was to develop a passwordless authentication solution based on public-key cryptography [30]. In the following years, the alliance was joined by many more companies. In April 2024, the FIDO Alliance already comprised 45 board-level, 67 sponsor-level, and ten government-level members, among them big tech companies such as Google, Apple, and Microsoft [26].

The first protocols developed by the alliance were the FIDO Universal Authentication Framework (UAF) [28] and FIDO Universal 2nd Factor (U2F)[2] [31], published at the end of 2014. FIDO U2F was proposed to enhance passwords with a more secure secondary authentication factor based on a device, e.g., a security key. FIDO UAF is a protocol that enables passwordless authentication by using local device authentication mechanisms like a PIN or face recognition [29, 32].

This was followed by the successor FIDO2, officially launched in 2018 [25], providing a more generic solution that allows different devices to be used as FIDO2 authenticators in MFA and passwordless scenarios. It is composed of two protocol specifications: the W3C WebAuthn standard [51] and the Client to Authenticator Protocol 2 (CTAP2) [23]. FIDO2 has since been adopted by all major browsers. When used for passwordless authentication, it is now predominantly called *passkeys* [38].

### 2.3.2 FIDO2

FIDO2 is an authentication method where a user possesses a cryptographic key in the form of an authenticator device. This key is registered with an online service. Similar to SSO (see Section 2.2.5), this service is usually called RP. The user needs to prove the ownership of the respective key in order to sign in to this

[2] FIDO U2F was later renamed Client to Authenticator Protocol 1 (CTAP1).

RP. As shown in Figure 2.3, there are two different categories of such devices: *platform* and *roaming* authenticators. Platform authenticators are integrated into a client device that is interacting with the online service, e.g., a computer or smartphone. Here, access to the credentials is restricted and requires some form of local authentication, such as Windows Hello or Apple's Face ID. In contrast, roaming devices are separated from the actual client device. These are specific authenticator devices connected to the client via Universal Serial Bus (USB), Bluetooth, or Near Field Communication (NFC). Common examples of roaming authenticators are hardware keys like YubiKey or Google Titan. Notably, a smartphone can also serve as a roaming authenticator when used in conjunction with a computer client.

Authentication with FIDO2 is conducted through a public-key-based challenge-response protocol. An authenticator registers with an RP by creating a new credential, including a public-private key pair, and sending the public key to the RP. During authentication, the RP first creates a random challenge value. The authenticator responds with a signature created using the respective private key for this RP. The signature includes several parameters, such as the challenge value, RP identifier, and web origin. The RP finally verifies the signature and confirms (or denies) the authentication procedure.

As mentioned previously, FIDO2 comprises two distinct protocols: WebAuthn and CTAP2. The W3C WebAuthn standard [51] specifies a JavaScript API that enables browsers and platforms to support FIDO2 authentication [32]. It defines functions and parameters for both registration and authentication on the client side. Furthermore, it describes how and what parameters need to be created and processed by the RP respectively. In addition, WebAuthn requires the browser only to process registration and authentication function calls in the so-called *secure context*. This means, in particular, that it only processes WebAuthn calls if the website content was retrieved over HTTPS. Furthermore, it makes sure that the *origin* parameter included in registration and authentication calls matches the domain name of the website that is processed in the same browser context. This should ensure that the WebAuthn message cannot be tampered with during transit between the server and the client. Furthermore, this helps to avoid a malicious website changing the origin parameter in order to obtain a valid assertion for another domain name. This is a crucial step in making FIDO2 phishing-resistant.

CTAP2 is a standard for the communication between a client device and roaming authenticators. It specifies the implementation of the interfaces for the different transport protocols, i.e., USB, Bluetooth, and NFC. Furthermore, it defines the messages that are exchanged between the client and the authenticator. These messages are specified in the Concise Binary Object Representation (CBOR) format [12]. Many parameters used in WebAuthn are simply converted into CBOR format and forwarded to the authenticator. In addition, CTAP2 offers further features, e.g., local PIN-based authentication of the user and other authenticator extensions [23].

### 2.3.3 Passkeys

FIDO2 authentication was originally meant to follow the principle that the secret, i.e., the private key, never leaves an authenticator device, thus being protected from remote attacks. Despite its security promises, it was adopted somewhat reluctantly by online services and especially by users. It is challenging to convince users to use new authentication methods compared to more familiar ones, such as passwords. They are especially concerned about the risk of losing the authenticator and, thus, access to an account [55, 67]. Also, FIDO2 was usually only implemented as a second authentication factor, which did not satisfy the FIDO Alliance's main objective of a passwordless solution. As a result, passkeys were introduced in 2022 [38], which are essentially FIDO2 credentials that can be synchronized between different user devices. Passkeys are supposed to ensure that the credentials are not entirely lost when one FIDO2 authenticator is lost. According to *Passkeys.directory*[3], more than 120 vendors were supporting passkeys in April 2024, including several major service providers.

FIDO2 credentials that are only stored on a single device can make use of secure hardware that protects the private keys even from local attackers. Passkeys, in contrast, can only be stored in ordinary memory because the private key must be accessible so it can be transmitted to other devices. This may lead to the assumption that passkey credentials are vulnerable to being read from the device's memory by malware. In addition, passkeys involve an additional entity that provides passkey management. These are referred to as *passkey providers*. Their main task is to synchronize passkey credentials between different devices. The robustness of passkey protection consequently also depends on the methods used by passkey providers to transmit the credentials between the devices, e.g., if they are transmitted using end-to-end encryption. Furthermore, a critical property is whether passkeys are only stored offline or if the passkey provider's server additionally backs them up.

## 2.4 Self-Sovereign Identity

Self-sovereign identity (SSI) is a new form of identity management that aims to let users keep all control over their identifiable attributes, as opposed to the federated identity model (see Section 2.2.5), where a central third-party component is governing the identities. SSI can be realized by a decentralized architecture in which attribute issuance, storage, and verification are separated from each other, as depicted in Figure 2.4. Users can request credentials from an identity issuer, store and manage them locally, and present them to an online service. Essentially, any service can verify attributes created and signed by a trusted issuer. This has the advantage that the issuer does not learn which services the users access, thus preserving their privacy [74].

[3] https://passkeys.directory (Last accessed: 2024-04-15)

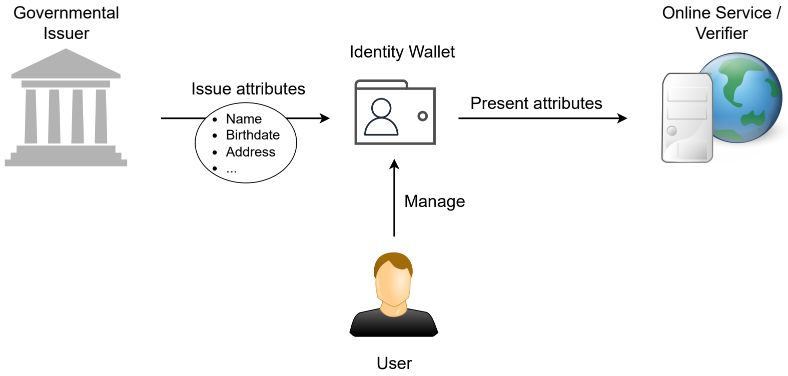


Figure 2.4: Self-sovereign identity with governmental attribute issuer.

### 2.4.1 Components

Various technologies are being considered for the implementation of an SSI that enables decentralized identity management. A key component is the so-called verifiable credentials (VCs), which intend to hold attributes that can be issued and verified by different entities. The architecture of the different components and their interaction are defined in a W3C standard [87]. Moreover, VCs make use of decentralized identifiers (DID), another W3C standard that specifies the implementation of identifiers that do not rely on a central provider. DIDs allow binding a subject, e.g., a person, organization, or any other arbitrary type of entity, to a document that represents data or attributes associated with this subject [84]. Distributed ledgers, such as blockchains, are a relevant technology in this context as well. Their key characteristic is to provide decentralized architectures with strong integrity properties. While mostly known for use cases like cryptocurrencies and smart contracts, they are also considered for serving as a decentralized repository for storing DIDs. This enables transparency in the issuance (and revocation) of DIDs and allows anybody to verify DIDs and VCs independent of a central trust entity [64].

### 2.4.2 EU Digital Identity Wallet

There is an endeavor to offer electronic identification (eID) for governmental services within and across EU countries [16]. The first EU electronic identification and trust services (eIDAS) regulation was based on a federated identity model [19]. In their amended regulation from 2021, also referred to as *eIDAS 2.0*, the European Commission has instead acknowledged the use of decentralized technologies. Specifically, they consider using the EU Digital Identity Wallet [17], which follows a decentralized paradigm and involves those technologies described above [18]. At this point, however, this is still a work in progress, and most countries have not implemented it yet.

Chapter 3

# Contributions

## 3.1 Summary of Research Papers

This thesis includes five research papers that address the given research questions (see Section 1.4). The author of this thesis has been the lead author of papers I to IV and has contributed significantly to the respective theoretical parts as well as to the experiments and analyses. In Paper V, he served as co-author, and his and the lead author's contributions were evenly distributed. The remainder of this section summarizes each research paper and puts them in the context of the respective research question.

### Paper I: Evaluating the Influence of Multi-Factor Authentication and Recovery Settings on the Security and Accessibility of User Accounts

Paper I investigates the use of authentication and recovery methods by users of two of the most prominent online vendors, namely Apple and Google. The main objective was to analyze respective user accounts by evaluating their security and accessibility. First, a survey was conducted where Apple and Google users provided information about their user accounts. They had to specify, e.g., how they could access their password, which authentication methods they had enabled, and what devices (e.g., computers and smartphones) were required. After that, a method to model authentication settings and account dependencies was applied, which is denoted as *account access graphs* (AAGs) [44, 72]. Based on the survey responses, AAGs were created for each test participant. These were subsequently used to evaluate the user accounts by applying specific scoring schemes. The security was assessed in a qualitative manner by assigning the values *low*, *medium*, and *high* to the different authentication methods and deriving an overall score for each account. Furthermore, an accessibility scoring scheme was used to indicate the minimal number of devices or access methods that must be lost to risk being locked out of their accounts.

The results show that a majority of Google accounts depend on their email accounts as a recovery factor. This was considered insecure because email accounts are often weakly protected, and as the service provider, Google can not ensure the security of those accounts. The user accounts of the Apple test participants were mostly regarded as medium in terms of security, as they usually had at least SMS enabled as the MFA method. Beyond this, it was noticeable that a considerable number of Apple and Google accounts depended entirely on the availability of the user's phone. Hence, this indicates that the users might be locked out of their accounts when they lose their phone.

This study contributes to answering RQ1 as it provides insights into how widely the account configurations of actual users in practice vary and their effect on the users' security and accessibility. Apple and Google differ in the authentication and recovery options they offer. Apple mainly focuses on binding users' devices to their accounts, while Google lets users choose between several different MFA and recovery methods. Compared to Google, more Apple user accounts performed better in terms of security but worse regarding accessibility. The results for Apple somewhat show that it has a positive effect on the users' security when the options to set up authentication are rather limited and MFA is enforced. Google, on the other hand, gives their users a lot more freedom, possibly leading to naive account configurations. The study further proves that account accessibility is not trivial. If multiple MFA alternatives and recovery methods are enabled, they may often be linked to the same device. With the use of on-device password managers, even the password may eventually be stored on the same device.

### Paper II: Is it Really You Who Forgot the Password? When Account Recovery Meets Risk-Based Authentication

In Paper II, the focus lies on the account recovery of online accounts. Previous research showed that many online services apply RBA (see Section 2.2.4). Inspired by this, the paper introduces the concept of *risk-based account recovery* (RBAR). With this approach, an online service may process similar client features in its account recovery as during authentication, such as the IP address, device information, or others, to determine if the values of those features vary significantly from previous logins. By this, a service can derive a risk score and use it to decide whether a user needs to provide additional authentication factors when recovering their account or to block the user entirely.

With this in mind, two experiments were performed to test the extent of RBAR use by actual online services. The first experiment was conducted in an ad-hoc manner with Google accounts. Here, it was observed that Google offered different account recovery procedures when using different IP addresses or browser configurations. The recovery procedures turned out rather unpredictable. Sometimes, Google would ask for custom security questions or request pre-configured MFA methods. The second experiment was conducted with four other online services, including Amazon, Dropbox, GOG, and LinkedIn. This time, the experiment was performed systematically by observing the difference between account recoveries from a previously used browser and from a Tor browser. The only difference discovered here was that Amazon occasionally and LinkedIn always requested a CAPTCHA[1] from the user when performing account recovery from the Tor browser. The results of both experiments were finally summarized in a maturity model to classify the strength of the RBAR challenges.

A significant concern pointed out in this paper is the use of CAPTCHA puzzles. Even if they can mitigate automated attacks, they are ineffective against targeted

[1] A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [39] is a feature used by online services to prevent automated agents from interacting with a website. This is often implemented by interactive puzzle games, which are typically hard to automate, or by human behavior detection.

attacks. Moreover, researchers and attackers frequently discover ways to bypass CAPTCHA algorithms. Online services that distinguish between legitimate and suspicious users during authentication or recovery should, therefore, implement more effective measures like Google.

The paper addresses RQ1 mainly with regard to measures implemented by the service provider. In particular, it is revealed that authentication and recovery procedures are sometimes determined not only by the users' account configurations but also by a risk assessment conducted by the service provider. This makes it more difficult for an attacker to take over the account, increasing the legitimate user's security. With RBA and RBAR this is true for both authentication and recovery. RBAR may be particularly beneficial when otherwise account recovery is relatively easy, e.g., if access to an email account is required only. Nonetheless, this comes with the downside of requiring more recovery factors by a legitimate user trying to access an account from an unusual client device or location, which negatively affects the accessibility. Besides, there is research on the accessibility of CAPTCHAs showing that they might be challenging to people with, e.g., visual or auditory impairment [3, 62, 81]. Consequently, while CAPTCHAs do not significantly improve security, they can instead present an obstacle to legitimate users.

### Paper III: Enhancing FIDO Transaction Confirmation with Structured Data Formats

FIDO2 authentication is primarily meant to be used in standard authentication scenarios, i.e., when signing in to an online user account. Nevertheless, it is also considered to be used in other contexts, including banking transactions. For this, the *transaction confirmation* extension [27] was specified.

Paper III addresses the potential risks of using this extension. In particular, it points out the possible violation of the *what-you-see-is-what-you-sign*[2] paradigm. The signed transaction information is a generic text string, which makes it susceptible to ambiguous formulations or homograph attacks [34] where similar-looking or invisible symbols replace numbers or characters to obfuscate users. Based on prior literature and the fact that extensions are not cryptographically protected, it can be assumed that an attacker may manipulate them in transit. As a countermeasure, the use of structured data formats is proposed to mitigate the risk of ambiguities. By enforcing a machine-readable format, transaction details could be validated against certain policies, and the information presented to the user could be normalized, representing the actual meaning of a transaction to be made.

This paper investigates a specific case in which FIDO2 is possibly vulnerable to a MitM attack and thus addresses RQ2. It shows that the security of FIDO2 authentication in more complex cases, such as transactions, is not necessarily

[2] What-you-see-is-what-you-sign (WYSIWYS) [52] designates the property of a document to be signed of being self-describing and unambiguous. This is particularly important when approving banking transactions or legal contracts, which shall be confirmed with a legally binding signature.

sufficient. Given the high expectations of it as a replacement for passwords, particularly in sensitive use cases like banking transactions, the risk of ambiguous or manipulated transaction texts should be considered carefully. From this specific use case, Paper IV was inspired where the problem is addressed more broadly.

### Paper IV: Protecting FIDO Extensions against Man-in-the-Middle Attacks

In contrast to the previous paper, Paper IV looks at the security of FIDO2 protocol extensions in general. It was found that extensions are neither end-to-end encrypted nor authenticated between the relying party and the authenticator device. FIDO2 messages are, however, exposed to a large attack surface, as they have to pass through different intermediary components. An attacker may, therefore, be able to eavesdrop or manipulate protocol extensions to their advantage. This could eventually make a user confirm an unintended extension or lead to the disclosure of sensitive information.

This paper first presents a theoretical discussion of the attack service and identifies the following four different points where attackers may intercept FIDO2 messages in cleartext:

- Web intermediaries
- Malicious browser/client
- Malware on the client
- Malicious device between client and authenticator

In order to mitigate the risk of attacks, a protocol is proposed that can be applied to sensitive FIDO2 extensions to add authenticated encryption between the relying party and the authenticator. The cryptographic protocol verifier tool *ProVerif* [10] has been used to test and verify its effectiveness against active and passive MitM attacks. Furthermore, a proof-of-concept implementation is provided to demonstrate how it can be applied in practice, along with a performance evaluation, showing that only a neglectable delay would be introduced with the suggested approach.

This research uncovers vulnerabilities allowing attackers to intercept FIDO2 messages and provides a solution to prevent such attacks, thereby addressing RQ2. It is assumed that basic FIDO2 messages are not vulnerable to MitM attacks as they do not contain any valuable or identifiable information. Furthermore, changes are eventually noticed during the signature verification by the relying party. However, FIDO2 extensions can pose a risk when used for arbitrary purposes or if they contain sensitive information. The proposed protocol mitigates such risks and provides a solution that is compatible with the FIDO2 standard.

### Paper V: Secure and Privacy-Preserving Authentication for Data Subject Rights Enforcement

Online services collect a lot of data about users, including personal data. This data gets increasingly valuable and may be sold to or shared with third-party services, e.g., for targeted advertisement. According to the GDPR, services collecting data within the EU must respect certain *data subject rights*, such as the right of access, the right to be forgotten, and more [20]. This requirement ensures that data subjects can always influence how and to what extent their data is processed. As a consequence, data controllers must implement technical and administrative measures to enable data subjects to exercise their data subject rights. This also implies that data controllers must be able to identify and authenticate data subjects reliably so that only the rightful owners can access (or modify) their data.

Paper V investigates what approaches are commonly used to authenticate data subjects, particularly in the case of third-party data collectors. Prior studies revealed that often, an email address or phone number is used to verify the identity of a data subject. It was further shown that the verification procedures were prone to improper validation and could be bypassed by an attacker. This paper proposes a generic authentication solution based on attribute-based credentials and the self-sovereign identity (see Section 2.4) as a countermeasure. Specifically, the EU electronic identity (eID) is considered, which enables the verification of specific attributes of a data subject that a service provider can match with the data in question. For this, an architecture is presented describing which components are required and how they interact. Furthermore, two scenarios are discussed. The first one is an SSI-based case, where a service provider can take care of matching relevant attributes and only requires an identity provider to serve as an attribute verifier. In the second case, the matching of specific attributes is outsourced to an identity provider to cover scenarios where a service provider does not have sufficient attributes to verify a data subject's identity.

With respect to RQ3, the literature review results suggest that the methods used in practice to authenticate data subjects are unreliable and pose a risk to the rightful owners of the data. Therefore, this paper describes an SSI-based approach to authenticate data subjects to third-party data collectors. This provides a method for data collectors without any other identity management implementation in place to verify the identity of the respective data owners before giving them access to their data.

## 3.2 Further Findings

The research papers described in the previous section were initially aimed at answering one of the research questions RQ1-RQ3. However, the research carried out allows conclusions to be drawn that go beyond this. Figure 3.1 illustrates how papers I, II, and V provide additional knowledge for the research questions and authentication technologies.

### 3.2.1 Federated Credential Management

In general, Paper I and Paper II can be considered relevant for any authentication technology where some form of account federation occurs. In fact, it appears

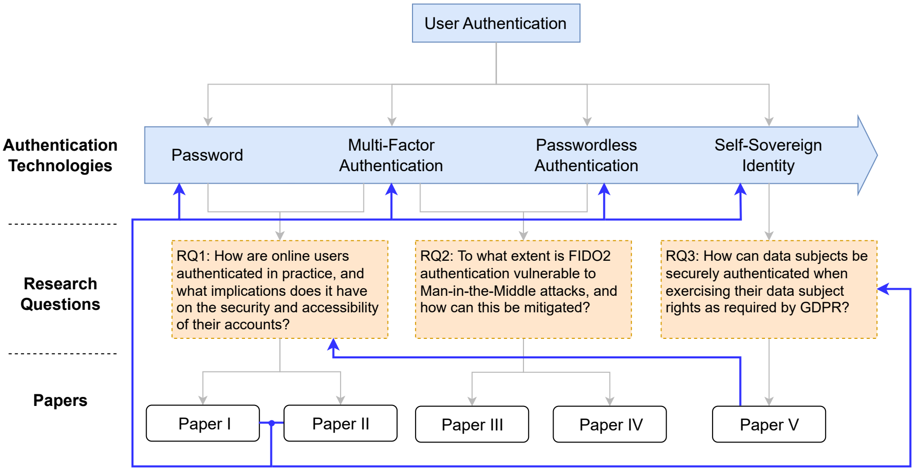

User Authentication
Authentication Technologies
Password
Multi-Factor Authentication
Passwordless Authentication
Self-Sovereign Identity
Research Questions
RQ1: How are online users authenticated in practice, and what implications does it have on the security and accessibility of their accounts?
RQ2: To what extent is FIDO2 authentication vulnerable to Man-in-the-Middle attacks, and how can this be mitigated?
RQ3: How can data subjects be securely authenticated when exercising their data subject rights as required by GDPR?
Papers
Paper I
Paper II
Paper III
Paper IV
Paper V

Figure 3.1: Additional impact of the papers on the different research questions and authentication technologies.

there is almost always some identity provider behind any kind of authentication method. This is illustrated in Figure 3.2. The most obvious example is SSO, where the account for an identity provider grants direct access to another online service (see Section 2.2.5). This can also apply to other authentication methods where this is less apparent. For instance, passwords can be stored and accessed through a password manager, which helps users to choose more secure passwords by generating them automatically and storing them offline or online. Devices and browsers often also provide password storage and synchronization across devices, thus serving as online password managers. Passwords stored online are consequently accessible by signing in to the respective password manager account.

OTPs provided by authenticator apps like Google Authenticator or Authy may also be synchronized across different devices. Therefore, such secondary authentication factors are linked to another account and may be accessible even without the device on which they were generated. This is true for passwordless authentication as well, specifically FIDO2 passkeys (see Section 2.3.3). In fact, the idea behind Paper I was originally based on the introduction of passkeys, which are meant to be backed up by a passkey provider. It was assumed that passkey credentials, including the private key, would be accessible through a passkey provider account similar to a password manager, which is also why password managers such as 1Password and Bitwarden later started offering passkey management as third-party providers [1, 9]. When this research was conducted, passkeys were yet at an early stage, and it was unclear how this would eventually be implemented. By then, FIDO2 was still only used as a secondary authentication method. Therefore, it was decided to first focus on online accounts in general. Since Apple and Google are now passkey providers,

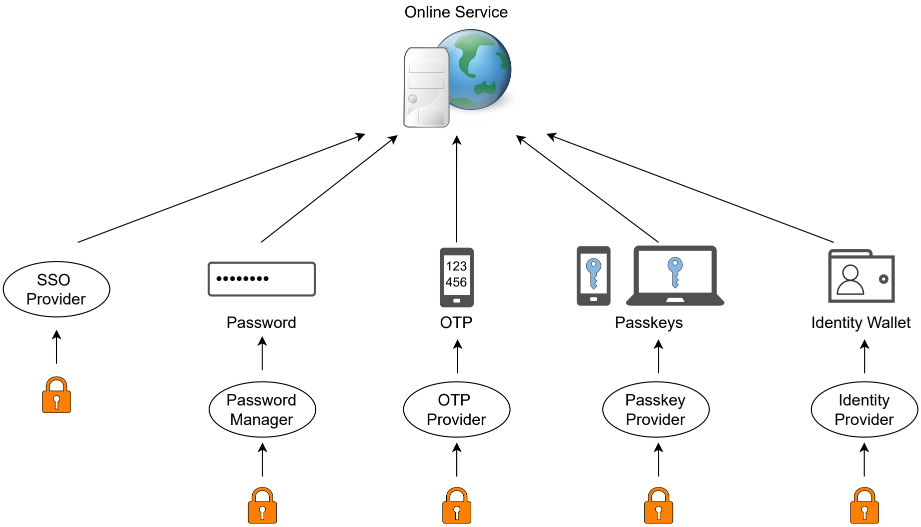


Figure 3.2: Federated credential management.

accessing them is crucial to having access to their respective credentials. The results regarding accessibility can consequently apply to passkeys in that a high risk of losing access to an Apple or Google account also poses a significant risk of losing the corresponding passkey credentials. Since there were several users with a low accessibility score, they could lose their passkeys even if they were backed up in their Apple or Google accounts. The security of a passkey provider account might also indicate if there is a high risk of these credentials being stolen. If, for instance, only a weak password is used, an attacker might have a chance to take over the passkey provider account and access the corresponding credentials. Since account recovery also needs to be considered when it comes to authentication, implications from Paper II are equally relevant. However, passkeys—and also passwords stored by a password manager—are often encrypted with some form of local authentication. Thus, even if an attacker can take over an account of a passkey provider, they do not necessarily have immediate access to the passkey credentials. Yet, account access can facilitate this, and credentials may be unlocked by brute-forcing local protection mechanisms like a PIN. There is no standard for a passkey provider's credential management, so a wide variety of implementations is to be expected.

Similar assumptions can be made when it comes to SSI-based authentication. Identity wallets storing user attributes may be cloud-based and thus be linked to an online account. Hence, it is equally important to consider how this wallet can be accessed, i.e., what authentication and recovery options are offered by the wallet provider and ultimately enabled by the user. The federated approach that is used for these different authentication technologies makes it increasingly hard to comprehend the overall implications on a user's account security and the risk of being locked out. Therefore, account access graphs, as used in Paper I, can be especially useful to comprehend the overall authentication setup for an account.

### 3.2.2 Data Subject Rights Authentication

Paper V focuses on services that do not actively provide identity management for the data subject. However, services that offer user management collect personal data as well. Apple and Google, in particular, are known to collect a lot of personal data and are consequently required to adhere to the GDPR, i.e., to implement data subject rights. If user accounts are weakly protected, their privacy is at risk, and attackers may be able to access the data of other data subjects unlawfully. At the same time, losing access to their account would prevent legitimate users from exercising their data subject rights. The results of Paper I suggest that the data of some users are either weakly protected or prone to easily becoming inaccessible. Consequently, the users' account configurations directly affect their data subject rights. A good example is user accounts that are old and unmaintained but still contain personal data. The owners of such accounts may not be able to access their account credentials or required devices anymore. Yet, they may still want to exercise their *right to erasure* (GDPR, Art. 17) [20], i.e., to get their accounts and data deleted. Beyond that, RBA [90] and RBAR, as described in Paper II, can also have an essential influence on the availability and security of the users' data subject rights. Thus, the findings of these two papers are highly relevant regarding RQ3 as well.

### 3.2.3 Implicit User Authentication

Some services need to authenticate online users implicitly without prior enrolment of specific credentials. There still needs to be some form of trust relation between services and users. Besides services with proper user management, such cases are also relevant in connection with RQ1. Paper V focuses exactly on those services, i.e., third-party data collectors, where data subjects need to authenticate themselves to exercise their data subject rights. The data subjects can not influence how these services provide access to their data, and the GDPR does not provide any clear guidelines for this matter. It is entirely up to the data collector how to authenticate the users. Considering the current methods to identify data subjects in these cases, such as email or phone number verification, it might happen that they cannot exercise their data subject rights at some point. For example, they might change their phone number or email address. Moreover, since these methods to authenticate a user are generally considered insecure and were often applied improperly (cf. [11]), attackers may be able to bypass them. This further emphasizes why security and accessibility are crucial and should be considered by all online services.

## 3.3 Limitations

The research has shed light on how users adopt authentication methods and how this affects their security and account lockout risks. Furthermore, a specific part of the FIDO2 protocol has been enhanced to prevent attacks that would enable eavesdropping or manipulation. Finally, problems with authentication in the context of data subject rights have been investigated, and a solution has been proposed to counteract them. The scope of this thesis, however, does by no means cover all issues related to user authentication.

Also, the research questions have been answered to a certain degree, but they still leave room for future investigations. RQ1 has been looked at with a limited set of online services only. Even though these services are among the most popular, studies like those in Paper I and Paper II should be conducted on other services as well. Further, the survey in Paper I may be repeated with more participants and possibly in a lab environment in order to obtain more generalizable results. Nonetheless, the studies provide valuable insights into the implementation and adoption of authentication methods, particularly how widely they vary even among a small set of test participants. This suggests that future studies like these can indeed help in improving the security and accessibility of online users.

With regard to RQ2, where vulnerabilities in FIDO2 were investigated, only security issues were found in specific protocol extensions for certain use cases like online transactions. While it could be shown that there are actual issues that may be abused by attackers, this does not mean that there are no other vulnerabilities that could affect more common authentication scenarios with FIDO2. Passkeys offer a significantly bigger attack surface than traditional FIDO2 authentication, where credentials would never leave the authenticator device. However, this could not be covered within the scope of this thesis.

In Paper V, an SSI-based architecture is proposed to offer data controllers and, in particular, data subjects a more reliable solution for authentication in the context of data subject rights. A limitation is the requirement for data subjects to use a digital identity wallet, which is not fulfilled at the time of writing. According to the current eIDAS regulation, an EU identity wallet should be available to most EU citizens by 2030 [18]. Therefore, using an attribute-based identification could be a viable solution in the long run. Nevertheless, it will take at least several years until a wider use of such identity wallets can be expected. Given the above-mentioned risks of email- or phone-based data subject authentication, it would be beneficial to have other secure solutions that can be implemented in a timely manner. Therefore, further research concerning RQ3 is encouraged to find suitable alternative solutions.

## 3.4 Other Contributions

Beyond the main contributions included in this thesis, the author has contributed to the following articles in the field of information security throughout the PhD project:

- Andre Büttner, Hoai Viet Nguyen, Nils Gruschka, and Luigi Lo Iacono. “Less is Often More: Header Whitelisting as Semantic Gap Mitigation in HTTP-Based Software Systems”. In: *IFIP International Conference on*

*ICT Systems Security and Privacy Protection.* ISBN 978-3-030-78120-0; pages 332–347; Springer International Publishing (2021). DOI: 10.1007/978-3-030-78120-0_22. (June 2021).

**Summary:** This paper investigates specific security vulnerabilities of web applications resulting from intermediaries and servers with inconsistent HTTP implementations. In particular, the semantic gap of the HTTP protocol is analyzed, which is caused by ambiguities in the specifications and improper implementations. This leads to web applications processing invalid or non-standardized HTTP header fields. As a countermeasure, header whitelisting is proposed that can be applied to any HTTP component in order to mitigate related attacks.

- Johan Ivar Sæbø, Andre Büttner, Nils Gruschka, Bob Jolliffe, and Austin McGee. “Where There is no CISO”. In: *International Conference on Social Implications of Computers in Developing Countries*; ISBN 978-3-031-19429-0; pages 187–200; Springer International Publishing (2022). DOI: 10.1007/978-3-031-19429-0_12.

  **Summary:** This paper focuses on security in the context of health information systems in developing countries. It was found that there has been relatively little research on this particular area. Based on a literature review and experience from related research projects, relevant challenges for this context are addressed, and further recommendations for practitioners and researchers in this area are provided.

- Daniela Pöhn, Nils Gruschka, Leonhard Ziegler, and Andre Büttner. “A framework for analyzing authentication risks in account networks”. In: *Computers & Security*; Volume 135; Elsevier (2023). DOI: 10.1016/j.cose.2023.103515.

  **Summary:** In this paper, different risks related to user authentication are addressed, including security and accessibility. For that purpose, a framework is presented based on account access graphs to model user accounts, their authentication methods, and their dependency on other accounts. Furthermore, different scoring schemes are proposed to evaluate the security and accessibility of an account, and a KeePass plugin for end users is provided, as well as a web-based research tool.

Chapter 4

# Conclusion

## 4.1 Summary

In a world where nearly everything and everyone is connected to the Internet, people tend to make themselves, their companies, or organizations vulnerable to criminals, competitors, or adversaries, who can potentially steal their identities and, thereby, valuable assets. Nevertheless, digital identities have become an indispensable part of their lives. It is, therefore, of uttermost importance to secure these identities with strong and reliable authentication methods.

This thesis contributes to the protection of online users by exploring the use and security of modern authentication technologies. It shows that current solutions are, by far, not perfect, and users are often not sufficiently protected from account takeovers or other types of attacks in the context of authentication. It is the responsibility of service providers to offer adequate authentication methods and assist users in protecting their online accounts sufficiently. In most cases, however, it still depends on the users' decisions as to which authentication methods they ultimately activate and how they use them, especially the latter, which is beyond the control of the service provider. Service providers thus face a dilemma in addressing the users' preferences for convenience, providing the best security, and ensuring that their accounts remain accessible. Furthermore, this work proposes several technical solutions to improve user security in different authentication contexts. In the following, the main findings regarding the overarching research questions are recapitulated, and general conclusions on the development of authentication technologies are described.

### 4.1.1 Summary of Research Questions

> **RQ1** How are online users authenticated in practice, and what implications does it have on the security and accessibility of their accounts?

Online services offer different authentication methods, and often, users are left with the choice of which authentication methods to enable. The users themselves need to judge how much protection their account requires. Meanwhile, they also have to consider the availability of authentication methods whenever they want to sign into their account. Account recovery is equally important in this context and basically represents alternative authentication methods. One particular finding of this work is that if users have many different authentication methods to choose from, as with Google, the actual account configurations will vary significantly. As a consequence, many users end up with somewhat insecure accounts. Compared to this, the analyzed Apple user accounts appear to be more secure. However, for both of these services, there are several users whose accounts are at risk of being locked out. This is because their account access depends entirely on their smartphone, even when account recovery methods are set up, which actually intend to mitigate account lockout risks. As it turns out, account recovery methods can eventually rely on the same device used for the main authentication methods, thus not affecting accessibility at all.

Further, it was found that some services go beyond the authentication and recovery configurations made by the user. While previous work by other researchers [90] has analyzed the use of risk-based decision-making in standard authentication scenarios, similar observations have been made for account recovery. Risk-based authentication and recovery can increase the difficulty of accessing an online account for unusual and potentially malicious clients by incorporating additional authentication or recovery factors. While two of the five tested services do not apply any risk-based methods, two others only use a CAPTCHA in high-risk scenarios, which is presumably ineffective. However, one of them uses more sophisticated methods requiring special background knowledge or old authentication factors, which can present a significant obstacle to an attacker.

Another relevant use case discovered in this context is third-party data collectors, where online users need to prove their identity without any prior interaction with a particular service. The users have no choice in how these third-party services protect their data. Nonetheless, there needs to be an adequate balance between accessibility to the user and protection from illegitimate access. The methods currently used in practice, such as email and phone number verification, are considered unreliable. A possible solution for this could be the approach suggested in this thesis that takes advantage of SSI-based user attribute verification.

**RQ2** To what extent is FIDO2 authentication vulnerable to Man-in-the-Middle attacks, and how can this be mitigated?

FIDO2 is a new authentication technology aiming to overcome the security problems of passwords by replacing them with authenticator devices. Public-key cryptography and binding a key to a specific web domain will presumably prevent traditional phishing attacks. However, the WebAuthn and CTAP2 protocols have become significantly complex and may suffer from potential vulnerabilities.

This work indicates that FIDO2 is indeed vulnerable under specific circumstances. For instance, it is also considered for particular scenarios where, e.g., a banking transaction may be confirmed with a FIDO2 authenticator. This can be enabled by particular protocol extensions, such as the *transaction confirmation*, by which the authenticator signs a text representation of a transaction. Regarding this specific extension, it is argued that such transaction texts could be manipulated to make a user confirm an unintended transaction. As a countermeasure, the use of structured, machine-readable representations of transactions is suggested that do not allow for any unambiguity. Moreover, the risk of modified FIDO2 extensions is investigated in general. Since these are not cryptographically secured on transit, neither against manipulation nor against eavesdropping, a protocol has been proposed and tested that is compatible with FIDO2 and successfully mitigates any attacks.

> **RQ3** How can data subjects be securely authenticated when exercising their data subject rights as required by GDPR?

The GDPR requires that EU citizens have the possibility to exercise their data subject rights, including accessing, modifying, or deleting their data. Therefore, data collected by services, either directly or as a third party, should remain accessible to the rightful owner at any time. This work investigates those services that collect data indirectly, such as third-party advertising services. For these, data subjects do typically not have distinct user accounts. As a consequence, such services verify the ownership of, e.g., email addresses, phone numbers, or scanned ID documents. Since this was found to be error-prone, allowing attackers to access other people's data, a solution is proposed based on the self-sovereign identity paradigm. This entails a cryptographically secure verification of a set of identity attributes issued by the government or some other trusted service. The premise is that if the data held by a service is considered identifiable, in which case the data subject rights apply, the user should be able to prove the ownership of a sufficient set of personal attributes included in their identity wallet, like the name, date of birth, and more.

For those services where users have online accounts, the security is linked to their authentication configurations. As mentioned above, several users of Google, a service that processes a lot of personal data, may only have weak account protection. Consequently, there is a risk that others access their accounts maliciously to take advantage of their data subject rights. At the same time, some users of Apple and Google are at risk of losing access to their accounts, which, however, is crucial for them to make use of their data subject rights.

### 4.1.2 Evolving Authentication Technologies

This thesis has focused particularly on recent authentication technologies, including multi-factor authentication, passwordless authentication, and self-sovereign identity. A key observation from this work is that the addressed authentication technologies change rapidly. In fact, multi-factor authentication evolves continuously in that new authentication factors are included. In this regard, account recovery is closely related and must be balanced so the security of an online account is not compromised while the users do not lock themselves out entirely. Online services apply different strategies for this, and there does not appear to be a perfect solution. Risk-based approaches are another development observed within this work. Applying this to both authentication and recovery could help improve users' security as well. However, due to their rather untransparent use by online services, further work must investigate its actual potential.

Passwordless authentication, which barely drew any attention in the beginning, is presumably on the rise. FIDO2, as the key standard for this, started with a somewhat simple concept, where the security relied on a secret held by a single device. However, the risk of losing the device and the expected user effort for backups has raised concerns, ultimately slowing down its adoption. From there, FIDO2 has developed into passkeys with credentials that are backed up automatically across a user's devices. Despite the possible security compromise, this has led to significant adoption by major online service providers. At this point, its user adoption is uncertain and will likely be subject to future studies.

The self-sovereign identity is a new paradigm that allows users to keep full control over their identity attributes. This thesis has introduced a new use case, the implementation of data subject rights, which could further justify a wide use. While the EU makes an effort to enable its citizens to make use of this concept, the future will tell whether it will prevail successfully.

Moreover, it appears that there is a trend for any authentication method to be managed in a federated manner. Passwords can be accessed by password manager accounts, OTPs by OTP provider accounts, FIDO2 credentials by passkey provider accounts, and the self-sovereign identity by cloud-based identity wallets. In the long term, this can make authentication even more difficult to comprehend, and eventually, the security of the users will depend on how these accounts are protected.

## 4.2 Future Work

The research conducted within this thesis has addressed different aspects of authentication technologies. However, it still leaves room for a lot more to explore in this field. Some possible future directions that can be derived from this work are described below.

The federated nature of today's authentication methods makes it increasingly difficult to grasp how the online accounts of a single user depend on each other. An attacker may only need to find the weakest of those accounts to take over several of the user's accounts consecutively. For this reason, the overall account security and accessibility may be analyzed more thoroughly with account access graphs as used in Paper I. More such studies could contribute to a better understanding of whether online accounts are actually becoming more secure. The current developments lead to the assumption that, despite the efforts of service providers and users, their accounts might eventually depend just on a single password.

Another topic raised by this work is the use of risk-based account recovery, as discussed in Paper II. So far, it could only be observed to a limited degree by services like Amazon, Google, and LinkedIn. Future work may investigate if there are other services that apply similar approaches. Moreover, it could be researched what features are particularly useful in the context of recovery to determine if a client is suspicious or benign. At this point, account access graphs only consider static authentication mechanisms. Therefore, it would also be compelling to model risk-based authentication and recovery using such graphs.

FIDO2 has gained increasing attraction from the research community, and service providers have begun implementing it. Yet, FIDO2 authentication does not seem to have experienced its desired revolution to replace passwords. It currently occurs either as a secondary authentication factor in conjunction with or as an alternative to passwords. The goal of completely replacing passwords has not been achieved at this point, and it remains uncertain whether this will be the case in the near future. Hence, a comprehensive attacker model on passkeys and studies on the users' perception of this particular technology could help to clarify its potential in terms of security and adoption. Beyond this, the application of FIDO2 in use cases with online transactions is still under development and, therefore, has not been finalized. On the one hand, there is the *Secure Payment Confirmation* W3C standard [83] that may supersede the rather simplistic *transaction confirmation* extension addressed in this thesis. However, the extension is still under discussion [54], and similar arguments are brought up as those addressed in Paper III and Paper IV. Therefore, FIDO2-based online transactions may be further analyzed in future work.

The topic of authentication in the context of data subject rights could also benefit from further investigations. In particular, the approach of using identity attribute verification as presented in Paper V should be implemented and tested. In addition, alternative solutions may be developed that are more secure than the current state-of-the-art. Furthermore, instead of changing the entire procedure of existing methods, the verification of, e.g., email addresses, phone numbers, or ID documents may be improved systematically.

The concept of self-sovereign identities has been showcased within a proposed architecture. However, it is currently at an early stage and not widely used. The EU appears to be convinced that this is a promising authentication technology. Similar to FIDO2, its adoption may be studied from the users' and service providers' perspectives to reveal its actual usefulness.

# Bibliography

[1] 1Password. *Passkeys in 1Password: The Future of Passwordless Authentication | 1Password*. 2024. URL: https://1password.com/product/passkeys (visited on 04/02/2024).

[2] Alkhalil, Z. et al. “Phishing attacks: A recent comprehensive study and a new anatomy”. In: *Frontiers in Computer Science* vol. 3 (2021), p. 563060.

[3] Alnfiai, M. “Evaluating the accessibility and usability of a universal CAPTCHA based on gestures for smartphones”. In: *Universal Access in the Information Society* vol. 20, no. 4 (2021), pp. 817–831.

[4] Amft, S. et al. “"We’ve Disabled MFA for You": An Evaluation of the Security and Usability of Multi-Factor Authentication Recovery Deployments”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2023, pp. 3138–3152.

[5] B2B CYBER SECURITY. *Analysis: This is how an attack by the Akira ransomware group works - B2B Cyber Security*. Nov. 2023. URL: https://b2b-cyber-security.de/en/Analysis-This-is-how-an-attack-by-the-Akira-ransomware-group-works/ (visited on 04/03/2024).

[6] Bakhshi, T. “Social engineering: Revisiting end-user awareness and susceptibility to classic attack vectors”. In: *2017 13th International Conference on Emerging Technologies (ICET)*. IEEE. 2017, pp. 1–6.

[7] Bhattacharyya, D. et al. “Biometric authentication: A review”. In: *International Journal of u-and e-Service, Science and Technology* vol. 2, no. 3 (2009), pp. 13–28.

[8] Bisztray, T. et al. “Emerging biometric modalities and their use: Loopholes in the terminology of the GDPR and resulting privacy risks”. In: *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE. 2021, pp. 1–5.

[9] Bitwarden. *Passwordless and Passkeys | Bitwarden*. 2024. URL: https://bitwarden.com/passwordless-passkeys/ (visited on 04/02/2024).

[10] Blanchet, B. “Modeling and verifying security protocols with the applied pi calculus and ProVerif”. In: *Foundations and Trends® in Privacy and Security* vol. 1, no. 1-2 (2016), pp. 1–135.

[11] Boniface, C. et al. “Security analysis of subject access request procedures: How to authenticate data subjects safely when they request for their data”. In: *Privacy Technologies and Policy: 7th Annual Privacy Forum, APF 2019, Rome, Italy, June 13–14, 2019, Proceedings 7*. Springer. 2019, pp. 182–209.

[12] Bormann, C. and Hoffman, P. E. *Concise Binary Object Representation (CBOR).* RFC 8949. Dec. 2020. DOI: 10.17487/RFC8949. URL: https://www.rfc-editor.org/info/rfc8949.

[13] Corbató, F. J. "On building systems that will fail". In: *ACM Turing award lectures.* 2007, p. 1990.

[14] Das, S., Wang, B., and Camp, L. J. "MFA is a Waste of Time! Understanding Negative Connotation Towards MFA Applications via User Generated Content". In: *arXiv preprint arXiv:1908.05902* (2019).

[15] Das, S. et al. "MFA is A Necessary Chore!: Exploring User Mental Models of Multi-Factor Authentication Technologies." In: *HICSS.* 2020, pp. 1–10.

[16] European Commission. *Electronic Identification | Shaping Europe's digital future.* Apr. 2024. URL: https://digital-strategy.ec.europa.eu/en/policies/electronic-identification (visited on 04/08/2024).

[17] European Commission. *EU Digital Identity Wallet Pilot implementation.* en. 2023. URL: https://digital-strategy.ec.europa.eu/en/policies/eudi-wallet-implementation (visited on 06/15/2023).

[18] European Commission. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity.* SEC(2021) 228 final - SWD(2021) 124 final - SWD(2021) 125 final. June 3, 2021. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52021PC0281 (visited on 07/05/2023).

[19] European Commission. *Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.* July 23, 2014. URL: http://data.europa.eu/eli/reg/2014/910/oj (visited on 05/07/2023).

[20] European Parliament and Council. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance).* May 4, 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679 (visited on 03/21/2023).

[21] Fano, R. M. and Corbató, F. J. "Time-sharing on computers". In: *Scientific American* vol. 215, no. 3 (1966), pp. 128–143.

[22] Ferraiolo, D., Cugini, J., Kuhn, D. R., et al. "Role-based access control (RBAC): Features and motivations". In: *Proceedings of 11th annual computer security application conference.* 1995, pp. 241–48.

[23] FIDO Alliance. *Client to Authenticator Protocol (CTAP).* June 2021. URL: https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-20210615.html (visited on 01/26/2024).

[24] FIDO Alliance. *FIDO Alliance - Open Authentication Standards More Secure than Passwords*. 2024. URL: https://fidoalliance.org (visited on 01/04/2024).

[25] FIDO Alliance. *FIDO Alliance and W3C Achieve Major Standards Milestone in Global Effort Towards Simpler, Stronger Authentication on the Web*. Apr. 2018. URL: https://fidoalliance.org/fido-alliance-and-w3c-achieve-major-standards-milestone-in-global-effort-towards-simpler-stronger-authentication-on-the-web/ (visited on 01/10/2024).

[26] FIDO Alliance. *FIDO Alliance Member Companies & Organizations*. 2024. URL: https://web.archive.org/web/20240404095456/https://fidoalliance.org/members/ (visited on 04/04/2024).

[27] FIDO Alliance. *FIDO Transaction Confirmation White Paper*. Tech. rep. Aug. 2020. URL: https://media.fidoalliance.org/wp-content/uploads/2020/08/FIDO-Alliance-Transaction-Confirmation-White-Paper-08-18-DM.pdf.

[28] FIDO Alliance. *FIDO UAF Architectural Overview*. Feb. 2017. URL: https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-uaf-overview-v1.1-id-20170202.html (visited on 04/08/2024).

[29] FIDO Alliance. *History of FIDO Alliance*. 2023. URL: https://fidoalliance.org/overview/history/ (visited on 01/10/2024).

[30] FIDO Alliance. *History of FIDO Alliance*. 2024. URL: https://web.archive.org/web/20240324095729/https://fidoalliance.org/overview/history/ (visited on 03/24/2024).

[31] FIDO Alliance. *Universal 2nd Factor (U2F) Overview*. Nov. 2014. URL: https://web.archive.org/web/20230416040757/https://fidoalliance.org/specs/fido-u2f-v1.0-ps-20141009/fido-u2f-overview-ps-20141009.html (visited on 04/08/2024).

[32] FIDO Alliance. *User Authentication Specifications Overview*. 2024. URL: https://fidoalliance.org/specifications/ (visited on 01/04/2024).

[33] Freeman, D. et al. “Who Are You? A Statistical Approach to Measuring User Authenticity”. In: *NDSS ’16*. Internet Society, 2016. DOI: 10.14722/ndss.2016.23240.

[34] Gabrilovich, E. and Gontmakher, A. “The homograph attack”. In: *Communications of the ACM* vol. 45, no. 2 (2002), p. 128.

[35] Gartner. *Definition of Identity and Access Management (IAM) - Gartner Information Technology Glossary*. 2024. URL: https://www.gartner.com/en/information-technology/glossary/identity-and-access-management-iam (visited on 03/25/2024).

[36] Gelo, O., Braakmann, D., and Benetka, G. “Quantitative and qualitative research: Beyond the debate”. In: *Integrative psychological and behavioral science* vol. 42 (2008), pp. 266–290.

[37] Gerlitz, E. et al. "Adventures in recovery land: testing the account recovery of popular websites when the second factor is lost". In: *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. 2023, pp. 227–243.

[38] Google. *Passkey support on Android and Chrome*. https://developers.google.com/identity/passkeys/supported-environments. 2022. (Visited on 04/15/2024).

[39] Grassi, P. A., Garcia, M. E., and Fenton, J. L. *Digital identity guidelines: revision 3*. Tech. rep. NIST SP 800-63-3. Gaithersburg, MD: National Institute of Standards and Technology, 2017. DOI: 10.6028/NIST.SP.800-63-3.

[40] Grassi, P. A. et al. *Digital identity guidelines: authentication and lifecycle management*. Tech. rep. NIST SP 800-63b. Gaithersburg, MD: National Institute of Standards and Technology, 2017. DOI: 10.6028/NIST.SP.800-63b.

[41] Grimes, R. A. *Hacking Multifactor Authentication*. John Wiley & Sons, Ltd, 2020. DOI: 10.1002/9781119672357.

[42] Gritzalis, S., Spinellis, D., and Georgiadis, P. "Security protocols over open networks and distributed systems: Formal methods for their analysis, design, and verification". In: *Computer Communications* vol. 22, no. 8 (1999), pp. 697–709.

[43] Hadnagy, C. *Social Engineering: The Science of Human Hacking*. 2nd. Wiley Publishing, 2018.

[44] Hammann, S. et al. "User account access graphs". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 1405–1422.

[45] Hardt, D. *The OAuth 2.0 Authorization Framework*. RFC 6749. Oct. 2012. DOI: 10.17487/RFC6749. URL: https://www.rfc-editor.org/info/rfc6749.

[46] Identity Theft Resource Center. *2022 Annual Report*. 2022. URL: https://www.idtheftcenter.org/wp-content/uploads/2023/06/ITRC_2022-Annual-Report_Final-1.pdf (visited on 04/03/2024).

[47] Identity Theft Resource Center. *Microsoft hack targets senior leadership and staff | Fortune*. Jan. 2024. URL: https://fortune.com/2024/01/19/microsoft-senior-leadership-team-emails-accessed-russia-nation-state-hack/ (visited on 04/03/2024).

[48] Ives, B., Walsh, K. R., and Schneider, H. "The domino effect of password reuse". In: *Communications of the ACM* vol. 47, no. 4 (2004), pp. 75–78.

[49] Kirk, R. E. "Experimental design". In: *Sage handbook of quantitative methods in psychology* (2009), pp. 23–45.

[50] Krischer, Heinz. *Wer ist das Hacker-Netzwerk "Akira"? - Westfalen-Lippe - Nachrichten - WDR*. Nov. 2023. URL: https://www1.wdr.de/nachrichten/westfalen-lippe/wer-ist-hacker-netzwerk-akira-100.html (visited on 04/03/2024).

[51] Kumar, A. et al. *Web Authentication: An API for accessing Public Key Credentials - Level 2*. W3C Recommendation. Apr. 2021. URL: https://www.w3.org/TR/2021/REC-webauthn-2-20210408/.

[52] Landrock, P. and Pedersen, T. "WYSIWYS?—What you see is what you sign?" In: *Information Security Technical Report* vol. 3, no. 2 (1998), pp. 55–61.

[53] Li, Y. et al. "Understanding account recovery in the wild and its security implications". In: *IEEE Transactions on Dependable and Secure Computing* vol. 19, no. 1 (2020), pp. 620–634.

[54] Lindeman, Rolf. *Improved version of extension for Transaction Confirmation by rlin1 · Pull Request #2020 · w3c/webauthn*. 2024. URL: https://github.com/w3c/webauthn/pull/2020 (visited on 03/25/2024).

[55] Lyastani, S. G. et al. "Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication". In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 268–285.

[56] M'Raihi, D. et al. *HOTP: An HMAC-Based One-Time Password Algorithm*. RFC 4226. Dec. 2005. DOI: 10.17487/RFC4226. URL: https://www.rfc-editor.org/info/rfc4226.

[57] M'Raihi, D. et al. *TOTP: Time-Based One-Time Password Algorithm*. RFC 6238. May 2011. DOI: 10.17487/RFC6238. URL: https://www.rfc-editor.org/info/rfc6238.

[58] Masiliauskas, P. *Most Common Passwords 2024 - Is Yours on the List? | CyberNews*. 2024. URL: https://cybernews.com/best-password-managers/most-common-passwords/ (visited on 04/08/2024).

[59] McMillan, Robert. *The World's First Computer Password? It Was Useless Too | WIRED*. Jan. 2012. URL: https://www.wired.com/2012/01/computer-password/ (visited on 04/03/2024).

[60] Metz, C. "AAA protocols: authentication, authorization, and accounting for the Internet". In: *IEEE Internet Computing* vol. 3, no. 6 (1999), pp. 75–79.

[61] Misbahuddin, M., Bindhumadhava, B., and Dheeptha, B. "Design of a risk based authentication system using machine learning techniques". In: *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE. 2017, pp. 1–6.

[62] Moreno, L., González, M., and Martínez, P. “Captcha and accessibility”. In: *Is this the best we can do* (2014), p. 172.

[63] Morris, R. and Thompson, K. “Password security: A case history”. In: *Communications of the ACM* vol. 22, no. 11 (1979), pp. 594–597.

[64] Mühle, A. et al. “A survey on essential components of a self-sovereign identity”. In: *Computer Science Review* vol. 30 (2018), pp. 80–86.

[65] NordPass. *Top 200 Most Common Passwords.* 2024. URL: https://nordpass.com/most-common-passwords-list/ (visited on 04/08/2024).

[66] Organization for the Advancement of Structured Information Standards. *Security Assertion Markup Language (SAML) v2.0.* 2005.

[67] Owens, K. et al. “User Perceptions of the Usability and Security of Smartphones as FIDO2 Roaming Authenticators”. In: *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021).* USENIX Association, Aug. 2021, pp. 57–76.

[68] Panasiuk, P. et al. “A multimodal biometric user identification system based on keystroke dynamics and mouse movements”. In: *Computer Information Systems and Industrial Management: 15th IFIP TC8 International Conference, CISIM 2016, Vilnius, Lithuania, September 14-16, 2016, Proceedings 15.* Springer. 2016, pp. 672–681.

[69] Pashalidis, A. and Mitchell, C. J. “A taxonomy of single sign-on systems”. In: *Information Security and Privacy: 8th Australasian Conference, ACISP 2003 Wollongong, Australia, July 9–11, 2003 Proceedings 8.* Springer. 2003, pp. 249–264.

[70] Peffers, K. et al. “A design science research methodology for information systems research”. In: *Journal of management information systems* vol. 24, no. 3 (2007), pp. 45–77.

[71] Picard, C. and Pierre, S. “RLAuth: A Risk-based Authentication System using Reinforcement Learning”. In: *IEEE Access* (2023).

[72] Pöhn, D., Gruschka, N., and Ziegler, L. “Multi-Account Dashboard for Authentication Dependency Analysis”. In: *Proceedings of the 17th International Conference on Availability, Reliability and Security.* ARES ’22. Vienna, Austria: Association for Computing Machinery, 2022, pp. 1–13. DOI: 10.1145/3538969.3538987.

[73] Positive Technologies. *Social Engineering: How the human factor puts your company at risk.* 2018. URL: https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Social-engineering-2018-eng.pdf (visited on 03/09/2024).

[74] Preukschat, A. and Reed, D. *Self-sovereign identity.* Manning Publications, 2021.

[75] Reese, K. et al. “A usability study of five two-factor authentication methods”. In: *Proceedings of the Fifteenth Symposium on Usable Privacy and Security.* 2019.

[76] Sakimura, N., Bradley, J., and Jones, M. *Final: OpenID Connect Core 1.0 incorporating errata set 1.* en. 2014. URL: https://openid.net/specs/openid-connect-core-1_0.html (visited on 04/03/2023).

[77] Salazar, M. K. "Interviewer bias: How it affects survey research". In: *Aaohn Journal* vol. 38, no. 12 (1990), pp. 567–572.

[78] Saltzer, J. H. and Schroeder, M. D. "The protection of information in computer systems". In: *Proceedings of the IEEE* vol. 63, no. 9 (1975), pp. 1278–1308.

[79] Schwarz, N. "Survey methods". In: *The handbook of social psychology* vol. 1 (1998), pp. 143–179.

[80] Shaw, M. "What makes good research in software engineering?" In: *International Journal on Software Tools for Technology Transfer* vol. 4 (2002), pp. 1–7.

[81] Shirali-Shahreza, S. and Shirali-Shahreza, M. H. "Accessibility of CAPTCHA methods". In: *Proceedings of the 4th ACM workshop on security and artificial intelligence.* 2011, pp. 109–110.

[82] Simkus, Julia. *Between-Subjects Vs. Within-Subjects.* July 2023. URL: https://www.simplypsychology.org/between-subjects-vs-within-subjects-design.html (visited on 04/05/2024).

[83] Solomakhin, R. and McGruer, S. *Secure Payment Confirmation.* Candidate Recommendation. W3C, Dec. 2023. URL: https://www.w3.org/TR/2023/CRD-secure-payment-confirmation-20231213/.

[84] Sporny, M. et al. *Decentralized Identifiers (DIDs) v1.0.* W3C Recommendation. W3C, July 2022. URL: https://www.w3.org/TR/2022/REC-did-core-20220719/.

[85] Suchman, E. A. "An analysis of" bias" in survey research". In: *Public Opinion Quarterly* (1962), pp. 102–111.

[86] Summers, W. C. and Bosworth, E. "Password policy: the good, the bad, and the ugly". In: *Proceedings of the winter international synposium on Information and communication technologies.* 2004, pp. 1–6.

[87] Thibodeau Jr, T. et al. *Verifiable Credentials Data Model v2.0.* Candidate Recommendation. W3C, Apr. 2024. URL: https://www.w3.org/TR/2024/CRD-vc-data-model-2.0-20240402/.

[88] Tolosana, R. et al. "Biometric presentation attack detection: Beyond the visible spectrum". In: *IEEE Transactions on Information Forensics and Security* vol. 15 (2019), pp. 1261–1275.

[89] U.S. Department of Health and Human Services. *The Health Insurance Portability and Accountability Act of 1996 (HIPAA).* 1996. URL: https://www.hhs.gov/hipaa (visited on 03/04/2024).

[90] Wiefling, S., Lo Iacono, L., and Dürmuth, M. “Is this really you? An empirical study on risk-based authentication applied in the wild”. In: *ICT Systems Security and Privacy Protection: 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings 34*. Springer. 2019, pp. 134–148.

[91] Yuan, E. and Tong, J. “Attributed based access control (ABAC) for web services”. In: *IEEE International Conference on Web Services (ICWS’05)*. IEEE. 2005.

# Papers

Paper I

# Evaluating the Influence of Multi-Factor Authentication and Recovery Settings on the Security and Accessibility of User Accounts

**Andre Büttner, Nils Gruschka**

**Revision 01**
Date: April 2024
Editor: Andre Büttner

This is an updated version of the paper presented and published at ICISSP 2024. It includes changes in the format and minor text modifications. These changes do not affect the scope or conclusions of the paper in any way.

Originally published in *Proceedings of the 10th International Conference on Information Systems Security and Privacy - ICISSP*; ISBN 978-989-758-683-5; pages 691–700; INSTICC, SciTePress (2024). DOI: 10.5220/0012319000003648.

## Abstract

Nowadays, most online services offer different authentication methods that users can set up for multi-factor authentication but also as a recovery method. This configuration must be done thoroughly to prevent an adversary's access while ensuring the legitimate user does not lose access to their account. This is particularly important for fundamental everyday services, where either failure would have severe consequences. Nevertheless, little research has been done on the authentication of actual users regarding security and the risk of being locked out of their accounts. To foster research in this direction, this paper presents a study on the account settings of Google and Apple users. Considering the multi-factor authentication configuration and recovery options, we analyzed the account security and lock-out risks. Our results provide insights into the usage of multi-factor authentication in practice, show significant security differences between Google and Apple accounts, and reveal that many users would miss access to their accounts when losing a single authentication device.

## I.1 Introduction

Online services play an ever-increasingly important role in our digital world. We use them in many aspects of our private and business lives, like finance, transport, communication, entertainment, etc. Big tech companies—like Google and Apple—provide us with different services, including cloud storage, device management, payment, or single sign-on (SSO) to other online services. We highly depend on these services for more and more everyday tasks, and their availability and security are more crucial than ever.

Most online services require an account, and the user must authenticate to give proof of the ownership of their account. Authentication is still done mainly through passwords, although it is commonly known that passwords are a relatively weak method to protect a user account [26]. This has been confirmed in many studies, highlighting the problems with simple passwords and the reluctance to use password managers [6, 14, 25]. Therefore, multi-factor authentication (MFA) is usually offered to increase security and make compromising an account by a malicious user significantly harder. With MFA, the user needs to provide, in addition to the password, one[1] or more authentication factors, such as a one-time password (OTP) from an authenticator app or sent via SMS or a security key. Despite the apparent advantage, many users consider MFA inconvenient and refrain from activating it [5]. However, an increasing number of services make MFA mandatory and enforce configuring a second authentication factor. In addition, there is a movement towards passwordless authentication with the use of FIDO2 passkeys [10], but, at the time of writing, it is not widely adopted yet.

Suppose the MFA methods are not accessible, or users forget their password (or lose access to their password storage). In that case, most online services provide additional authentication factors, so-called account recovery mechanisms. Typical examples of these mechanisms are a link or code sent via email or SMS. However, such account recovery can be exploited if they are less secure than the main authentication, e.g., if a recovery email address is not protected sufficiently [16]. Also, a recovery factor might be bound to the same device as the main authentication. For example, the phone receiving the recovery SMS might be the same as the phone used for generating the MFA OTPs, which is certainly not an uncommon combination on a smartphone. Statistics show that smartphones often get stolen, damaged, or lost [2, 3], which can (in addition to the financial loss) make connected accounts inaccessible. Many providers warn users if the authentication is insecurely configured (e.g., no MFA activated or weak password). Still, they can not derive which device the configured factors are bound to. Furthermore, little research was done on this issue.

To tackle this research question, we conducted an online survey with 185 test participants to study the authentication configuration of their Google and Apple accounts[2]. To analyze the participants' responses, we extended and applied

[1] In this case, also called two-factor authentication (2FA). In this paper, we will generally use the term MFA.

[2] We selected Google and Apple, as they are two of the largest online services and offer a large variety of authentication configuration options.

*Account Access Graphs* (AAG) [13, 20], a new approach to evaluate MFA and recovery configurations. The analysis revealed insights into the account security and accessibility "in the wild". The main contributions of this paper are thereby as follows:

1. An improved accessibility scoring for AAG models with higher practical significance.
2. A security and accessibility analysis of the authentication setups by actual Apple and Google users.

The remainder of this paper is structured as follows. Section I.2 presents related work on MFA and AAGs. In Section I.3, the methodology of AAGs is described, as well as our security and accessibility scoring. Afterwards, we describe our study design in Section I.4. In Section I.5, we present our study results. Section I.6 discusses the limitations, our new accessibility scoring, and the implications of our results on MFA. The paper is finally concluded in Section I.7.

## I.2 Related Work

A study from 2015 analyzed how widely MFA authentication is adopted among Gmail users [19]. Within their test set, 6.39% of all users had an MFA method enabled, about 62% had configured a phone for account recovery, and 17% had provided a recovery email address. This gives a good picture of the adoption of MFA at the time of the study, but it might have changed significantly throughout the years. Their study is similar to ours regarding the collected data, i.e., enabled MFA and recovery methods. However, they obtained the data by using the *Google Password Reminder* feature with leaked email lists while we conducted an anonymous survey with the participants' consent.

Much research was done on the authentication and account recovery procedures offered by online services. A study by Gavazzi et al. [8] investigated how popular websites support MFA and risk-based authentication (RBA). They found that among the 208 websites they tested, about 42% provided MFA, and about 22% applied RBA. Amft et al., for example, evaluated the account recovery of 1303 websites based on their documentation and tested the actual recovery procedure on 71 of these websites. By this, they discovered many insecure procedures, eventually allowing them to bypass MFA authentication through recovery and significant deviations between the documentation and the actual procedure [1]. Another study analyzed usability flaws in account recovery when MFA is enabled on 78 popular websites [9]. Our paper, in contrast, focuses on which authentication and recovery methods provided by a service are adopted by users.

Other work has been done to compare different authentication methods based on security, privacy, and usability properties. The conclusion is that it is not feasible to find authentication schemes that perfectly satisfy all these properties [18]. More recent research has conducted a formal analysis of multi-factor authentication methods provided by Google, revealing some weaknesses in their protocols [15].

Furthermore, different studies exist that compare the usability of MFA methods. For example, Reese et al. have looked at the perceived usability of five different MFA methods in terms of usability and their efficiency [22]. In another study, the time used on MFA methods has been compared in two universities, showing that this can take up a significant amount of working hours over a long time [23].

Account Access Graphs [13, 20] is a methodology to analyze authentication systems systematically. It allows the modeling of authentication and recovery methods of online services and the dependencies between different user accounts. Further, it is possible to define dedicated scoring schemes to evaluate various aspects of an account, such as security and accessibility. It was shown that this could even become a practical tool for users to increase awareness about their account use and to support them in making it more secure [20]. Moreover, AAGs were used in a lab study to observe patterns in account setups [12]. The methodology used in this work mostly leans on the foundation provided in [20].

In this paper, we have enhanced the methodology of AAGs with an improved accessibility scoring scheme and applied it to actual user accounts. By automatically creating graphs of users, we show that analyzing user accounts with AAGs can be scalable for research on larger populations.

## I.3 Account Access Graphs

In an Account Access Graph as presented in [20], authentication methods and accounts are modeled as nodes in a graph. Account nodes are illustrated as rectangles, and authentication nodes as rounded rectangles. These nodes can be connected through intermediary nodes representing logical operators, including conjunctions '&' and disjunctions '|', depicted as circles. For instance, MFA can be modeled as a conjunction of two or more authentication methods. Recovery or alternative authentication mechanisms are modeled as a disjunction. Graphs can be created for the entire authentication system of an online service or the account setup of a specific user and can be used for qualitative and quantitative analysis.

In this paper, we use AAGs to evaluate the security and accessibility of actual user accounts. The respective scoring schemes are described below.

### I.3.1 Security Scoring

As suggested in [13], there are different possibilities to assess the security in an AAG. We adopted a scoring scheme inspired by the level of assurance as found in standards like NIST [11] or the EU electronic identification and trust services (eIDAS) [7], which is a well-established method to evaluate authentication methods. Consequently, we chose to use an ordinal scale including *low*, *medium* and *high*. An overview of how security values are assigned to different authentication methods is given in Table I.1.

Table I.1: Security scores assigned to different example authentication methods.

| Score | Category | Authentication Methods |
|---|---|---|
| High | Hardware-based | Security Key, Smart Card |
| Medium | Software-based | SMS Code, OTP Apps |
| Low | Knowledge-based | Password, PIN |

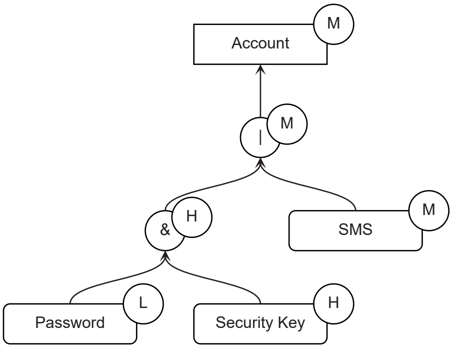


Figure I.1: Example graph for showing how security scores are calculated. The scores are indicated as L (low), M (medium), and H (high).

The security scores of parent nodes are calculated as follows. For the logical conjunction, the score is calculated using the maximum score of the child nodes. This is because all child nodes must be accessed to access this node, so the highest score defines the security level. The score for a logical disjunction is calculated by the minimum score of the children, as accessing the weakest of the children is sufficient to access a node. Figure I.1 shows an example for calculating security scores.

Some nodes in AAGs might represent other accounts, such as recovery email addresses. Ideally, the scores for these nodes should be derived from the respective AAGs specifically for that email account. However, since we could not consider all possible email accounts used, we opted for a worst-case calculation and assigned the value *low* to those leaf nodes.

### I.3.2 Accessibility Scoring

Users might lose their credentials or a device required for MFA authentication and be locked out of their accounts. Account accessibility scores indicate how many options users have to access their accounts. Consequently, a low accessibility score implies a high risk and a high score means a low risk of losing access to an account.

The accessibility scoring, as proposed in [20], provides a first estimation of whether the accessibility of a user account is stronger or weaker compared to others. However, it is not precise enough to give practical conclusions. We,

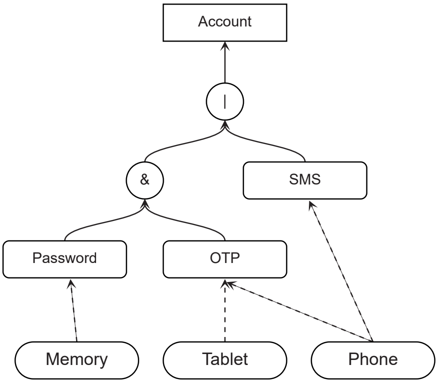


Figure I.2: Example graph with access methods.

therefore, propose an improved accessibility scoring.

The basic idea behind our improved scoring remains the same. All authentication methods have a physical "representation", e.g., passwords might be stored on a computer or remembered in memory, and SMS retrieval requires a phone to which it is sent. These elements are represented as *access method* in the AAG graph and are connected to the respective authentication node.

In contrast to the other nodes in an AAG, access methods may have more than one parent node (as shown in Figure I.2). If multiple different access methods can access an authentication method, it means that either one can access it. As a consequence, these access methods have a disjunctive relation. To assess the accessibility of a specific AAG, we use the logical formula based on the nodes in the graph.

$$(\text{Memory} \wedge (\text{Tablet} \vee \text{Phone})) \vee \text{Phone} \tag{I.1}$$
$$(\text{Memory} \wedge \text{Tablet}) \vee (\text{Memory} \wedge \text{Phone}) \vee \text{Phone} \tag{I.2}$$
$$(\text{Memory} \wedge \text{Tablet}) \vee \text{Phone} \tag{I.3}$$

As an illustrating example, the equations above show the boolean terms describing the accessibility of the instance in Figure I.2. First, we derive the term from the leaf nodes of the graph and replace them with the respective access methods (1). We then transform the term (2) to get a complete list of possible combinations of access methods for accessing that account. Finally, we reduce the term so only those access methods remain that are at least required for accessing the account (3). This means that an account cannot be accessed if all of the device combinations in this term are inaccessible.

We derive a numerical score based on this reduced term for a simple comparison at a larger scale. This is done by counting the number of occurrences $n_i$ of each access method $i$ within the reduced term and assigning them the value $s_i = 1/n_i$. The total accessibility score is calculated using the minimum of the conjunctive terms and the sum of the disjunctive terms. The resulting score $s_{acc}$ gives us the lower bound number of access methods one must lose to lose access completely. At the same time, $s_{acc} - 1$ can be interpreted as the upper bound number of access methods that can be lost without any risk of being locked out of an account. For the given example, the accessibility score is calculated as follows:

$$s_{acc} = min(1, 1) + 1 = 2$$

The accessibility score is 2, and losing one access method is therefore not critical. However, losing two methods, e.g., the phone and tablet, can make the account inaccessible to the user.

## I.4 Study Design

This section presents our study design. In particular, the research questions to be answered by this study are formulated. Furthermore, we describe the study procedure, the AAG models created for Apple and Google, the user account survey, and some ethical considerations.

### I.4.1 Research Questions

We designed a study to learn how users set up their Google and Apple accounts and how much they depend on their devices. One goal was to analyze how users access their passwords. Nowadays, password managers are becoming increasingly popular and are often even integrated into operating systems or browsers. This might affect the choice of passwords, i.e., whether they are easy to remember or rather created randomly and how accessible the password is. Furthermore, we wanted to check how users adopt MFA and recovery methods compared to earlier research [19]. In addition, we were interested in the general security of the users' accounts by considering the primary authentication and recovery methods. Finally, we wanted to assess how many devices users depend on. From this, the following research questions were derived:

- **RQ1** How do the users access their passwords?
- **RQ2** Which MFA and recovery methods did the users enable?
- **RQ3** How secure are the account setups?
- **RQ4** How many access methods do the user accounts depend on?

### I.4.2 Methodology

For this study, we first created models that describe the authentication systems of Google and Apple. We then created an online survey based on the authentication systems in which actual users were asked which authentication and recovery methods they use for their Google or Apple accounts and what means or devices

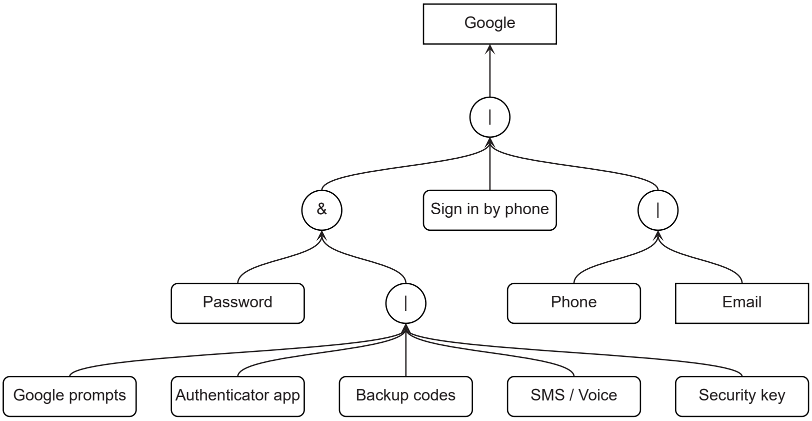


Figure I.3: AAG for Google.

they use to access them. After that, we generated user-specific AAG models based on the survey responses and calculated the respective security and accessibility scores. The general survey responses, as well as the security and accessibility scores, were finally analyzed concerning the research questions.

A research tool and some scripts have been used to facilitate the creation and analysis of the AAGs[3]. The web application tool can create and visualize AAGs and calculate security and accessibility scores. Moreover, we created Python scripts to automatically convert the survey responses of the test participants into actual AAGs. These AAGs were then imported into the research tool and subsequently examined.

### I.4.3 General AAG Models

The general AAG models for Google and Apple were created in January 2023. It was ensured that the AAGs were valid when the survey was conducted. This is important to consider as authentication systems change over time. For instance, since Spring 2023, Google has started to support FIDO2 passkeys as a passwordless alternative to its other authentication methods [4], which could not be covered in our study. The two AAG models used in this study are described in the following paragraphs.

#### Google

To create an AAG model for Google accounts, the security settings have been examined. This is where users can change their passwords or enable other types of authentication. Note that figuring out all authentication methods requires

[3] Our tools, survey questionnaires, anonymous participant data, and AAG files can be accessed at https://github.com/Digital-Security-Lab/user-account-study-icissp2024 (Last accessed: 2023-12-06)

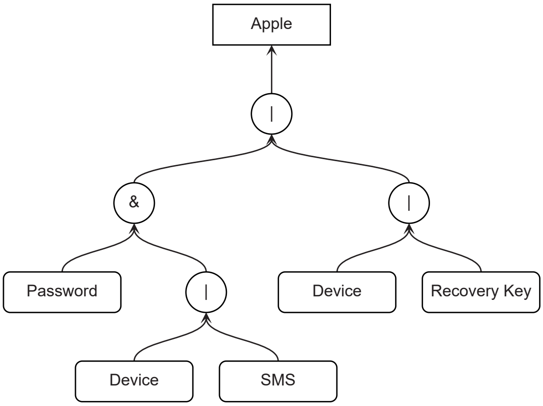


Figure I.4: AAG for Apple.

experimenting because specific options are interdependent. For Google accounts, the default primary authentication method is the password. In addition to that, it provides the option *sign-in by phone* and several MFA methods. Google users can optionally set up a recovery email address or phone number for recovery. The respective AAG model is shown in Figure I.3. In practice, Google is applying *risk based authentication* (RBA) [27], which means that it might request different authentication factors depending on certain features of the user's client. This, however, can not be modeled by the AAGs because they currently only represent static models.

### Apple

The AAG model for Apple has been created based on the Apple ID security settings. As with Google, the primary authentication method here is a password. However, users have fewer options to configure MFA in their Apple accounts. Secondary authentication factors can be either trusted phone numbers that are manually added by the user or trusted devices that are automatically configured when an Apple device is signed in to that Apple ID account. There is also the possibility of recovering accounts through customer service. This can not easily be modeled or evaluated using AAGs. Also, there is an option to set a recovery contact. These two options were excluded from the model. However, an Apple account can also be recovered using a trusted device. This option is implicitly enabled. Therefore, this has been added to the model. Furthermore, a user can explicitly configure a printable recovery key. If this is selected, this is the only possible way to recover an Apple account. The final AAG model for the Apple account is shown in Figure I.4.

### I.4.4 User Account Survey

The surveys were created as two separate online forms, each adapted to Google or Apple account settings respectively. Both surveys consist of two main parts.

First, the test participants had to make an enumerated list of the devices that they actively use. They should assign them to the categories of (1) *phone*, (2) *computer/laptop*, (3) *tablet*, (4) *smart watch* and (5) *security key*. These devices were referred to in subsequent questions.

For the second part, the participants were supposed to log in to their Google or Apple accounts. Afterward, they had to specify by which means they could access their password, i.e., whether they could remember it, whether it was stored in a password manager, in a browser, on a device, or whether they wrote it down on paper. They could choose multiple options if applicable. After that, they were guided to specific account settings and asked about them.

In the case of Google, they were asked whether MFA methods were enabled and, if so, which ones. In addition, they had to choose the devices by which these methods could be accessed. For instance, if Google prompts were selected, they had to indicate which phones or tablets were configured for this method. If MFA was disabled, they were asked if *sign-in by phone* was enabled and for which of the phones it was enabled. Last, they had to specify which recovery options were enabled, including email and phone. If a phone was selected, they had to choose the phone from their list of devices with the corresponding phone number.

For Apple, participants had to indicate which devices were registered as trusted devices and which used a trusted phone number. Finally, the participants should state which recovery options were enabled in their Apple accounts.

### I.4.5 Ethical Considerations

In the survey, the test participants were asked about what means they use to authenticate themselves to their Google or Apple accounts. This could generally be misused to exploit weak account configurations. However, the survey was completely anonymous, and no personal data was collected that could identify any of the participants by any means. Also, no other sensitive information, such as passwords or other secrets, was collected. The study was conducted in compliance with our university's research ethics guidelines. The test participants were thoroughly informed about the procedure and fairly compensated for their time. They could stop participating in the study at any time.

## I.5 Study Results

After several preliminary test runs, the survey was conducted in mid-January 2023, with test participants acquired using Prolific [21]. There were 94 submissions for Google (age 18-70, MV 34.89, SD 10.25) and 91 for Apple (age 19-67, MV 32.77, SD 9.7). In both cases, about half of the participants

Table I.2: Responses of test participants about their password usage.

| **Password access** | **Apple**<br>($n = 91$) | **Google**<br>($n = 94$) |
|---|---|---|
| I can remember it | 64 | 64 |
| Password manager | 28 | 37 |
| Stored by browser/device | 28 | 46 |
| I wrote it down on paper | 9 | 7 |

Table I.3: Frequency of authentication and recovery methods used in the participants' Google accounts ($n = 94$).

| **Authentication method** | **MFA** | **Frequency** |
|---|---|---|
| Password-only | ○ | 22 |
| Sign-in by phone | ○ | 8 |
| Google prompts | ● | 26 |
| Authenticator app | ● | 15 |
| Backup codes | ● | 8 |
| Voice or text message | ● | 38 |
| Security key | ● | 2 |
| Recovery phone | n/a | 64 |
| Recovery email | n/a | 76 |

were residents of the USA and the other half of Germany. Below, we describe the study results concerning the research questions.

## I.5.1 Password Access (RQ1)

To answer RQ1, we assessed how people access their passwords. A summary of the responses from our survey is given in Table I.2. Most participants still tend to use a password they can remember. This applies to both Google and Apple. Yet, many also use a password manager or the password storage functionality of a browser or device. According to our results, less than 10% of the participants have written it down on paper. It can be further observed that about one-third of the Apple test participants use more than one method to store their password, which applies to almost 48% of the Google participants.

## I.5.2 Adoption of MFA and Recovery Methods (RQ2)

In RQ2, we wanted to analyze to what extent the users have adopted MFA and recovery methods. Apple and Google are very different in their approach to MFA. Apple devices are implicitly configured as MFA methods for accounts they are signed in. Therefore, Apple users most likely have MFA enabled automatically, so we did not examine this for Apple in more detail.

Google users have several different authentication and recovery methods that need to be configured manually. The usage of each method is shown in Table I.3. According to our results, 68% of the Google test participants have at least one MFA method enabled. In the previously mentioned study from 2015, only less than 7% had MFA enabled [19]. Furthermore, in 2018, a Google researcher stated that by that time, only 10% of Gmail users had set up MFA [17]. Compared to that, there is a significant increase in our test results. One must keep in mind that our sample size of 94 Google test participants is relatively small. Yet, an increase was expected since Google started to enroll MFA automatically for its users as announced in 2021 [24]. The most common secondary authentication method is voice or text message. Many of the participants also appear to use Google prompts and authenticator apps. Backup codes and security keys, in contrast, are used only rarely.

Regarding recovery methods, there also seems to be a wider adoption of a recovery email address (80%) and a slightly increased use of the phone as a recovery method (68%) compared to the results in [19].

### I.5.3 Account Security (RQ3)

RQ3 was addressed by assessing the security scores of the AAGs created for all test participants. In general, it was observed that Google's recovery methods often decrease the security of an account, even if a user intends to use more secure authentication methods like the security key. This was particularly the case because most test participants used a recovery email. Nevertheless, it must be kept in mind that the security score for email was set to *low* because the accurate score could not be assessed within this study. In reality, the score might be higher depending on the email provider and account setup. However, as mentioned before, it was shown that recovery emails are often a weakness in an account setup [16]. The frequency of each security score for Google and Apple is summarized in Figure I.5.

One can observe that there is no Google account with a total security score *high*. The only way to achieve this would be to disable all recovery methods and only enable a security key as a second authentication factor. Two test participants have not enabled recovery methods but used Google prompts as the second authentication factor, with a score of *medium* within our scoring scheme. Most Google accounts have a total score of *low*. This is mainly because most test participants have enabled email as a recovery method. The lesson learned is that Google account security in practice often depends on the security of recovery email accounts.

Two Apple users are rated with the score *high*. This is because they have only set up a *trusted device* but no *trusted phone number*. Most Apple accounts have a score of *medium*. The main reason here is likely that most Apple users use their account in connection with at least one Apple device, which, as mentioned before, automatically becomes a second authentication factor.

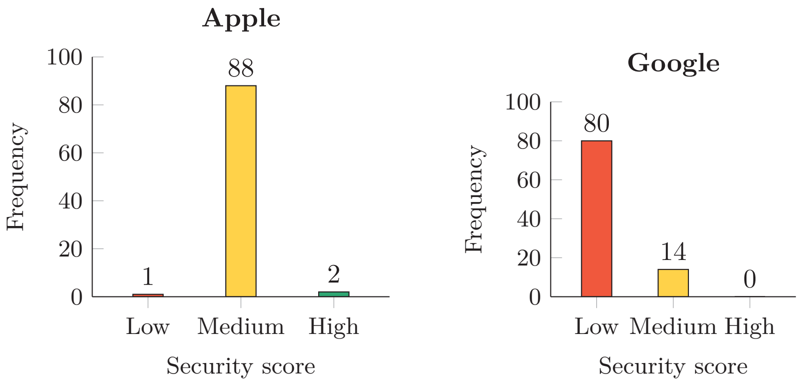


Figure I.5: Histogram over security scores of the participants' Apple and Google accounts.

### I.5.4 Account Accessibility (RQ4)

We answered RQ4 by analyzing the accessibility scores for each test participant's Google and Apple accounts. The scores were derived from the respective AAGs, including the access methods used by the test participants. In Figure I.6, it can be observed that the range of accessibility scores is between 1 and 5 for Apple and between 1 and 6 for Google.

As described in Section I.3.2, an accessibility score equal to 1 indicates that an account can become inaccessible if one of the access methods is unavailable, and we, therefore, analyzed these setups in more detail. Within the test sample, ten of the Google accounts have an accessibility score equal to 1. For Apple, this applies to seventeen accounts. After manual analysis, it was found that in sixteen cases for Apple and all ten cases for Google, the primary phone was the key access method that, if lost, could cause an account lockout. This is reasonable since smartphones play a crucial role in our lives today. The Google account of one test participant was even dependent on the availability of both a memorized password and a phone for Google prompts.

Beyond that, most test participants had accessibility scores of 2 or higher. This means that the majority's risk of being locked out of their account is relatively low.

## I.6 Discussion

In this section, we discuss the limitations of the conducted study, reflect on the new accessibility scoring, and, finally, discuss what conclusions can be drawn by our research for MFA in practice and how our study might contribute to future improvements.

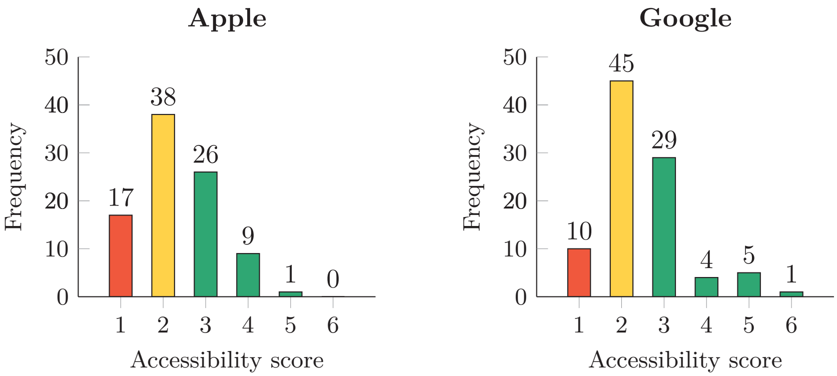


Figure I.6: Histogram over accessibility scores of the participants' Apple and Google accounts.

### I.6.1 Limitations

First of all, the study was conducted with a relatively low sample size of 91 Apple users and 94 Google users. Hence, the study is rather an initial study, and the results must be interpreted with that in mind. Yet, the results give us an idea of how many possible account setups are used in practice and what potential accessibility risks exist. Also, the results only reflect a snapshot of when the study was conducted, as authentication systems change with time.

Another limitation is the simplification of the Google model, where email accounts have been assigned a low security score. In practice, this will differ for each account. However, it requires a rather sophisticated survey covering all possible email account setups.

Moreover, as mentioned earlier, RBA has not yet been considered in AAGs because the models are currently static. This might influence both security and accessibility. For that, a better understanding of RBA is required, and future work should look into how dynamic AAGs may be modeled and evaluated.

### I.6.2 Accessibility Scoring

An essential contribution of this work is the new accessibility scoring described in Section I.3.2. In contrast to the previously proposed method [20], where no transformation or reduction was conducted after assessing the first Boolean term, our approach has a more practical meaning since it shows how many access methods a user depends on. For the example shown in Figure I.2, the score as in [20] would result in an accessibility score of 1.5, while our method resulted in a score of 2.

A numerical score does not provide context about the devices. Therefore, it could be beneficial to parse the logical formula into a human-readable description with more context about the actual devices. For instance, the final term from the example in Figure I.2 can be compiled into a description like "Access to *Account* might be lost when losing both *Phone* and *Tablet*, or losing your *Phone* and forgetting your password". In practice, this could help inform online users more effectively of the consequences of losing a particular access method and motivate them to set up an alternative access method for authentication or recovery.

### I.6.3 MFA in Practice

Our study shows that several users depend solely on their primary phones. For online services, it is challenging to assess the risk of their users being locked out of their account because the services can not know if, e.g., an authenticator app is installed on the same device that is used for SMS as a second authentication factor, or that has stored a user's password. We found that, especially for Apple, seventeen test participants might lose access to their accounts if they lose their phones. For Google, this applies to ten users. Given that, this also means that having access to the users' phones is enough to access an account. From a user perspective, one could argue that the term *multi-factor* authentication does not always apply, even though users have enabled multiple authentication methods in their accounts. The likelihood of a phone being compromised is relatively low, as it usually requires local authentication through a PIN or biometrics or rather sophisticated malware. It is still a scenario that must be considered, especially by users with high-security requirements, e.g., due to their profession.

Also, we found that MFA and recovery are often linked to the same device. Beyond this study, we observed that, e.g., Facebook prohibits using the same phone number for MFA and account recovery, which is a step in the right direction. Surprisingly, other applications, such as LinkedIn, automatically enable both options when setting up a phone number. Web applications should encourage users to use different devices for different authentication methods. Therefore, it might be helpful to have a tool for users to keep an overview of their account and device dependencies, as suggested in [20].

It is generally not very obvious how accessible an account is. This depends on how people set up their accounts and how devices and password access are linked. AAGs can help assess this and encourage service providers and users themselves to improve the security and accessibility of online accounts. Therefore, we suggest the integration of AAGs into consumer tools and online services. Likewise, AAGs can be a powerful tool in an enterprise context for an administrator to get an overview of the account security and accessibility of employees.

## I.7 Conclusion

In this paper, we have extended the methodology of Account Access Graphs with a new accessibility scoring scheme. Moreover, we have developed and conducted a study in which we analyzed several aspects of Apple and Google user accounts, such as their MFA adoption and their security and accessibility. Within our security scoring scheme, Apple accounts turned out to be more secure, likely due to the nature of Apple devices being implicitly configured for multi-factor authentication. For Google, we found that several users still have not enabled any second authentication factor. One of the most important findings concerning accessibility was that several Google test participants and even more Apple test participants entirely depended on their phones. Even though this was not the majority, several users risk losing access to their user accounts. All in all, our study has demonstrated that AAGs can contribute to a deeper understanding of authentication methods.

We will pursue this research direction in future work. Similar studies may be conducted for other web services to see how secure and accessible their users' account setups are. Furthermore, it can be investigated how service providers can use the knowledge gained to improve their authentication systems concerning security and accessibility. Finally, dynamic AAG models may be developed for analyzing accounts of online services that apply risk-based authentication.

## References

[1] Amft, S. et al. “"We've Disabled MFA for You": An Evaluation of the Security and Usability of Multi-Factor Authentication Recovery Deployments”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2023, pp. 3138–3152.

[2] Beatriz Henríquez. *Mobile Theft and Loss Report - 2020/2021 Edition*. en. https://preyproject.com/blog/mobile-theft-and-loss-report-2020-2021-edition. June 2022.

[3] Bitkom. *Gestohlen oder verloren: Vier von zehn Personen ist schon mal das Handy abhandengekommen*. en. https://www.bitkom.org/Presse/Presseinformation/Gestohlen-oder-verloren-Vier-von-zehn-Personen-ist-schon-mal-das-Handy-abhandengekommen. 2021.

[4] Brand Christiaan, K. S. *The beginning of the end of the password*. https://blog.google/technology/safety-security/the-beginning-of-the-end-of-the-password/. May 2023. (Visited on 04/15/2024).

[5] Das, S., Wang, B., and Camp, L. J. “MFA is a Waste of Time! Understanding Negative Connotation Towards MFA Applications via User Generated Content”. In: *arXiv preprint arXiv:1908.05902* (2019).

[6] Davis, D. K., Chowdhury, M. M., and Rifat, N. “Password Security: What Are We Doing Wrong?” In: *2022 IEEE International Conference on Electro Information Technology (eIT)*. IEEE. 2022, pp. 562–567.

[7] European Commission. *eIDAS Levels of Assurance*. en. URL: https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/eIDAS+Levels+of+Assurance (visited on 04/14/2023).

[8] Gavazzi, A. et al. “A Study of Multi-Factor and Risk-Based Authentication Availability”. In: *32nd USENIX Security Symposium (USENIX Security'23). USENIX Association, Anaheim, CA, USA*. 2023.

[9] Gerlitz, E. et al. “Adventures in recovery land: testing the account recovery of popular websites when the second factor is lost”. In: *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. 2023, pp. 227–243.

[10] Google. *Passwordless login with passkeys.* https://developers.google.com/identity/passkeys. 2022. (Visited on 04/15/2024).

[11] Grassi, P. et al. *Digital Identity Guidelines: Authentication and Lifecycle Management.* en. 2020. DOI: https://doi.org/10.6028/NIST.SP.800-63b.

[12] Hammann, S. et al. “I’m Surprised So Much Is Connected”. In: *CHI Conference on Human Factors in Computing Systems.* 2022, pp. 1–13.

[13] Hammann, S. et al. “User account access graphs”. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security.* 2019, pp. 1405–1422.

[14] Hayashi, E. and Hong, J. “A diary study of password usage in daily life”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 2011, pp. 2627–2630.

[15] Jacomme, C. and Kremer, S. “An extensive formal analysis of multi-factor authentication protocols”. In: *ACM Transactions on Privacy and Security (TOPS)* vol. 24, no. 2 (2021), pp. 1–34.

[16] Li, Y., Wang, H., and Sun, K. “Email as a master key: Analyzing account recovery in the wild”. In: *INFOCOM ’18.* IEEE. 2018. DOI: 10.1109/INFOCOM.2018.8486017.

[17] Milka, G. “Anatomy of Account Takeover”. In: *Enigma 2018 (Enigma 2018).* Santa Clara, CA: USENIX Association, Jan. 2018. URL: https://www.usenix.org/node/208154.

[18] Ometov, A. et al. “Multi-factor authentication: A survey”. In: *Cryptography* vol. 2, no. 1 (2018), p. 1.

[19] Petsas, T. et al. “Two-factor authentication: is the world ready? Quantifying 2FA adoption”. In: *Proceedings of the eighth european workshop on system security.* 2015, pp. 1–7.

[20] Pöhn, D., Gruschka, N., and Ziegler, L. “Multi-Account Dashboard for Authentication Dependency Analysis”. In: *Proceedings of the 17th International Conference on Availability, Reliability and Security.* ARES ’22. Vienna, Austria: Association for Computing Machinery, 2022, pp. 1–13. DOI: 10.1145/3538969.3538987.

[21] Prolific. *Prolific · Quickly find research participants you can trust.* https://www.prolific.com. 2023. (Visited on 10/02/2023).

[22] Reese, K. et al. “A usability study of five two-factor authentication methods”. In: *Proceedings of the Fifteenth Symposium on Usable Privacy and Security.* 2019.

[23] Reynolds, J. et al. “Empirical measurement of systemic 2fa usability”. In: *Proceedings of the USENIX Conference.* 2020.

[24] Risher, M. *A simpler and safer future — without passwords.* https://blog.google/technology/safety-security/a-simpler-and-safer-future-without-passwords/. 2021.

[25] Shen, C. et al. “User practice in password security: An empirical study of real-life passwords in the wild”. In: *Computers & Security* vol. 61 (2016), pp. 130–141.

[26] Taneski, V., Heričko, M., and Brumen, B. “Systematic overview of password security problems”. In: *Acta Polytechnica Hungarica* vol. 16, no. 3 (2019), pp. 143–165.

[27] Wiefling, S., Lo Iacono, L., and Dürmuth, M. “Is this really you? An empirical study on risk-based authentication applied in the wild”. In: *ICT Systems Security and Privacy Protection: 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings 34*. Springer. 2019, pp. 134–148.

Paper II

# Is it Really You Who Forgot the Password? When Account Recovery Meets Risk-Based Authentication

**Andre Büttner, Andreas Thue Pedersen, Stephan Wiefling, Nils Gruschka, Luigi Lo Iacono**

**Revision 01**
Date: April 2024
Editor: Andre Büttner

This is an updated version of the paper presented and published at UbiSec 2023. It includes changes in the format and minor text modifications. These changes do not affect the scope or conclusions of the paper.

Originally published in *Ubiquitous Security. UbiSec 2023*; ISBN 978-981-97-1274-8; pages 401–419; Springer Nature Singapore (2024). DOI: 10.1007/978-981-97-1274-8_26.

## Abstract

Risk-based authentication (RBA) is used in online services to protect user accounts from unauthorized takeover. RBA commonly uses contextual features that indicate a suspicious login attempt when the characteristic attributes of the login context deviate from known and thus expected values. Previous research on RBA and anomaly detection in authentication has mainly focused on the login process. However, recent attacks have revealed vulnerabilities in other parts of the authentication process, specifically in the account recovery function. Consequently, to ensure comprehensive authentication security, the use of anomaly detection in the context of account recovery must also be investigated.

This paper presents the first study to investigate risk-based account recovery (RBAR) in the wild. We analyzed the adoption of RBAR by five prominent online services (that are known to use RBA). Our findings confirm the use of RBAR at Google, LinkedIn, and Amazon. Furthermore, we provide insights into the different RBAR mechanisms of these services and explore the impact of multi-factor authentication on them. Based on our findings, we create a first maturity model for RBAR challenges. The goal of our work is to help developers, administrators, and policy-makers gain an initial understanding of RBAR and to encourage further research in this direction.

## II.1 Introduction

Passwords are still the pre-dominant authentication method for online services, even for services that give access to confidential data or financial resources [14, 31]. However, attacks on password authentication can be automated—e.g., credential stuffing using leaked passwords—and therefore scaled with little effort. This makes account takeover attacks on password-protected online services very lucrative for hackers [3]. As a countermeasure, more and more services offer multi-factor authentication (MFA) as an extension to password authentication. In this case, the user has to give additional proof of their identity, e.g., by entering a code from a one-time password (OTP) app or a text message (SMS). However, the additional step makes the authentication process more cumbersome and increases the risk of account lockouts in case the additional token gets lost [30].

The idea of risk-based authentication (RBA) [12, 14, 37] is to balance security and usability. Here, the online service only requests additional authentication steps or blocks a client when it detects suspicious behavior. RBA does this by analyzing a set of feature values (e.g., location, browser, or login time) during the login process [14, 37].

A general problem with authentication is that the user might lose access to the authentication method—in the case of password authentication, this means primarily forgetting the password. In such a case, the user has to pass the *account recovery* process to regain access to their account. The process often involves sending a password reset link or an OTP to a pre-configured email address or phone number. If the required authentication (e.g., ownership of a phone, login to the email account) is weaker than the primary authentication, account recovery puts the overall account security at risk [27, 29].

A high and common threat to account recovery mechanisms via email is when an attacker gains access to the corresponding email account, e.g., via credential stuffing [2, 33]. The recent FBI cybercrime report [11] shows that compromised email addresses and phishing attacks are very popular attacks with potentially high financial loss for the hacked victims. Therefore, it is very important for online services to secure account recovery, for example, with MFA or RBA. So far, risk-based mechanisms have mostly been studied in the context of login authentication. However, we observed that mechanisms similar to RBA are also used for account recovery.

We define *Risk-Based Account Recovery* (RBAR)[1] as a dynamic account recovery process on online services. It was indicated that such a method is used at a large online service [7], but beyond that, RBAR and its appearances in the

[1] To the best of our knowledge, there is no standard term for it yet.

wild have not been publicly investigated yet. This is, however, important as it has the potential to protect a large number of users from account recovery attacks immediately. To learn about the current use of RBAR, we address the following research questions in this paper:

**RQ1:** Do RBA-instrumented online services also use RBAR mechanisms?

**RQ2:** What RBAR challenges are used in practice?

**RQ3:** Are different RBAR challenges required when setting up MFA?

**Contributions.** This paper presents the first scientific insight into using RBAR in practice. We performed an exploratory analysis of RBAR behavior at Google and a systematic experiment on four other popular online services. We verified RBAR at three of the five services. The analysis also included the influence of MFA configurations and different (virtual) locations. The main contributions achieved from these activities are the following:

- Identification of RBAR at popular online services
- A maturity model for different RBAR mechanisms

The remainder of this paper is structured as follows. Section II.2 provides an overview of related work. In Section II.3, we describe details behind how RBAR works. Section II.4 explains the methodology of our experiments. The findings of the two experiments are described in Sections II.5 and II.6, respectively. Our overall results are discussed in Section II.7. Section II.8 summarizes our work and suggests possible future work.

## II.2 Related Work

Most of the previous work on account recovery considered it a static mechanism. For instance, a lot of research focused on different additional authentication challenges for recovery that can be solved easily by legitimate users but not by potential attackers. Examples include cryptographic keys [9], delegated account recovery [20, 22], dynamic security questions [1, 19], and email address or phone number verification [26]. While these works do not address risk-based use cases, we argue that such methods would be beneficial in conjunction with a risk analysis of the user context.

Further research evaluated online services in the wild. Li et al. [24] studied the account recovery mechanisms of 239 popular online services in 2017 and 2019. They found that most of them implemented email address or mobile phone verification as a recovery mechanism. Amft et al. [6] conducted a large-scale study investigating which recovery methods are usually deployed in conjunction with MFA methods. They unveiled that website documentation usually does not correspond with the actual recovery procedure, showing the lack of transparency in account recovery. We confirm this as we analyzed the documentation of the

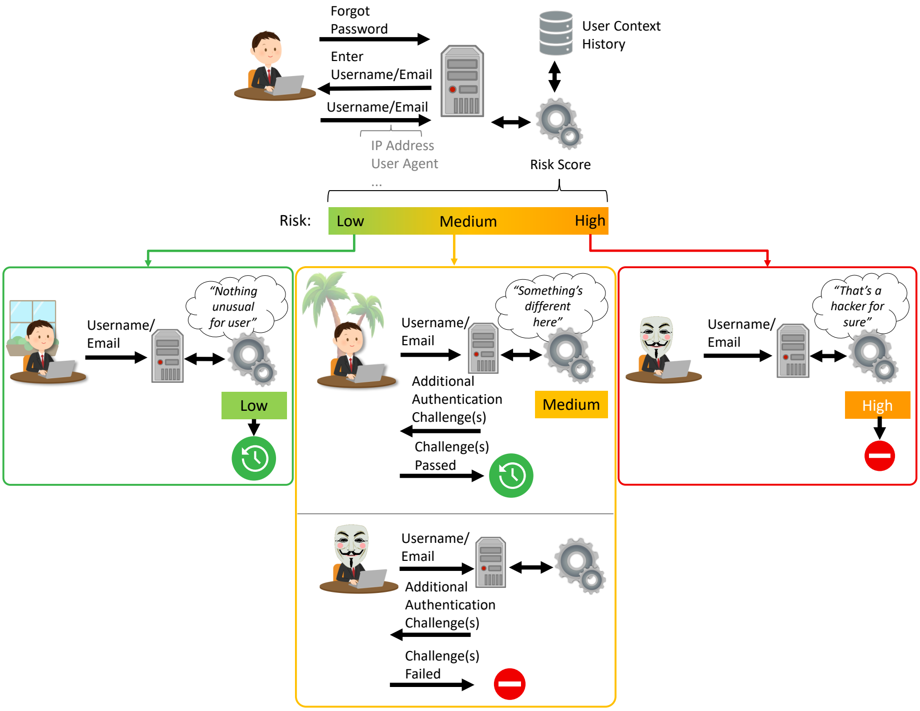


Figure II.1: Overview of the RBAR procedure (based on RBA illustration in [35])

services we tested for any references to RBAR, which in most cases were absent (see Section II.7).

The only indication of risk-based recovery mechanisms we found in literature was mentioned by Bonneau et al. [7], where they noted that Google performed a *"risk analysis"* for account recovery. However, they did not further investigate how it works or what mechanisms are applied depending on the risk scenario.

Research on RBA is especially relevant for our work as it provides us with methods to analyze and develop risk-based systems. For example, Wiefling et al. [37] studied RBA re-authentication mechanisms on five popular online services. They found that most online services used email verification to re-authenticate users. Gavazzi et al. [14] leaned on this work to identify that more than 75% of the 208 studied online services do not use any form of RBA. While the research in this field only addresses plain user authentication, our work extends it by showing that the methods used in RBA research can be equally applied in the context of account recovery. Consequently, we used the insights from prior work on RBA as a basis to study the use of RBAR on Google and other online services.

## II.3 Risk-Based Account Recovery

Since there is no official description of RBAR yet, we describe its basic concept. Based on our observations on online services and previous knowledge in the related RBA field [36, 37], RBAR works as follows (see Figure II.1):

A user typically starts an account recovery process, e.g., by clicking *"forgot password"* at the online service's login form. After that, the user is asked to enter the username or email used for the account to recover. While submitting this identifier, the user also submits additional feature data that is available in the current context to the online service, e.g., IP address or user agent string. Based on this information, RBAR compares these values with the user context history and calculates a risk score. The user context history contains feature values of past user actions, like previous legitimate logins that might have been validated by RBA mechanisms [38] before. The risk score is then classified into low, medium, and high risk. Based on the risk, the online service performs different actions.

At a *low* risk, the feature values likely belong to the legitimate user, and the online service proceeds with the account recovery process (e.g., verify email address). A *medium* risk occurs if the user's feature values deviate from the expected values. The online service then introduces additional authentication challenges that require more user effort (e.g., solving a CAPTCHA or answering questions related to the account). After successfully solving these challenges, the online service proceeds with the account recovery process. A *high* risk means that the online service suspects that the user is likely targeted by a hacking attempt. The online service might block the account recovery process in these cases. However, to avoid locking out legitimate users trying to recover their accounts, this possibility has to be carefully selected by the online service.

## II.4 Methodology

We investigated the research questions by conducting two experiments. Prior research has indicated that Google applies risk-based decision-making for account recovery [7], making it a suitable candidate for our first experiment. Therefore, we conducted an exploratory experiment on Google. We created test cases with different account setups, i.e., different authentication and recovery factor combinations. These were then tested with different user features to see how these could affect the recovery procedure. The study considered two RBA features, as suggested in Wiefling et al. [37]: known/unknown browser and known/unknown IP address. A *known* browser is the one that was used before to sign in to Google, i.e., it has stored cookies from prior sessions. The *unknown* browser was tested using the browser's incognito mode to have a clean browser session without previously set cookies. The IP address feature was varied by using a VPN connection to be able to study the uncertain area of medium to high risk scores [37]. By comparing the recovery procedures of the different features for each test case, we identified the mechanisms used for RBAR. The test cases and the final results are given in Section II.5.

For the second experiment, we developed an improved and more systematic approach. As the experiment required manual effort, we limited the number of tested services to the following services that are known to use RBA [14, 37]:

- LinkedIn (`linkedin.com`)
- Amazon (`amazon.com`)
- GOG (`gog.com`)
- Dropbox (`dropbox.com`)

The experiment was composed of three phases. First, we prepared user accounts for each service. Afterward, we checked whether any of the online services indicated RBAR behavior. Finally, since LinkedIn clearly turned out to implement RBAR, we analyzed if RBAR on LinkedIn is influenced by the MFA settings (as was the case with Google). More details on the steps and the results are presented in Section II.6.

## II.5 Experiment 1: RBAR Use by Google

In the first experiment, we investigated previous assumptions [7] on whether Google used RBAR and identified features that might have an influence on the RBAR behavior. We describe the experiment and its results in the following.

### II.5.1 Preparation

The exploratory experiment on Google was conducted between October 2021 and March 2022. We set up four Google user accounts that were created at intervals of several weeks to mitigate being detected as a researcher. Based on the visible feedback from the online service, we assume that we remained under the respective detection thresholds. In order to test the use of RBAR on Google, we defined the test cases based on the authentication and recovery factors offered in the Google account settings. At the time of the study, Google provided the following factors:

- **Main authentication**: password, sign in by phone
- **Secondary authentication**: Google prompt, phone call or text message, backup codes, security key, authenticator app
- **Recovery factors**: email, phone

The experiment on Google covered every possible single-factor authentication (SFA) account setup and eight MFA account setups. Each account setup was tested with all four RBA feature combinations. For each combination, all possible recovery options were explored.

Table II.1: Examples for Google account recovery without MFA enabled

| **Recovery factor** | **Phone signed in** | **Known browser** | **Known IP** | **Recovery procedure** |
|---|---|---|---|---|
| None | ○ | ● | ● | Recovery not possible |
| None | ● | ● | ● | 1. Google prompt |
| None | ● | ○ | ○ | 1. Enter old password<br>2. Google prompt (two steps) |
| Email | ○ | ● | ● | 1. Verify account email |
| Email | ○ | ○ | ● | 1. Enter old password<br>2. Verify account email |

● = *Feature present,* ○ = *Feature not present*

### II.5.2 Results

The study found that Google used RBAR for both SFA and MFA account setups. This became clear as using an unknown browser and/or an unknown IP address increased the difficulty of recovering the account compared to using a known browser and IP address. This was indicated by requiring additional authentication factors, recovery options that were made unavailable, or an extra prompt like asking for the phone number of a registered phone.

**Recovery Without MFA Enabled.** Table II.1 lists a few examples[2] of the tests from studying SFA account recovery that clearly show the different recovery procedures based on RBA features. One can observe that in cases where an unknown browser was used for recovery, Google initially asked for an old password that the user could remember. This was not the case when using a known browser and a known IP address. The recovery procedure continued the same way, even if this step was skipped.

When a phone was signed in to the same Google account, this phone was prompted with a button showing *"Yes, it's me"*. Users had to click this button to confirm the ownership of the account. This behavior changed when trying to recover the account from an unknown browser and an unknown IP address. In this case, Google also showed a two-digit number on the recovery web page and presented a dialogue with three number options on the phone. Users then had to select the correct number on the phone to proceed with the recovery.

**Recovery With MFA Enabled.** Table II.2 shows some of the results that indicated obvious differences when trying to recover an account with a phone number configured for MFA. Note that in the given examples, we omitted the step of verifying access to the actual Google account email address to see what alternatives would be offered. When the recovery was performed from a known browser, it was sufficient to verify the phone that was set up for MFA by entering

[2] All results for the tests on Google are published on https://github.com/AndreasTP/GoogleAccountRecovery (Last accessed: 2024-04-24).

Table II.2: Examples for Google account recovery with phone (text message) enabled for MFA

| Recovery factor | Known browser | Known IP | Recovery procedure |
|---|---|---|---|
| None | ● | ● / ○ | 1. Verify MFA phone<br>2. ~~Verify account email~~<br>3. Verify new email<br>→ Reset email after 48hrs |
| None | ○ | ● | 1. Verify MFA phone<br>2. ~~Verify account email~~<br>→ Recovery not possible |
| None | ○ | ○ | 1. Enter MFA phone number<br>2. Verify MFA phone<br>3. ~~Verify account email~~<br>→ Recovery not possible |

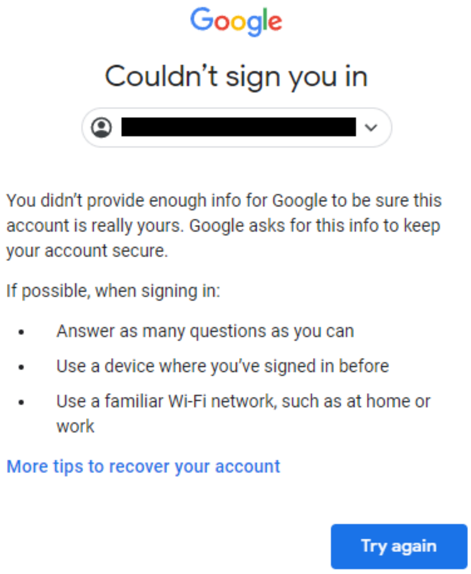
● *= Feature present,* ○ *= Feature not present,* ~~*XXX*~~ *= Step omitted*

an OTP code that was sent to the phone via text message. Afterward, Google provided the user with an option to register and verify a new email address. A password reset email was sent to the newly registered email after 48 hours. In the meantime, the (legitimate) account owner got notifications about the ongoing recovery attempt. This allowed them to stop the procedure in case they did not request the recovery. However, this recovery option was not available when using an unknown browser. In that case, the user needed access to both the phone number and the email address registered on the actual Google account. This highlights how much RBAR features can impact the user's chance of a successful recovery.

The last example in Table II.2 shows a recovery procedure when using both an unknown browser and an unknown IP address. In this case, the user was first asked to enter the phone number used for MFA before actually verifying the ownership of this phone number.

**Further Observations.** Also, we observed that when failing a recovery, Google revealed some information on how its RBAR mechanism might work. The message displayed to the user on a failed recovery attempt suggested using a known device and Wi-Fi during recovery (see Figure II.2).

However, during the study, we experienced that the recovery process could change from one day to another. This was true despite using the same account, having the same recovery options configured, and using the same browser and IP address. For instance, a recovery procedure that earlier gave access to the account after 48 hours through a password reset email ended in a failed recovery. An authentication factor that could previously be used to help recover an account was occasionally removed as a recovery option. This suggests that Google uses more RBAR features than the two tested in this study. Nonetheless, we confirm



Couldn't sign you in

[redacted]

You didn't provide enough info for Google to be sure this account is really yours. Google asks for this info to keep your account secure.

If possible, when signing in:

- Answer as many questions as you can
- Use a device where you've signed in before
- Use a familiar Wi-Fi network, such as at home or work

Figure II.2: Message shown when failing Google's account recovery using an unknown browser and an unknown IP address. It reveals information that might give indications of their inner RBAR workings.

the assumption in prior work that Google implements a risk assessment in its recovery [7].

## II.6 Experiment 2: RBAR Use by Other Services

The second experiment focused on online services that are known to use RBA [37] and investigated whether and how they also use some form of RBAR. We describe the experiment and its results below.

### II.6.1 Preparation

For this experiment, we began by setting up user accounts for all four online services (see Section II.4). Testing account recovery with personal accounts is not ideal since there is always the risk that accounts will be locked out or disabled entirely. However, RBA is oftentimes triggered only for legitimate accounts with a certain history of activity [37]. This makes sense from a technical perspective, as such algorithms need a certain amount of training data from the legitimate user to work correctly [38]. Therefore, we created four new test accounts for each of the services. These accounts were set up with the most basic settings, i.e., with a password and one email address. To avoid bias, we made sure to create and use new email addresses on general-purpose email providers not linked to universities for each account. In addition, we were able to provide one old account for each service, some of which were either personal or created in previous studies.

A training was conducted in which the test accounts were logged in more than 20 times within a time period of about 1.5 months (December 2022–January 2023). We based the number of logins on Wiefling et al.'s study [37]. Furthermore, it was ensured that the logins for each account were performed with a similar context, i.e., from the same browser and the same IP location. Also, logins from university IP addresses were avoided since experience has shown that online services might recognize these IP addresses and block accounts to prevent systematic analyses of their services. For reproducibility, we documented the context before each account login. We did this by recording all information from the IP address and HTTP header and the browser's internal JavaScript functions, as in related work [36].

### II.6.2 Identifying RBAR Usage

After training the test accounts, we analyzed whether the online services actually use RBAR mechanisms. As in related work by Wiefling et al. [37] and Gavazzi et al. [14], this was tested by discovering differences in two distinct user contexts: *normal* and *suspicious*. This time, we considered a normal user to perform the account recovery from the same browser and IP location as in the training phase. In contrast, the suspicious user performs account recovery from a Tor browser. Web services can typically recognize Tor browser clients by the IP address of the exit nodes or by other browser features. Moreover, using a Tor browser is often considered suspicious [37]. We expected this to increase the likelihood of triggering risk-based mechanisms, if any, and compared to the first experiment on Google, where the Tor browser was not used. Note that we only considered differences that occurred after starting the recovery procedure for a specific account, e.g., after entering an email address. Any differences beforehand would not be relevant as it would mean that it is independent of the history of a user account.

**Experimental Procedure.** For this within-group experiment, account recovery was performed twice for each test account on different days at the end of January 2023, once with a normal user context and once with a suspicious user context, in varying orders, to avoid bias. This means we performed two account recoveries with all provided accounts. In the case of Amazon, we repeated the experiment with one of the new accounts and another old account due to inconsistent results, as described in more detail below.

**Results.** Table II.3 summarizes the recovery procedures for each online service and account. Overall, the presentation of a CAPTCHA was the only noticeable difference that was found. The CAPTCHAs in the table are underlined in those cases where they appeared only in the suspicious user context. Note that Amazon uses its own AWS WAF CAPTCHA [5], while GOG uses the Google reCAPTCHA v2 [17] and LinkedIn appears to use a custom CAPTCHA implementation. Dropbox did not use any CAPTCHA within our experiments.

Table II.3: Account recovery procedures for a normal and suspicious user context for the different test accounts of each online service

| **Online Service** | **Account** | **User context** | |
|---|---|---|---|
| | | Normal | Suspicious |
| Amazon | A1, A2, A4, A6* | EC | EC |
| | A3, A1† | CA → EC | CA → EC |
| | A5* | EC | $\underline{\text{CA}}$ → EC |
| Dropbox | D1 – D4, D5* | EL | EL |
| GOG | G1 – G4, G5* | CA → EL | CA → EL |
| LinkedIn | L1 – L4, L5* | EC | $\underline{\text{CA}}$ → EC |

*EC = Email (Code), EL = Email (Link), CA = CAPTCHA,*
** = Old account, † = Experiment repeated,* $\underline{\textit{XXX}}$ *= Additional step*

For **Amazon**, in three cases, only an OTP code sent via email was requested. Afterward, the password could be changed. For one of the new test accounts (A3), Amazon first requested a CAPTCHA before the email OTP code, but for both normal and suspicious contexts. For the old account (A5), there was an actual difference as the CAPTCHA was only displayed in the suspicious context. Because of this inconsistent behavior, we did an additional test with A1, which this time required solving a CAPTCHA for both user contexts, similar to A3. Furthermore, we included a test with another personal account (A6) that was actively used to check if the behavior was related to the account age or activity. This time, no CAPTCHA had to be solved. Consequently, the risk assessment was more complex and could not be easily reproduced with our experimental setup.

**Dropbox** only requested the verification of the email address through a link before the password could be changed. This was the same for all user accounts, including the old one, and for both user contexts.

For **GOG**, a CAPTCHA had to be solved before verifying the email address through a link and finally changing the password. This was again equal for all accounts and both normal and suspicious user contexts.

**LinkedIn** was the only online service that consistently showed a different behavior depending on the context. For a normal user context, the email address had to be verified by an OTP code before the password could be changed. However, when performing recovery from a suspicious user context, a CAPTCHA had to be solved, sometimes multiple times.

In summary, Amazon and LinkedIn used RBAR, while Dropbox and GOG have not indicated any risk-based behavior during recovery. The only challenge that was shown depending on the user context was a CAPTCHA. The results for Amazon, however, were inconsistent for the different accounts. It was decided not to do a deeper analysis here, as the experimental setup clearly did not consider enough context parameters to simulate both a normal and a suspicious user context reliably. Yet, we conclude that Amazon must have used some form

Table II.4: Account recovery procedures for a normal and suspicious user context for the different LinkedIn account setups

| # | Recovery | | MFA | | User context | |
|---|---|---|---|---|---|---|
| | Second Email | Text (SMS) | Auth. App | Text (SMS) | Normal | Suspicious |
| 1 | ● | ○ | ○ | ○ | EC1 \| EC2 | CA → EC1 \| EC2 |
| 2 | ○ | ● | ○ | ○ | EC1 \| P1 | CA → EC1 \| P1 |
| 3 | ○ | ○ | ● | ○ | EC1 → AU | CA → EC1 → AU |
| 4 | ○ | ○ | ○ | ● | EC1 → P2 | CA → EC1 → P2 |
| 5 | ● | ○ | ● | ○ | EC1 \| EC2 → AU | CA → EC1 \| EC2 → AU |
| 6 | ● | ● | ○ | ● | EC1 \| EC2 → P2 | CA → EC1 \| EC2 → P2 |
| 7 | ○ | ● | ○ | ● | EC1 → P2 | CA → EC1 → P2 |
| 8 | ○ | ● | ● | ○ | EC1 \| P1 → AU | CA → EC1 \| P1 → AU |

*● = Feature present, ○ = Feature not present, EC1 = Primary Email (Code), EC2 = Secondary Email (Code), P1: Recovery Phone (SMS Code), P2 = MFA Phone (SMS Code), AU = Authenticator App, CA = CAPTCHA, | = Alternative XXX = Additional step*

of RBAR. For LinkedIn, on the other hand, the RBAR behavior could clearly be reproduced with all accounts. Thus, we conducted a second experiment on LinkedIn using the newly created test accounts described in the subsequent section.

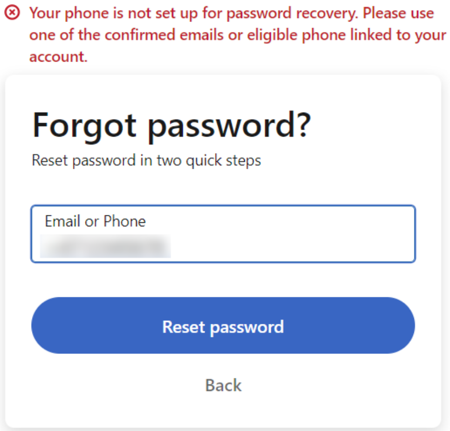
### II.6.3 Analyzing the Influence of MFA Settings on Account Recovery on LinkedIn

In Section II.5, we showed that Google implements RBAR by incorporating different authentication mechanisms that are set up as MFA factors in a user account. Since we could prove that LinkedIn also provides some form of RBAR, we conducted another experiment to determine whether LinkedIn used any other RBAR challenges beyond the CAPTCHA.

**Experimental Procedure.** For this experiment, we changed the authentication and recovery options in the LinkedIn test accounts. At the time of this experiment (January–February 2023), LinkedIn provided the following authentication and recovery methods:

- **Main authentication**: password
- **Secondary authentication**: phone (SMS), authenticator app
- **Recovery factors**: email address, phone (SMS)

We tested the effects of all possible combinations of these methods. In addition, LinkedIn also offered a non-digital recovery method requiring the user to submit a copy of a government-issued ID. As this would have revealed the experimenters' identities, we did not include this method in the experiment. Similar to Google, the expected outcome for LinkedIn was that different authentication factors would be requested in a suspicious user context.



⊗ Your phone is not set up for password recovery. Please use one of the confirmed emails or eligible phone linked to your account.

Forgot password?

Reset password in two quick steps

Email or Phone

[redacted]

Figure II.3: Error message for phone recovery, when also Text Message MFA is activated

**Results.** Table II.4 shows the results for the different tested account setups. Note that in setups 1, 2, 5, 6, and 8, there are two possibilities for receiving the verification code: as an alternative to the primary email address, the secondary email address or the phone number could be entered (indicated by the "|" symbol). LinkedIn allows configuring the same phone number as a second authentication factor and as a recovery method. In fact, when enabling the phone number for MFA, the same number is activated automatically for recovery by phone. However, in such cases, using the phone for account recovery does not make much sense as only a single factor (ownership of the SIM card) is required for resetting the password and logging in afterward, which contradicts the idea of *multi*-factor authentication. In these cases, i.e., setups 6 and 7, we only received an inaccurate error message (see Figure II.3). We filed a bug report for this to LinkedIn on February 24, 2023. However, the response from LinkedIn (one day later) indicated that it will not be fixed anytime soon unless it gets noticed by several other users.

The experiments show that the behavior when configuring further recovery or authentication methods is identical to the base setup. The only difference in the account recovery procedure for all setups was the initial CAPTCHA shown in the suspicious user context. Apart from that, the account recovery procedure always started with the verification of the primary email address or phone number by an OTP code, followed by the verification of the MFA method if one was activated.

**Variation of CAPTCHA Iterations.** In addition to our main results, we observed that the number of iterations of the CAPTCHA on LinkedIn varied in different experiments between 1 and 5. When mapping the number of iterations to the pretended location (i.e., the location of the Tor exit node), an interesting correlation showed up (see Table II.5). The normal usage location for all accounts was in Europe, and when the pretended location was also in Europe (just another

Table II.5: Number of CAPTCHA iterations for different (pretended) locations for account recovery on LinkedIn

| **CAPTCHA iterations** | **Location of Tor exit** |
|---|---|
| 1 | Sweden, Poland, United Kingdom, *Mexico* |
| 2 | United Kingdom, Germany |
| 3 | 3× USA, *Czech Republic* |
| 5 | USA, Canada, *Netherlands* |

country), 1 or 2 repetitions of the CAPTCHA were required. In cases where the suspicious location was on a different continent, 3 or 5 repetitions were needed. However, there were also cases (marked in italics) where this was not true. Nonetheless, it indicates that LinkedIn's RBAR might give different suspicious risk classifications that are reflected in the number of CAPTCHA iterations. It also seems that the location is one important feature. Further experiments are needed to analyze to what extent other features are included.

## II.7 Results and Discussion

In our exploratory study on Google and the follow-up experiment with other online services, we confirmed that several online services apply RBAR to a certain degree. In this section, we describe the results of the experiments with regard to the research questions. Furthermore, we summarize the results in a maturity model that we propose for RBAR implementations. Finally, we outline the limitations of our experiments and discuss further aspects of RBAR usage in practice.

### II.7.1 Experiment Results

Within the scope of our experiments, we observed that Google implements RBAR in quite a sophisticated manner. It showed different authentication methods depending on the account setup and the user context. Dropbox and GOG did not apply any risk-based mechanisms during account recovery. Amazon actually indicated the use of RBAR, however, by assessing context information that was not considered by our two different user contexts. In some tests, a CAPTCHA had to be solved, while in others, it was not required. LinkedIn clearly behaved differently in a suspicious user context. When trying to recover an account from a Tor browser, LinkedIn showed a CAPTCHA challenge before entering an email verification code. In contrast to Google, however, the RBAR for LinkedIn did not involve MFA settings in a user account.

With regard to **RQ1**, we conclude that there are online services that use RBA, which also use RBAR—including Google, Amazon, and LinkedIn—but not all of them. To answer **RQ2**, the challenges we found on Google include pre-configured MFA methods (e.g., phone number) and questions requiring background knowledge (e.g., old passwords). On LinkedIn and Amazon, we only

Table II.6: Maturity model with maturity levels, mapping of RBAR challenges to the tested services and possible attacks against these challenges

| Maturity | RBAR challenge | Identified on | Possible attacks |
|---|---|---|---|
| 3 | Pre-configured MFA | Google | Physical attack, malware [8] |
| 2 | Background knowledge | Google | OSINT, leaked passwords, phishing [1, 19] |
| 1 | CAPTCHA | LinkedIn, Amazon | Manual recovery, CAPTCHA bypass algorithm [21, 32] |
| 0 | None | Dropbox, GOG | n/a |

observed a CAPTCHA challenge in connection with RBAR. Concerning **RQ3**, we found that the MFA settings influenced the recovery procedure on Google only, while LinkedIn did not vary RBAR challenges depending on any configured MFA methods.

### II.7.2 Maturity Model

Based on our results and inspired by [30], we propose a maturity model that ranks the different RBAR challenges by difficulty for an attacker (see Table II.6). Due to the nature of RBAR, the model only considers the measures used in connection with a risk assessment. It describes the additional security gain in case the primary recovery factor (e.g. email address), if any, has already been compromised. Thus, no RBAR at all is considered the least mature as it does not involve any risk assessment and does not provide additional measures. Showing a CAPTCHA is ranked as level 1 as it can prevent automated attacks. Yet an attacker might bypass it or manually exploit account recovery. Background questions are ranked as level 2 as they require an attacker to gather knowledge of a victim. However, it also increases only the cost of the attack. MFA methods that are pre-configured in an account are considered the most mature as they require more sophisticated methods or even physical access for a successful attack.

The model can be used, e.g., to assess the security of an RBAR implementation. Online services can also use such a model for their RBAR implementations to enable certain challenges with a higher maturity ranking at higher risk scores. Note that the model is only one possible way to assess RBAR. It might be different if other types of RBAR challenges are used that were not discovered within our study.

### II.7.3 Comparison with Official Documentation

To the best of our knowledge, the experiments showed for the first time that Amazon, LinkedIn, and Google use RBAR. To compare our findings with the public communications of the online services, we took their official documentation into consideration [4, 10, 15, 18, 25]. Interestingly, none of the RBAR-instrumented online services mentioned that they change the account recovery behavior based on contextual information collected during the recovery process [4, 18, 25]. Only Google hinted that users should possibly use a familiar device and location [18]. However, they did not mention why users should do this, i.e. because they use RBAR. Our results show that the account recovery mechanisms of these online services seem to do more to protect their users than what is officially communicated to them.

Trying to hide implemented security mechanisms from the user base has already been observed in the related case of RBA [16] and other research on account recovery [6]. We do not consider this a good practice, as it follows the anti-pattern of *“security by obscurity”*. Users also tend to get frustrated when they experience security barriers that were not communicated to them beforehand [34]. Beyond that, attackers are known to adapt to obscured security mechanisms [28, 33]. We assume that public RBAR research, to increase the body of knowledge, will increase the overall adoption of online services and enable a large user base to be protected with RBAR following the principle of *“good security now”* [13].

### II.7.4 Ethics

We only tested account recoveries with accounts owned by the researchers, i.e., we did not try to exploit the recovery of other users' accounts. Also, since we conducted manual tests, we did not create high traffic on the online services that could have affected other users.

While it could be reasoned that our findings are helpful for attackers, we argue that they are more valuable to the public. As the gained knowledge helps researchers and online service providers to get an understanding of how RBAR works, this can support the development of more secure and usable account recovery mechanisms.

### II.7.5 Limitations

Beyond Google, only four online services were analyzed in terms of RBAR. This was mainly due to the lack of any automatism for training user accounts and testing account recovery, therefore requiring manual effort to conduct our experiments. Nevertheless, as mentioned before, these services have been carefully selected as they are known to use RBA [37].

We could not find any RBAR mechanisms in Dropbox and GOG. Due to the nature of a black-box test, we do not know the implementation details of the tested online services. Thus, there is always uncertainty involved. Nevertheless, we are confident that the accounts were sufficiently trained—especially since we also tested older accounts—and tested with the highest risk possible [37].

### II.7.6 RBAR

Attackers may abuse account recovery to circumvent authentication. Hence, the security of account recovery is as essential as the security of login authentication.

Previous research showed that email addresses often become a single point of failure [23, 24]. RBAR might be an advantageous way to increase the difficulty of a successful account takeover by incorporating additional authentication methods, as with RBA. At the same time, it may reduce the burden on legitimate users and increase their chances of recovering an account.

The RBAR used by Google is quite different from LinkedIn. Google uses additional authentication methods, while LinkedIn just requires a suspicious user to solve an additional CAPTCHA. This CAPTCHA actually only reduces the risk of automated attacks by making it more costly for an attacker. In general, CAPTCHAs mainly increase friction for users [39]. It may be an improvement to use a risk score to decide if a CAPTCHA should be solved, compared to, e.g., GOG, where a CAPTCHA is shown to all users. However, the security gain is insignificant since researchers have already demonstrated attacks against Google's widely known reCAPTCHA [21, 32]. Moreover, this does not prevent targeted attacks. We argue that if a service already implements a risk assessment in its account recovery, it should even go further and include actual authentication methods. In the case of LinkedIn, it could, for instance, request the verification of another recovery email or phone if set up.

## II.8 Conclusion

Account recovery mechanisms remain a relevant entry point for account takeover attacks [27, 29]. Online services should strengthen their account recovery with additional security mechanisms, like risk-based account recovery (RBAR), to protect their users.

In this paper, we investigated the use of RBAR in practice. We described the concept behind RBAR and conducted two experiments to learn about if and how online services use it. The results show that Google, Amazon, and LinkedIn used RBAR. However, their implementations differed widely in suspicious contexts, from asking users for background knowledge or pre-configured MFA methods (Google) to showing a CAPTCHA challenge (Amazon and LinkedIn). Based on our results, we proposed a maturity model that researchers or service providers can use to assess the security of RBAR systems or guide in implementing RBAR.

Following this first systematic analysis of RBAR, future work can extend our proposed model with other RBAR challenges. Furthermore, it can be studied what features specifically trigger RBAR challenges. As there seems to be a tendency to include risk-based decision-making into account recovery, there should be a comparison of RBA and RBAR and how they can complement each other in authentication systems as a whole.

## References

[1] Addas, A., Salehi-Abari, A., and Thorpe, J. "Geographical Security Questions for Fallback Authentication". In: *PST '19*. IEEE, 2019. DOI: 10.1109/PST47121.2019.8949063.

[2] Akamai. “Credential Stuffing: Attacks and Economies”. In: *[state of the internet] / security* vol. 5, no. Special Media Edition (2019). URL: https://web.archive.org/web/20210824114851/https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-credential-stuffing-attacks-and-economies-report-2019.pdf.

[3] Akamai. “Loyalty for Sale – Retail and Hospitality Fraud”. In: *[state of the internet] / security* vol. 6, no. 3 (2020). URL: https://web.archive.org/web/20201101013317/https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-loyalty-for-sale-retail-and-hospitality-fraud-report-2020.pdf.

[4] Amazon. *Reset Your Password.* 2023. URL: https://web.archive.org/web/20210918230138/https://www.amazon.com/gp/help/customer/display.html?nodeId=GH3NM2YWEFEL2CQ4.

[5] Amazon Web Services, Inc. *What is a CAPTCHA puzzle?* 2023. URL: https://docs.aws.amazon.com/waf/latest/developerguide/waf-captcha-puzzle.html.

[6] Amft, S. et al. “Lost and not Found: An Investigation of Recovery Methods for Multi-Factor Authentication”. In: *arXiv:2306.09708.* 2023. arXiv: 2306.09708 [cs.CR].

[7] Bonneau, J. et al. “Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google”. en. In: *WWW ’15.* ACM, 2015. DOI: 10.1145/2736277.2741691.

[8] Campobasso, M. and Allodi, L. “Impersonation-as-a-Service: Characterizing the Emerging Criminal Infrastructure for User Impersonation at Scale”. In: *CCS ’20.* ACM, 2020. DOI: 10.1145/3372297.3417892.

[9] Conners, J. S. and Zappala, D. “Let’s Authenticate: Automated Cryptographic Authentication for the Web with Simple Account Recovery”. In: *WAY ’19.* 2019.

[10] Dropbox. *Change or reset your Dropbox password.* 2023. URL: https://web.archive.org/web/20230518113022/https://help.dropbox.com/security/password-reset.

[11] Federal Bureau of Investigation. *Internet Crime Report 2022.* Mar. 2023. URL: https://web.archive.org/web/20230311011752/https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf.

[12] Freeman, D. et al. “Who Are You? A Statistical Approach to Measuring User Authenticity”. In: *NDSS ’16.* Internet Society, 2016. DOI: 10.14722/ndss.2016.23240.

[13] Garfinkel, S. L. “Design Principles and Patterns for Computer Systems that are Simultaneously Secure and Usable”. PhD thesis. Massachusetts Institute of Technology, 2005.

[14] Gavazzi, A. et al. “A Study of Multi-Factor and Risk-Based Authentication Availability”. In: *USENIX Security ’23.* USENIX Association, 2023.

[15] GOG. *How do I reset my password?* 2023. URL: https://web.archive.org/web/20230317223608/https://support.gog.com/hc/en-us/articles/212185409-How-do-I-reset-my-password-?product=gog.

[16] Golla, M. *I Had a Chat about RBA with @Google in April 2016. The Short Story: "RBA Is an Arms Race, and We Are Not Revealing Any Details That Could Potentially Help Attackers."* Apr. 2019. URL: https://web.archive.org/web/20210812104239/https://twitter.com/m33x/status/1120979096547274752.

[17] Google. *reCAPTCHA v2 | Google Developers.* 2021. URL: https://developers.google.com/recaptcha/docs/display.

[18] Google. *Tips to complete account recovery steps.* 2023. URL: https://web.archive.org/web/20230422113749/https://support.google.com/accounts/answer/7299973.

[19] Hang, A., De Luca, A., and Hussmann, H. "I Know What You Did Last Week! Do You?: Dynamic Security Questions for Fallback Authentication on Smartphones". In: *CHI '15.* ACM, 2015. DOI: 10.1145/2702123.2702131.

[20] Hill, B. "Moving Account Recovery beyond Email and the "Secret" Question". In: *Enigma '17.* USENIX Association, 2017.

[21] Hossen, M. I. et al. "An object detection based solver for google's image recaptcha v2". In: *RAID '20.* USENIX Association, 2020.

[22] Javed, A. et al. "Secure Fallback Authentication and the Trusted Friend Attack". In: *ICDCSW '14.* ACM, 2014. DOI: 10.1109/ICDCSW.2014.30.

[23] Li, Y., Wang, H., and Sun, K. "Email as a master key: Analyzing account recovery in the wild". In: *INFOCOM '18.* IEEE. 2018. DOI: 10.1109/INFOCOM.2018.8486017.

[24] Li, Y. et al. "Understanding Account Recovery in the Wild and its Security Implications". In: *IEEE TDSC* vol. 19, no. 1 (2020). DOI: 10.1109/TDSC.2020.2975789.

[25] LinkedIn. *Password Reset Basics.* 2023. URL: https://web.archive.org/web/20221229120339/https://www.linkedin.com/help/linkedin/answer/a1382101.

[26] Markert, P. et al. "Work in Progress: A Comparative Long-Term Study of Fallback Authentication". In: *USEC '19.* Internet Society, 2019. DOI: 10.14722/usec.2019.23030.

[27] Microsoft Detection and Response Team. *DEV-0537 criminal actor targeting organizations for data exfiltration and destruction.* 2022. URL: https://www.microsoft.com/security/blog/dev-0537.

[28] Milka, G. "Anatomy of Account Takeover". In: *Enigma '18.* Santa Clara, CA: USENIX Association, Jan. 2018. (Visited on 04/18/2019).

[29] MITRE Corporation. *CWE-640: Weak Password Recovery Mechanism for Forgotten Password.* 2021. URL: https://cwe.mitre.org/data/definitions/640.html.

[30] Pöhn, D., Gruschka, N., and Ziegler, L. "Multi-Account Dashboard for Authentication Dependency Analysis". In: *Proceedings of the 17th International Conference on Availability, Reliability and Security*. ARES '22. Vienna, Austria: Association for Computing Machinery, 2022, pp. 1–13. DOI: 10.1145/3538969.3538987.

[31] Quermann, N., Harbach, M., and Dürmuth, M. "The State of User Authentication in the Wild". In: *WAY '18*. Aug. 2018. URL: https://wayworkshop.org/2018/papers/way2018-quermann.pdf.

[32] Sukhani, K. et al. "Automating the bypass of image-based CAPTCHA and assessing security". In: *ICCCNT '21*. IEEE. 2021. DOI: 10.1109/ICCCNT51525.2021.9580020.

[33] Thomas, K. et al. "Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials". In: *CCS '17*. ACM, 2017. DOI: 10.1145/3133956.3134067.

[34] Wiefling, S., Dürmuth, M., and Lo Iacono, L. "More Than Just Good Passwords? A Study on Usability and Security Perceptions of Risk-based Authentication". In: *ACSAC '20*. ACM, 2020. DOI: 10.1145/3427228.3427243.

[35] Wiefling, S., Dürmuth, M., and Lo Iacono, L. "Verify It's You: How Users Perceive Risk-based Authentication". In: *IEEE Security & Privacy* vol. 19, no. 6 (2021). DOI: 10.1109/MSEC.2021.3077954.

[36] Wiefling, S., Dürmuth, M., and Lo Iacono, L. "What's in Score for Website Users: A Data-Driven Long-Term Study on Risk-Based Authentication Characteristics". In: *FC '21*. Springer, 2021. DOI: 10.1007/978-3-662-64331-0_19.

[37] Wiefling, S., Lo Iacono, L., and Dürmuth, M. "Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild". In: *IFIP SEC '19*. Springer, 2019. DOI: 10.1007/978-3-030-22312-0_10.

[38] Wiefling, S. et al. "Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service". In: *ACM TOPS* vol. 26, no. 1 (2023). DOI: 10.1145/3546069.

[39] Yan, J. and El Ahmad, A. S. "Usability of CAPTCHAs or usability issues in CAPTCHA design". In: *Proceedings of the 4th symposium on Usable privacy and security*. 2008, pp. 44–52.

Paper III

# Enhancing FIDO Transaction Confirmation with Structured Data Formats

**Andre Büttner, Nils Gruschka**

**Revision 01**
Date: April 2024
Editor: Andre Büttner

This is an updated version of the paper presented and published at NISK 2021. It includes changes in the format and minor text modifications. These changes do not affect the scope or conclusions of the paper in any way.

Originally published in *NISK Norsk informasjonssikkerhetskonferanse*; Issue No. 3 (2021). https://www.ntnu.no/ojs/index.php/nikt/article/view/5506.

### Abstract

FIDO Transaction Confirmation is an extension for the FIDO authentication protocols to enable the verification and signing of digital transactions, e.g., for online banking. The standard currently considers only to include a transaction message text in the assertion which is signed by the user's authenticator. However, this is not useful for more complex transactions and leaves room for ambiguities that might lead to security vulnerabilities. Therefore, we propose to include the transaction information to the FIDO protocols in a structured data format with a strictly defined schema to validate and sign transactions more reliably and securely.

## III.1 Introduction

In recent years, passwords have proven to be not secure enough to withstand attacks, such as phishing or brute-forcing [3]. Consequently, two-factor and multi-factor authentication have been introduced to make authentication more secure [1]. The FIDO Alliance has proposed protocols for using authenticators as an additional factor and even as a passwordless solution. An important extension to these protocols is the *Transaction Confirmation* [2], which allows users to confirm online transactions using a FIDO authenticator. A relying party can include a transaction message or an image to an assertion request, which is displayed to the user and signed by the authenticator. However, research has shown that it is possible to trick a user into approving a malicious transaction [9, 10].

Further, since the transaction is only represented as a text string or an image without clearly defined semantics, the transaction information leaves room for ambiguities. Therefore, the desirable *What-You-See-Is-What-You-Sign* [7] property is not sufficiently fulfilled. It would be more reliable to use a structured data format that contains a well-formed and self-describing representation of a transaction [4, 6]. Therefore, this paper's contribution is a proposal and discussion on the use of structured data formats for FIDO Transaction Confirmation.

The remainder of this paper is structured as follows. In Section III.2 some background on FIDO Transaction Confirmation is provided. Our proposed enhancement for the FIDO transaction extension is described in Section III.3. Section III.4 discusses the advantages and disadvantages of our approach. Finally, in Section III.5, our findings are concluded, and suggestions on future work are given.

## III.2 FIDO Transaction Confirmation

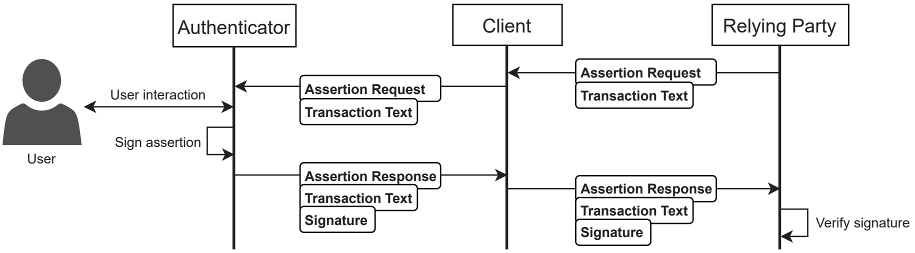


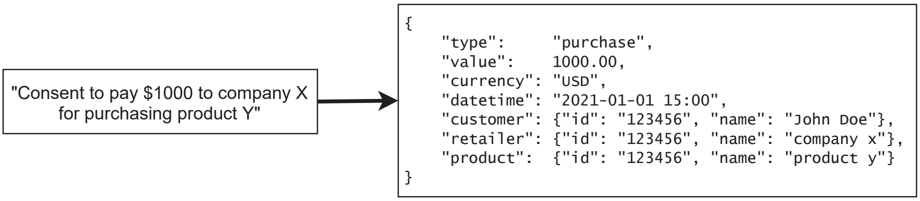
Figure III.1: Transaction Confirmation flow diagram showing the different processing steps.

The FIDO UAF and FIDO2/WebAuthn protocols are based on a challenge-response protocol, where an authenticator, e.g., a smartphone, hardware token or platform authenticator, registers with a public-key against a relying party. For authentication, the authenticator needs to sign a random challenge to prove possession of the corresponding private key.

Transaction Confirmation as an extension of these protocols seeks “a standardized and secure way of gathering explicit user consent for a specific action” [2]. Consent is based on the user's interaction with the respective authenticator to confirm that he has seen and approved the transaction message. This allows the use of FIDO authenticators for carrying out bank transactions, online purchases, granting access to certain information, and more.

Figure III.1 gives an overview of how a transaction is processed with the FIDO protocols. The relying party sends a FIDO assertion request to the client, which contains a human-readable representation of a transaction in the form of a simple text. The user confirms the transaction by interacting with the authenticator. Afterwards, the authenticator creates the assertion response along with the signature created with the corresponding private key. The assertion response is then returned via the client application to the relying party, which finally verifies the signature and executes the requested transaction [5].

## III.3 Structured Data for Transactions


"Consent to pay $1000 to company X for purchasing product Y"

```
{
    "type":     "purchase",
    "value":    1000.00,
    "currency": "USD",
    "datetime": "2021-01-01 15:00",
    "customer": {"id": "123456", "name": "John Doe"},
    "retailer": {"id": "123456", "name": "company x"},
    "product":  {"id": "123456", "name": "product y"}
}
```

Figure III.2: Example transaction as plain text and structured data.

Instead of just plain text, we propose to use a machine-readable representation of a transaction that is converted into a human-readable text by the client or authenticator. One common data format for structuring data is the Extensible Markup Language (XML), which is typically defined and validated using the XML Schema Definition (XSD) language. Also, there are respective W3C standards for signature generation and encryption.

The data formats used in the FIDO protocols are JavaScript Object Notation (JSON) and its binary counterpart Concise Binary Object Representation (CBOR). FIDO extensions are expected to be in the CBOR format. Thus, we consider this data format to be most suitable for transaction data structures as well. Similar to XSD, there already exists the Concise Data Definition Language (CDDL) which analogously enables the definition and validation of CBOR objects. Signatures, message authentication and encryption are standardized in the CBOR Object Signing and Encryption (COSE) protocol. In Figure III.2 on the left-hand side, an example mentioned in [2] is shown. With our approach, this can be replaced by a semi-structured representation as presented on the right-hand side. Some information like identifiers and time were added, showing how transactions could easily be extended with relevant information. Also, validation and limitations on each of the attributes could be applied by the authenticator. Further aspects are discussed in the following section.

## III.4 Discussion

Semi-structured data formats like XML, JSON or CBOR provide properties, e.g., well-formedness and being self-describing with clear semantics [8]. This avoids ambiguities from unclear formulations, which is common for plain text. Further, for more complex types of transactions, it might be useful to display only relevant parts to the user before signing. This can be realized more easily with structured data, if these parts are separate attributes inside the data structure, e.g., an account number inside a bank transaction. Also, structured data is machine-readable, which allows to define policies for certain attributes. These can be provided as CDDL schemas by the relying party during registration, which are then used by the client application or the authenticator for validation.

FIDO transactions may be manipulated or eavesdropped through XSS or malware on the client. Therefore it is reasonable to let the relying party sign [9] and encrypt the transaction data. If the CBOR format is used for transactions, the COSE protocol can provide a standardized way of ensuring both integrity and confidentiality on both ends.

An obvious disadvantage of using data structures for FIDO transactions is the complexity and its data overhead. This may especially be problematic for hardware tokens with limited computation and storage resources. The authenticator would need to perform CDDL schema validation. Ideally, it should also support the COSE signature validation and decryption. Increased latency may be acceptable since registration and normal authentication would not be affected. In case it does not work on hardware tokens, the validation can be outsourced to the client application, however, reducing the security gain.

## III.5 Conclusion and Future Work

The FIDO protocols are a promising step towards more secure authentication and a potential replacement for passwords. Transaction Confirmation is a good example of how these protocols support use cases beyond that. This paper addresses some shortcomings of this extension and proposes to use structured data instead of plain text. As discussed, our approach provides many opportunities, such as allowing an authenticator to validate transactions against policies and using standardized ways to ensure integrity and confidentiality.

In future work, we are planning to test the approach for different applications and different types of authenticators, to analyze different attack scenarios and to evaluate the application of CDDL schemas and COSE signature and encryption.

## References

[1] Dasgupta, D., Roy, A., and Nag, A. “Multi-Factor Authentication”. In: *Advances in User Authentication*. Cham: Springer International Publishing, 2017, pp. 185–233.

[2] FIDO Alliance. *FIDO Transaction Confirmation White Paper*. Tech. rep. Aug. 2020. URL: https://media.fidoalliance.org/wp-content/uploads/2020/08/FIDO-Alliance-Transaction-Confirmation-White-Paper-08-18-DM.pdf.

[3] Florêncio, D., Herley, C., and Coskun, B. “Do strong web passwords accomplish anything?” In: *HotSec* vol. 7, no. 6 (2007), p. 159.

[4] Gruschka, N., Reuter, F., and Luttenberger, N. “Checking and Signing XML Documents on Java Smart Cards”. In: *Smart Card Research and Advanced Applications VI*. Ed. by Quisquater, J.-J. et al. Boston, MA: Springer US, 2004, pp. 287–302.

[5] Hodges, J. et al. *Web Authentication: An API for accessing Public Key Credentials Level 1*. W3C Recommendation. W3C, Mar. 2019. URL: https://www.w3.org/TR/2019/REC-webauthn-1-20190304/.

[6] Jøsang, A. and AlFayyadh, B. “Robust WYSIWYS: A Method for Ensuring That What You See is What You Sign”. In: *Proceedings of the Sixth Australasian Conference on Information Security - Volume 81*. AISC ’08. Wollongong, NSW, Australia: Australian Computer Society, Inc., 2008, pp. 53–58.

[7] Landrock, P. and Pedersen, T. “WYSIWYS?—What you see is what you sign?” In: *Information Security Technical Report* vol. 3, no. 2 (1998), pp. 55–61.

[8] Nenadi, A. and Zhang, N. “Non-repudiation and Fairness in Electronic Data Exchange”. In: *Enterprise Information Systems V*. Springer, 2004, pp. 286–293.

[9] Xu, P. et al. “SDD: A trusted display of FIDO2 transaction confirmation without trusted execution environment”. In: *Future Generation Computer Systems* vol. 125 (2021), pp. 32–40.

[10] Zhang, Y. et al. “Secure display for FIDO transaction confirmation”. In: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018, pp. 155–157.

Paper IV

# Protecting FIDO Extensions against Man-in-the-Middle Attacks

**Andre Büttner, Nils Gruschka**

**Revision 01**
Date: April 2024
Editor: Andre Büttner

This is an updated version of the paper presented and published at ETAA 2022. It includes changes in the format and minor text modifications. These changes do not affect the scope or conclusions of the paper in any way.

Originally published in *Emerging Technologies for Authorization and Authentication. ETAA 2022*; ISBN 978-3-031-25467-3; pages 70–87; Springer Nature Switzerland, Cham (2023). DOI: 10.1007/978-3-031-25467-3_5.

### Abstract

FIDO authentication has many advantages over password-based authentication, since it relies on proof of possession of a security key. It eliminates the need to remember long passwords and, in particular, is resistant to phishing attacks. Beyond that, the FIDO protocols consider protocol extensions for more advanced use cases such as online transactions. FIDO extensions, however, are not well protected from Man-in-the-Middle (MitM) attacks. This is because the specifications require a secure transport between client and server, but there exists no end-to-end protection between server and authenticator.

In this paper, we discuss MitM scenarios in which FIDO extensions may be intercepted. We further propose an application-layer security protocol based on the CBOR Object Signing and Encryption (COSE) standard to mitigate these threats. This protocol was verified in a formal security evaluation using ProVerif and, finally, implemented in a proof-of-concept.

## IV.1 Introduction

In today’s digital era, almost everybody is used to logging in to a website with a password. Although passwords are easy to use, they have many disadvantages in terms of security. They are often easy to guess or sometimes publicly disclosed after a data breach. Furthermore, passwords are vulnerable to phishing attacks. Therefore, many services already implement multi-factor authentication.

FIDO authentication is a relatively young technology that intends to overcome the disadvantages of passwords. The basic idea behind it is to use an authenticator device as a more secure authentication factor, either in addition to or even as a replacement for passwords. A feature that is rarely used yet but may become important soon is FIDO extensions. These allow for more advanced functionality beyond simple authentication. FIDO authentication may, for example, be used to confirm online purchases or banking transactions. Initially, a FIDO Transaction Confirmation extension was proposed, which includes a human-readable text representation of a transaction as an extension [15]. This extension, however, was never implemented and already became deprecated. It is replaced by the more recent Secure Payment Confirmation [27]. We expect to see more different kinds of extensions like this in the future.

However, the FIDO specifications do not provide any specific protection for FIDO extensions. Since extensions can contain very sensitive information, it should be ensured that attackers cannot intercept or manipulate this information. There are several possibilities for attackers to act as Man-in-the-Middle (MitM). The FIDO protocols only protect the integrity of messages from the authenticator to the server. The integrity of messages from the server cannot be checked by the authenticator. While this is not necessary for basic authentication, it may be crucial for certain extensions. Also, confidentiality cannot be guaranteed as there is no encryption on the application layer.

To mitigate the risk of manipulation or disclosure of FIDO extensions, we propose to apply authenticated encryption to FIDO extensions. In this paper, we provide the following contributions:

1. An overview of different MitM attack scenarios against FIDO extensions.
2. A proposal for a security protocol to protect FIDO extensions.
3. A formal security verification of this protocol.
4. A proof-of-concept implementation.

The remainder of this paper is structured as follows. Section IV.2 gives an overview of FIDO authentication and the COSE protocol. In Section IV.3 related literature on FIDO is presented. The attack model addressed in this paper is explained in Section IV.4. In Section IV.5 we specify our proposed security protocol, which is evaluated in Section IV.6. In Section IV.7 we describe a proof-of-concept implementation. A discussion of the proposed solution is provided in Section IV.8. Our findings are summarized in Section IV.9 along with a brief outlook on future work.

## IV.2 Background

In this section, we first provide some background information on FIDO authentication. Afterward, the CBOR-based COSE protocol is described.

### IV.2.1 FIDO authentication

The Fast IDentity Online (FIDO) Alliance has been publicly active since 2013 [16]. Today, it includes members from several popular Internet companies. Their main objective is to provide industry standards for using authenticators to authenticate against web applications either as a single factor (password-less) or as an additional factor (2FA/MFA). There are two different types of authenticators. *Roaming authenticators* are external security tokens (for example, a YubiKey) that can be connected via USB, Bluetooth-Low-Energy (BLE) or Near-Field-Communication (NFC). In contrast to this, *platform authenticators* are integrated into client devices like computers and smartphones.

In this paper, we mainly focus on FIDO2, which consists of the Web Authentication (WebAuthn) API and the Client-to-Authenticator-Protocol 2 (CTAP2) [14]. WebAuthn has become a W3C standard [22] and defines a JavaScript API and data structures that can be used to create credentials and get assertions from the authenticator. CTAP2 defines the protocol between the client platform and the authenticator.

For FIDO authentication, security and trust are based on public key cryptography. Figure IV.1 gives a brief overview of the different roles and messages that are specified for FIDO2 authentication. At first, an authenticator (e.g., a security token) needs to be registered at a web service, the so-called relying party (RP). When a user registers at an RP, the RP first sends a registration request to the authenticator which includes a random nonce (challenge), its identifier (rpId), and further parameters. The authenticator creates a credential key pair and shares its public key with the RP by sending a registration response. In addition, the authenticator may include an attestation certificate that verifies the origin of the authenticator by a certified manufacturer. For this purpose, the FIDO Alliance provides a public service called *Metadata Service* [13], which contains a list of vendors, their public keys, and their certification levels. During authentication, the RP creates another challenge value and sends it to the authenticator in an assertion request. This challenge is signed by the authenticator, along with other parameters, using the private key of the credential that was previously registered with the RP. Using the corresponding public key, the RP can verify the signature of the assertion response. Both for registration and authentication, the user needs to interact with the authenticator, e.g., by pressing a button. For more security, the user can enter a PIN or interact with a biometric scanner, which provides an additional authentication factor.

The FIDO specifications leave room for additional features by using protocol extensions. Extensions sent by the RP to the authenticator are called *input extensions* and extensions from the authenticator to the RP *output extensions*. Furthermore, it is distinguished between *client extensions* and *authenticator extensions*. In this paper, we focus on authenticator extensions, i.e., those that are processed by the authenticator. Several different types of extensions have been proposed (see e.g. [20]); however, almost none of these have been implemented yet. The Secure Payment Confirmation [27] is a new W3C specification and describes a payment extension for FIDO authentication. It is a good example of

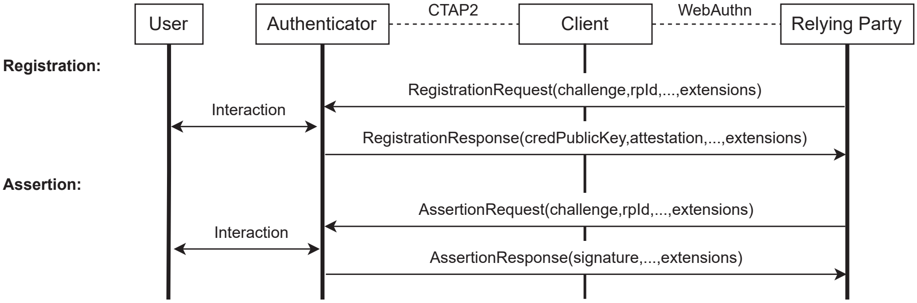

User
Authenticator
CTAP2
Client
WebAuthn
Relying Party
Registration:
RegistrationRequest(challenge,rpId,...,extensions)
Interaction
RegistrationResponse(credPublicKey,attestation,...,extensions)
Assertion:
AssertionRequest(challenge,rpId,...,extensions)
Interaction
AssertionResponse(signature,...,extensions)

Figure IV.1: FIDO authentication overview.

a more advanced application of FIDO authentication. However, it must be kept in mind that such an extension also requires high security standards.

### IV.2.2 COSE

The *CBOR Object Signing and Encryption* (COSE) [33] protocol aims to provide a standard for exchanging signed and encrypted data in the *Concise Binary Object Representation* (CBOR) [6] format. CBOR is a binary data format that is particularly useful for low-resource devices due to its lightweight and efficient design. Among other things, it is used in the CTAP2 protocol. Data can be structured into maps and arrays of various types. Consequently, JSON objects can be easily converted to CBOR objects, which makes it also usable from a developer's perspective. Furthermore, CBOR provides features like tags or flexible map and array lengths.

COSE is basically adapted from the JavaScript Object Signing and Encryption (JOSE) protocol. It defines data structures for exchanging data that is signed, encrypted or authenticated (MAC). COSE objects carry the payload together with additional information about the keys and algorithms that are used. A COSE message is composed of a CBOR array that contains a protected header, an unprotected header, the payload, and, depending on the type, additional values like the signature or the message authentication tag. A protected header is a CBOR-encoded map of certain header values. It is used as input in addition to the actual payload for cryptographic functions, e.g., as additional authenticated data (AAD) when using authenticated encryption or as input for the signature. The unprotected header is a map that contains further header values, which, in contrast to the protected header, are not cryptographically bound to the payload or signature. COSE signature messages may contain one (`COSE_Sign1`) or multiple signatures (`COSE_Sign`). Encryption messages can be intended for one recipient (`COSE_Encrypt0`) or for multiple recipients (`COSE_Encrypt`). Respectively, there are also two different COSE MAC structures.

The COSE protocol does not specify, how keys are negotiated by the different parties. However, it defines a *COSE Key* structure, which contains all necessary information for a key. This can be useful, e.g., for storing the key or for sending it to another party in a standardized manner. For example, the FIDO2 protocols make use of the COSE Key format to send the public key from the authenticator to the RP. There currently exists a draft for a COSE-based *Ephemeral Diffie-Hellman Over COSE* (EDHOC) protocol to provide additional features like key negotiation [34], which, however, is not standardized yet.

## IV.3 Related Work

Since FIDO authentication is a fairly new topic, research on the subject is still very limited. We, therefore, provide a brief overview of the related literature.

There has been quite some research on the usability of FIDO authentication [28, 29]. One of the major concerns of the users is the account recovery. If the authenticator gets lost, there must be some way to regain access to the account. At the same time, the recovery option should not reduce the security. As a solution, an enhancement for the protocol was proposed that enables the use of a backup key that only needs to be configured once in the beginning [17]. Furthermore, a study on different account recovery approaches was conducted to compare them in terms of usability, deployability and security [23]. Other researchers applied formal methods to analyze the security of the FIDO protocols [3, 11]. In particular, there still seems to be a lack of research that focuses on the CTAP2 protocol.

There has also been done little research specifically on the security of FIDO extensions. For example, it was proposed to use structured data formats for FIDO Transaction Confirmation to facilitate the validation of transaction information by the authenticator thereby making it more secure [8]. Furthermore, some researchers have pointed out that the FIDO Transaction Confirmation extension is vulnerable to manipulation. They propose to let an RP sign the transaction information, which can be validated on the client side in a trusted environment [37, 38]. However, they do not point out how the authenticity of the public key is guaranteed. In addition to this, we see further risks. If FIDO extensions can be manipulated, they can also be eavesdropped in similar attack scenarios. Therefore, confidentiality should be equally considered alongside integrity and authenticity.

## IV.4 Attacker Model

The WebAuthn standard [22] requires a so-called secure context, which includes the use of HTTP over TLS (HTTPS). This ensures that the client can verify the authenticity of a web server. Yet, there are more components involved that can interfere with FIDO messages beyond client and server. We, therefore, argue that HTTPS does not provide sufficient protection for FIDO messages at all. FIDO authentication can involve several different intermediaries between the RP and the authenticator. These include (1) web proxies between the client and the RP, (2) the client application, (3) intermediary processes on the client platform

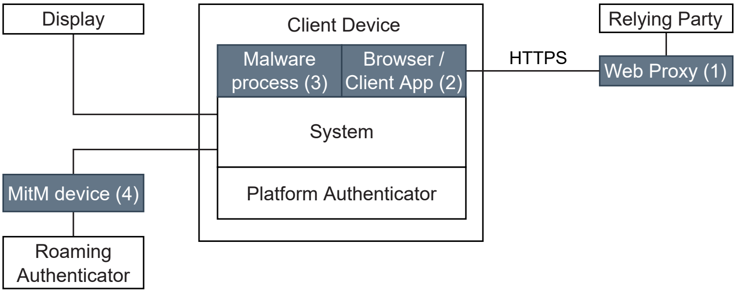


Figure IV.2: Attack surface: possible points of interception are highlighted.

and (4) hardware between a roaming authenticator and the client device. Thus, there is a significantly large attack surface, as illustrated in Figure IV.2.

Plain authentication—also referred to as *entity authentication*—without any extensions is not likely at risk because the protocol is designed not to contain sensitive information. Also, it is resistant to manipulation through the challenge-response protocol. However, authentication that involves extensions exchanged between the RP and the authenticator may contain valuable information, e.g., personal data, transaction details or other sensitive information. Such information could be obtained or manipulated by an adversary. In the following, we elaborate on the four different MitM scenarios in more detail.

### IV.4.1 Vulnerable web intermediaries

Distributed systems like web applications typically include intermediaries such as content delivery networks (CDN), load balancers, or web application firewalls (WAF). The secure context requirement mentioned above can only be verified for the connection between a browser and the next HTTP entity. As a consequence, it cannot be guaranteed that intermediaries communicate with other intermediaries or the server via HTTPS. Apart from that, HTTPS is only protecting on the transport layer. Web intermediaries usually operate on the application layer and, therefore, need to access HTTP messages, including the body. Consequently, they are able to read FIDO messages in clear text.

If a proxy behaves maliciously, this can have severe security implications. A proxy could be misused to intentionally read FIDO messages and disclose sensitive information. Beyond that, a malicious proxy could manipulate extensions. There is no integrity check considered for FIDO messages from the RP to the authenticator. Depending on the type or use of an extension, only a user may notice the manipulation through manual inspection. For messages from the authenticator to the RP, any manipulation will be detected by the RP since the authenticator data, including extensions, are signed. In any case, there is still the risk of information disclosure.

Another concern with HTTP intermediaries is the possibility of attacks that result from the semantic gap of the HTTP protocol [9]. In particular, cache poisoning vulnerabilities could be exploited to disclose FIDO messages. This can be realized by various techniques like request smuggling [26] or web cache deception [18].

### IV.4.2 Compromised client application

The client application on the user's device may be running in a browser as a JavaScript application or a native mobile application. Both browser clients and native mobile applications often use third-party libraries. If not checked properly, such libraries can include malicious code [2, 39]. Another possibility to compromise the client application is to exploit cross-site scripting (XSS) vulnerabilities in a JavaScript application to inject code that intercepts FIDO messages and modifies extensions or forwards them to an untrustworthy third party.

### IV.4.3 Malware on the client device

Malware can pose a further threat to FIDO extensions. An attacker may be able to install malicious software on a user's client device through an email attachment or some other exploit. By intercepting inter-process communication (IPC) or accessing the memory of other processes, the malware could read or manipulate FIDO messages. Moreover, it could bypass security measures by the browser and system and send its own FIDO assertions to the authenticator. It was already shown that this could cause a user to confirm a malicious transaction [7]. In addition, there may be specific types of viruses targeting browsers on client devices. By this, a browser may be corrupted in a way that it can be controlled by an attacker, which is also known as Man-in-the-Browser (MitB) attacks [10]. Beyond that, platform authenticators are generally at risk of behaving unintentionally when they are affected by malware. This can be mitigated with the use of trusted platform modules (TPM), which make sure that secret keys are not disclosed. Nevertheless, exploits against extensions remain a threat.

### IV.4.4 MitM between client device and authenticator

With respect to roaming authenticators, an attacker may try to intercept the connection between the client device and the authenticator. This is certainly a more difficult attack since an attacker needs physical access to the user's devices. One could argue that the security of the FIDO device is completely compromised in that case, and other MitM countermeasures would be pointless. However, this is only true for authenticators that just require a button press and not when the authenticator uses a more secure verification method such as biometrics.

Even if an authenticator uses a verification method like biometrics, an attacker may still be able to intercept the connection and eavesdrop or manipulate extensions. For example, there are known MitM attacks against Bluetooth [24, 35]. NFC is very unlikely to be intercepted without the owner's awareness. But still, a potential attack against NFC has been demonstrated [1].

## IV.5 Protocol design

As shown in the previous section, FIDO messages can indeed be vulnerable to several attacks. Extensions may include sensitive information and are at risk of being modified by or disclosed to unauthorized parties. Considering the large attack surface, we see the necessity to apply further measures. In this section, we present our proposed protocol to protect FIDO extensions.

### IV.5.1 Authenticated encryption

Sensitive FIDO extensions should provide secure properties such as confidentiality, integrity, and authenticity. Messages sent from the authenticator to the RP are already signed and, thus, do not require any additional authentication. However, messages from the RP to the authenticator are neither signed nor authenticated by any means. Xu *et al.* [37] suggest that the relying party shall sign extensions. In their approach, the verification of the signature is done on the client device, and the public key for verifying the signature is given by the TLS connection. We, however, want to enable the authenticator itself to validate the authenticity of extensions from the RP.

Since signatures only provide integrity and authenticity but no confidentiality, we propose to fulfill all these properties by using authenticated encryption (AE) instead. For this, the RP and the authenticator must first agree upon a shared secret during registration. After that, they can derive key material from the shared secret and use AE algorithms such as AES-GCM to encrypt and authenticate extensions that are included in assertion messages.

There can be multiple authenticators registered with one user account on the RP. However, there will be a different shared key between the RP and every authenticator. The RP does not know which registered authenticator will be used for the assertion. Therefore, we need to apply *key wrapping*. This means that the actual extension is encrypted with a newly created content encryption key. This key is then encrypted with the shared key and appended to the message for each authenticator. For encrypting the extensions in the assertion response by the authenticator, the shared key can be used directly, because the message is only intended for the RP. This is formalized in our model in Section IV.6.2.

### IV.5.2 Key exchange

Encrypting FIDO extensions requires the RP and the authenticator to exchange keys in advance. Normally, a public-key infrastructure (PKI) is used to create certificates that provide trust and authenticity for exchanged keys. Hardware tokens, however, are very limited and likely not powerful enough in terms of storage and computation to handle certificate chains. Since they operate offline, there is also no possibility for them to interact directly with certificate authorities over the network (e.g. to check on certificate revocations). This is different for other types of authenticators with more computing resources and networking capabilities. However, we want to address all types of authenticators with our solution. Because of this, we consider the *trust-on-first-use* authentication scheme [36]. This means that we trust the first connection between the authenticator and RP not being intercepted by an adversary.

To generate a shared secret, the RP and the authenticator perform a Diffie-Hellman key exchange (DHKE) during the registration. The RP includes the first part of the DHKE as a registration input extension. The authenticator generates and stores the shared secret from the DHKE and sends a registration response to the RP, which includes the second part of the DHKE as a registration output extension. Finally, the RP generates the shared secret from the DHKE and stores it together with the newly registered credential.

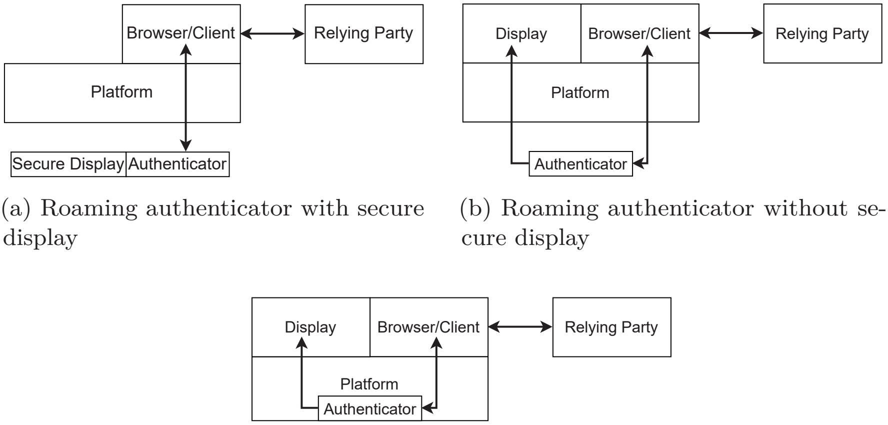
The (unauthenticated) DHKE is known to be secure against eavesdropping but vulnerable to active MitM attacks. Authenticators use attestations that should be validated by RPs to create a certain amount of trust. If properly done, this can mitigate active MitM attacks. However, for higher security, it is important that the authenticator includes both parts of the DHKE in its attestation signature, as shown in Sect IV.6.1.

### IV.5.3 Data format

A further important aspect is the format used to exchange the encrypted data along with required metadata like an input vector (IV) and the algorithm used. FIDO authenticator extensions must be provided in the CBOR format. As described in Section IV.2.2, the accompanying protocol for signature and encryption is COSE. Since the public key from the authenticator is transmitted as a COSE key, authenticators and RPs are both supporting parts of the COSE protocol already. The COSE standard supports encryption for single and multiple recipients and thus provides all functionality needed for the proposed protocol. Our suggestion is therefore to embed extensions in COSE structures. The COSE key format can also be used to encode the DH public keys that are exchanged during the registration to generate the shared secret.

### IV.5.4 Displaying user information

When encrypting extensions, we still need to be able to display information, such as transaction information, to the user in a secure manner. This is the key aspect of the What-You-See-Is-What-You-Sign principle [25]. The different possible architectures with FIDO authenticators are displayed in Figure IV.3. Ideally, an authenticator should provide a secure display (Figure IV.3a). However, there are no roaming authenticators with a display on the market yet. In most cases, the client platform would be responsible for displaying the information to the user. With our approach, the client will not be able to decrypt the extensions on transit. Therefore, the authenticator first needs to decrypt the extensions and then forward the user information to the client display on a secure path



(a) Roaming authenticator with secure display

(b) Roaming authenticator without secure display

(c) Platform authenticator

Figure IV.3: Architectures for different types of authenticators.

(Figure IV.3b). For this, the platform should provide appropriate measures to ensure that no interception is possible when displaying the information to the user.

Platform authenticators are integrated into computers or smartphones. Since these already have a display, platform authenticators can provide the decrypted user information instantly to the platform without an intermediary connection (Figure IV.3c). FIDO authentication is already supported by most platforms like Windows [21], MacOS/iOS [30] and Android [19]. The platform itself must ensure that the information that is displayed to the user has not been modified by another malicious process. When the client device is responsible for displaying the information, there are further risks like UI deception attacks [4, 12], which, however, must be taken care of by the operating system. The same applies when a roaming authenticator without a secure display is used.

## IV.6 Security Evaluation

ProVerif [5] is a common protocol verifier that uses Horn-clause-based representations of a protocol and the applied $\pi$-calculus for process verification. A formal model of the protocol and the security properties to be tested are defined in input files. The output is the test results indicating whether the defined security properties are met. If a test fails, a possible attack trace is provided. This tool has been used to conduct a security evaluation of our protocol.

Formal models have been created for the key exchange during the registration of an authenticator and for the exchange of encrypted and authenticated extensions. A basic description of the models, some code excerpts and the results are given below. For more details, the sources and results of the evaluation can be found in our Github repository[1].

## IV.6.1 Key exchange

In our protocol, a DHKE is performed to generate a shared secret on the relying party and the authenticator. Even though a client can verify the TLS certificate of an RP, we assume that an authenticator cannot validate the authenticity of an RP. However, we consider that an RP requires attestation from the authenticator and that it verifies the attestation signature with a known and trusted public key. This serves to validate the authenticity of a DH key that is received by the RP to compute the shared secret.

One obvious security requirement is that the shared secret is kept secret and cannot be obtained by an attacker. This is defined by the following two queries:

```
query attacker(computeSecret(publicKey(dh_priv_AU),dh_priv_RP)).
query attacker(computeSecret(publicKey(dh_priv_RP),dh_priv_AU)).
```

Further, we check the authenticity by verifying that a key exchange is only performed if both the authenticator and RP have generated the same secret:

```
query x:G; inj-event(sharedSecretRP(x))
    ==>inj-event(sharedSecretAU(x)).
```

For the key exchange, two different models of the protocol were created, because the first model did not pass the verification.

### IV.6.1.1 Protocol model 1

In this model, the authenticator creates an attestation signature including the nonce, the credential public key and the output extension containing its DH key:

```
sign((nonce,credPubKey,dh_pub_AU),privKeyAttestation)
```

This signature is verified by the RP against the original nonce, the credential public key and the extension with the DH key by the authenticator:

```
checksign((nonce,credPubKey,dh_pub_AU),signature,pubKeyAttestation)
```

When verifying this model with ProVerif, the authenticity test fails. The detailed output of ProVerif contains a trace where an attacker intercepts a registration request by the RP and replaces the DH key of the RP with its own key. Because of this, the authenticator will compute a different shared secret than the RP. As a consequence, the authenticator cannot authenticate or decrypt assertion extensions from the RP but from the attacker. However, the attacker cannot gain much from this, except for causing a denial of service. Nonetheless, this attack should be avoided.

[1] https://github.com/Digital-Security-Lab/protecting-fido-extensions-proverif (Last accessed: 2024-04-24)

#### IV.6.1.2 Protocol model 2

In the second model, the authenticator includes both DH keys in the attestation signature so the RP can verify that the same shared secret is computed on both ends. From a theoretical perspective, it would be enough only to modify the signature. However, in practice, the protocol proposed here should be compatible with the FIDO standards. Therefore, the authenticator will have to include both its own DH key and the DH key from the RP in the output extensions so both keys are implicitly included in the signature:

```
sign((nonce,credPubKey,dh_pub_AU,dh_pub_RP),privKeyAttestation)
```

The same is true when verifying the signature. In particular, the RP must check the signature with its own generated DH key and the authenticator's key:

```
checksign((nonce,credPubKey,dh_pub_AU,dh_pub_RP),signature,pubKeyAttestation)
```

With this model, all tests succeed, and the secrecy of the shared secret is guaranteed, as well as the authenticity of the public keys that were exchanged. We, therefore, consider this model in our final solution.

### IV.6.2 Encrypted assertion extensions

The second critical part of the proposed protocol is the exchange of encrypted and authenticated extensions between RP and authenticator during assertions. We make the following assumptions. First, the authenticator is successfully registered with the RP. This means that the RP has the credential public key of the authenticator to verify its signature, and both the authenticator and RP have exchanged a shared secret and derived from it a shared key. Second, while replay attacks against the RP are prevented by the nonce, replay attacks against the authenticator are not.

A security requirement here is the secrecy of input and output extensions, which is defined in the following two queries:

```
query attacker(AssertionInputExtensions).
query attacker(AssertionOutputExtensions).
```

In addition, the authenticator and the RP should both only accept authenticated extensions. An attacker must not be able to forge or manipulate extensions in a way that they are processed by either of them. As mentioned above, we assume an attacker to be able to replay assertion messages to the authenticator, but not to the RP. Therefore only events in connection with output extensions are defined as injective events:

```
query x:bitstring; event(receiveInputExtensionsAU(x))
    ==>event(sendInputExtensionsRP(x)).
query x:bitstring; inj-event(receiveOutputExtensionsRP(x))
    ==>inj-event(sendOutputExtensionsAU(x)).
```

This time only one model had to be created. In this model, the RP uses key wrapping to transmit a content encryption key together with the encrypted input extensions:

```
new cek:key;
let inputExtensions_enc = senc(AssertionInputExtensions,cek) in
let cek_enc = senc(key2Bitstring(cek), sharedKey) in
out(c,(nonceRP,cek_enc,inputExtensions_enc))
```

The authenticator, on the other hand, uses the shared key directly to encrypt the output extensions:

```
let outputExtensions_enc = senc(AssertionOutputExtensions,sharedKey) in
```

The evaluation of this model with ProVerif indicates that the expected security requirements are met and no attacks have been found. Hence, with this model, we can successfully exchange FIDO extensions while preserving their confidentiality, integrity, and authenticity.

## IV.7 Implementation

A proof-of-concept (PoC) application has been developed to demonstrate how to implement essential parts of the proposed protocol. The sources and further instructions can be found in a Github repository[2]. Since FIDO keys can be, e.g., USB devices with very limited resources, it was decided to provide a test application using the C programming language and libraries that are optimized for embedded devices. To implement the protocol, a CBOR library is needed, which is already included in each FIDO component, as it is required for implementing the basic FIDO protocols. Moreover, a COSE library is needed. Since we could not find a useful implementation, we developed an open-source COSE library[3] based on the RFC 8152 standard [33]. At the time of writing, this library is still a work in progress but already provides everything needed for the proposed protocol. Finally, additional crypto libraries may be needed to do certain operations such as generating private and public keys, to compute the shared secret from the DHKE and to derive key material using e.g. a Hash Key Derivation Function (HKDF).

The PoC application is meant to demonstrate the parts that have to be implemented in addition to the FIDO protocols. Basic features such as credential creation, attestation, and signature verification are therefore not included. For the DHKE, elliptic curve key pairs were used. The authenticated encryption is done using AES-GCM with a 128-bit key. In the example application, it is first shown how the RP and the authenticator exchange a shared secret. The RP creates the first part of the DHKE, which is encoded in a COSE Key structure and transmitted to the authenticator. The authenticator then creates the second part of the DHKE, computes the shared secret and derives from it a 128-bit key using a HKDF with SHA-256 as underlying hash function. Subsequently, the RP receives the second part of the DHKE (in a real-world application together with the first part of the DHKE as discussed in Section IV.6.1) and analogously computes the shared secret and derives from it a key the same way as the

[2] https://github.com/Digital-Security-Lab/protecting-fido-extensions-poc (Last accessed: 2024-04-24)

[3] https://github.com/abuettner/cose-lib (Last accessed: 2024-04-24)

authenticator. In the second part of the application, the RP is provided with the credential identifier and the shared key of an authenticator. The RP creates an encoded COSE Encrypt structure that contains an extension value encrypted with a content encryption key. This key is then encrypted using the shared key and attached as a recipient object. The credential id of the corresponding authenticator is used as key identifier. The authenticator receives this COSE Encrypt structure and identifies that a recipient is attached to its credential id. It can then decrypt the content encryption key and, finally, the extension value. The authenticator then creates an encoded COSE Encrypt0 structure that contains another extension value, this time encrypted with the shared key. The RP receives this structure and finally decrypts the extension value by the authenticator. Note that in a real application, the RP can identify the shared key used by the credential id that is part of the authenticator data.

Preliminary measurements on a Raspberry Pi Pico (264 kB SRAM, 2 MB on-board flash memory) [31] show that the steps performed by an authenticator during the registration take about 250 milliseconds, while the steps during the assertion take about 5 milliseconds. The additional delay during registration is acceptable since this is performed only once per application. The additional runtime on assertions would be unnoticeable by the user.

## IV.8 Discussion

In this section, we discuss our proposed protocol for protecting FIDO extensions with regard to several different aspects.

### IV.8.1 Security

There are several different ways to intercept FIDO messages in clear text, as described in Section IV.4. This allows an attacker to intercept FIDO extensions with valuable information and either eavesdrop or manipulate them. As shown by the evaluation, the security of FIDO extensions can be significantly improved with our proposed solution. However, the security of extensions also depends on a secure key exchange. While RPs can verify the attestation to get information on security properties provided by an authenticator to create trust, authenticators cannot reliably verify the origin of an RP. In our formal models, the client between the RP and the authenticator has not been considered. It could be argued that the client adds security to some extent, e.g., by validating the TLS certificate of the server. Yet, this is not sufficient and additional measures as proposed in this paper are justified.

The security also depends on the strength of cryptographic algorithms. This has not been evaluated within this work. We consider cryptographic algorithms that are widely accepted and used e.g. in the most recent TLS 1.3 [32]. However, the proposed protocol is meant to be generic, so cryptographic algorithms can simply be replaced if necessary (e.g. with post-quantum cryptography).

### IV.8.2 Implementation

We provide an example of how our protocol can be implemented. Our protocol is completely compatible with the FIDO standards. While the protocol is quite complex, our implementation can be used to integrate it into FIDO applications with low effort. Since there are not too many COSE library implementations, a further contribution of this work is such a COSE library which can be used by any other C application.

At the time of writing, FIDO implementations are quite restricted to standardized extensions. Even though the WebAuthn standard [22] defines how arbitrary extensions should be forwarded to the authenticator, browsers have not implemented this. This means that custom extensions are not passed to FIDO devices. It is therefore challenging to implement a real-world example at this stage. Our protocol should also be considered for extensions that will be standardized in the future, such as the Secure Payment Confirmation [27], which is clearly an extension with high security requirements.

### IV.8.3 Usability

Usability is an important aspect that can affect the user experience and acceptance. It is an essential criterion that will certainly determine how successful FIDO authentication will become in the future. The usability for FIDO authentication is, however, not affected by our proposed protocol. The protocol requires a key exchange and subsequent encryption of FIDO extensions, which happens autonomously and is, therefore, unnoticed by the user.

## IV.9 Conclusion and Outlook

The FIDO protocols are a promising way to prevent security risks that arise with password authentication. However, we describe several MitM attacks that show that FIDO extensions are vulnerable to disclosure and manipulation. In order to mitigate such attacks, we propose a protocol that secures FIDO extensions by authenticated encryption. While our methodology includes some challenges such as the initial key exchange and displaying user information for authenticators without a secure display, we see a considerable security gain and aim for a standardized way to secure any kind of FIDO extension.

There are not many extensions used in practice yet. Nevertheless, the standardization process of the Secure Payment Confirmation indicates that we can expect more extensions to appear in the near future. At the time of writing, it is still under discussion whether arbitrary extensions should be allowed or not. We argue that it would be beneficial from a developer's perspective to be able to add extensions for different applications. This should, however, be done with security in mind. The protocol presented in this paper could provide a way to satisfy this requirement. In future work, we will test our approach in real-world scenarios. Furthermore, we are working on a lab environment that will facilitate practical research with FIDO authentication.

## References

[1] Akter, S. et al. “Man-in-the-Middle Attack on Contactless Payment over NFC Communications: Design, Implementation, Experiments and Detection”. In: *IEEE Transactions on Dependable and Secure Computing* (2020).

[2] Arshad, S., Kharraz, A., and Robertson, W. “Include me out: In-browser detection of malicious third-party content inclusions”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 441–459.

[3] Barbosa, M. et al. “Provable security analysis of FIDO2”. In: *Annual International Cryptology Conference*. Springer. 2021, pp. 125–156.

[4] Bianchi, A. et al. “What the app is that? deception and countermeasures in the android user interface”. In: *2015 IEEE Symposium on Security and Privacy*. IEEE. 2015, pp. 931–948.

[5] Blanchet, B. “Modeling and verifying security protocols with the applied pi calculus and ProVerif”. In: *Foundations and Trends® in Privacy and Security* vol. 1, no. 1-2 (2016), pp. 1–135.

[6] Bormann, C. and Hoffman, P. E. *Concise Binary Object Representation (CBOR)*. RFC 8949. Dec. 2020. DOI: 10.17487/RFC8949. URL: https://www.rfc-editor.org/info/rfc8949.

[7] Bui, T. et al. “Man-in-the-machine: exploiting ill-secured communication inside the computer”. In: *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 2018, pp. 1511–1525.

[8] Büttner, A. and Gruschka, N. “Enhancing FIDO Transaction Confirmation with Structured Data Formats”. In: *Norsk IKT-konferanse for forskning og utdanning*. 3. 2021.

[9] Büttner, A. et al. “Less is Often More: Header Whitelisting as Semantic Gap Mitigation in HTTP-Based Software Systems”. In: *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer. 2021, pp. 332–347.

[10] Dougan, T. and Curran, K. “Man in the browser attacks”. In: *International Journal of Ambient Computing and Intelligence (IJACI)* vol. 4, no. 1 (2012), pp. 29–39.

[11] Feng, H. et al. “A formal analysis of the FIDO UAF protocol”. In: *Proceedings of 28th Network And Distributed System Security Symposium (NDSS)*. 2021.

[12] Fernandes, E. et al. “Android ui deception revisited: Attacks and defenses”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 41–59.

[13] FIDO Alliance. *FIDO Alliance Metadata Service*. 2021. URL: https://fidoalliance.org/metadata/ (visited on 11/17/2021).

[14] FIDO Alliance. *FIDO Alliance Specifications Overview*. 2021. URL: https://fidoalliance.org/specifications/ (visited on 11/17/2021).

[15] FIDO Alliance. *FIDO Transaction Confirmation White Paper*. Tech. rep. Aug. 2020. URL: https://media.fidoalliance.org/wp-content/uploads/2020/08/FIDO-Alliance-Transaction-Confirmation-White-Paper-08-18-DM.pdf.

[16] FIDO Alliance. *History of FIDO Alliance*. 2021. URL: https://fidoalliance.org/overview/history/ (visited on 11/17/2021).

[17] Frymann, N. et al. "Asynchronous Remote Key Generation: An Analysis of Yubico's Proposal for W3C WebAuthn". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 939–954.

[18] Gil, O. "Web cache deception attack". In: *Black Hat USA* vol. 2017 (2017).

[19] Google. *FIDO2 API for Android*. 2020. URL: https://developers.google.com/identity/fido/android/native-apps (visited on 11/17/2021).

[20] Group, W. W. A. W. *Web Authentication (WebAuthn)*. 2020. URL: https://www.iana.org/assignments/webauthn/webauthn.xhtml (visited on 11/17/2021).

[21] Jakkal, V. *The passwordless future is here for your Microsoft account*. 2021. URL: https://www.microsoft.com/security/blog/2021/09/15/the-passwordless-future-is-here-for-your-microsoft-account/ (visited on 11/17/2021).

[22] Kumar, A. et al. *Web Authentication: An API for accessing Public Key Credentials - Level 2*. W3C Recommendation. Apr. 2021. URL: https://www.w3.org/TR/2021/REC-webauthn-2-20210408/.

[23] Kunke, J. et al. "Evaluation of Account Recovery Strategies with FIDO2-based Passwordless Authentication". In: *Open Identity Summit 2021*. Ed. by Roßnagel, H., Schunck, C. H., and Mödersheim, S. Bonn: Gesellschaft für Informatik e.V., 2021, pp. 59–70.

[24] Lahmadi, A. et al. "Mitm attack detection in ble networks using reconstruction and classification machine learning techniques". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2020, pp. 149–164.

[25] Landrock, P. and Pedersen, T. "WYSIWYS?—What you see is what you sign?" In: *Information Security Technical Report* vol. 3, no. 2 (1998), pp. 55–61.

[26] Linhart, C. et al. *HTTP Request Smuggling*. 2005. URL: https://www.cgisecurity.com/lib/HTTP-Request-Smuggling.pdf (visited on 04/17/2020).

[27] McGruer, S. and Solomakhin, R. *Secure Payment Confirmation*. W3C Working Draft. https://www.w3.org/TR/2021/WD-secure-payment-confirmation-20210831/. W3C, Aug. 2021.

[28] Owens, K. et al. "User Perceptions of the Usability and Security of Smartphones as FIDO2 Roaming Authenticators". In: *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX Association, Aug. 2021, pp. 57–76.

[29] Pfeffer, K. et al. "On the Usability of Authenticity Checks for Hardware Security Tokens". In: *30th {USENIX} Security Symposium ({USENIX} Security 21)*. 2021.

[30] Porter, J. *Safari to support password-less logins via Face ID and Touch ID later this year*. 2020. URL: https://www.theverge.com/2020/6/24/21301509/apple-safari-14-browser-face-touch-id-logins-webauthn-fido2 (visited on 11/17/2021).

[31] Raspberry Pi Ltd. *Raspberry Pi Documentation - Raspberry Pi Pico*. 2022. URL: https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html (visited on 05/17/2022).

[32] Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: https://rfc-editor.org/rfc/rfc8446.txt.

[33] Schaad, J. *CBOR Object Signing and Encryption (COSE)*. RFC 8152. July 2017. DOI: 10.17487/RFC8152. URL: https://rfc-editor.org/rfc/rfc8152.txt.

[34] Selander, G., Mattsson, J. P., and Palombini, F. *Ephemeral Diffie-Hellman Over COSE (EDHOC)*. Internet-Draft draft-ietf-lake-edhoc-12. Work in Progress. Internet Engineering Task Force, Oct. 2021. 80 pp. URL: https://datatracker.ietf.org/doc/html/draft-ietf-lake-edhoc-12.

[35] Sun, D.-Z., Mu, Y., and Susilo, W. "Man-in-the-middle attacks on Secure Simple Pairing in Bluetooth standard V5. 0 and its countermeasure". In: *Personal and Ubiquitous Computing* vol. 22, no. 1 (2018), pp. 55–67.

[36] Wendlandt, D., Andersen, D. G., and Perrig, A. "Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing." In: *USENIX Annual Technical Conference*. Vol. 8. 2008, pp. 321–334.

[37] Xu, P. et al. "SDD: A trusted display of FIDO2 transaction confirmation without trusted execution environment". In: *Future Generation Computer Systems* vol. 125 (2021), pp. 32–40.

[38] Zhang, Y. et al. "Secure display for FIDO transaction confirmation". In: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018, pp. 155–157.

[39] Zhang, Z. et al. "An empirical study of potentially malicious third-party libraries in Android apps". In: *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 2020, pp. 144–154.

Paper V

# Secure and Privacy-Preserving Authentication for Data Subject Rights Enforcement

**Malte Hansen, Andre Büttner**

**Revision 01**
Date: April 2024
Editor: Andre Büttner

This is an updated version of the paper presented and published at IFIP Summer School on Privacy and Identity Management 2023. It includes changes in the format and minor text modifications. These changes do not affect the scope or conclusions of the paper in any way.

Originally published in *Privacy and Identity Management. Sharing in a Digital World*; ISBN 978-3-031-57978-3; pages 175–191; Springer Nature Switzerland, Cham (2024). DOI: 10.1007/978-3-031-57978-3_12.

## Abstract

In light of the GDPR, data controllers (DC) need to allow data subjects (DS) to exercise certain data subject rights. A key requirement here is that DCs can reliably authenticate a DS. Due to a lack of clear technical specifications, this has been realized in different ways, such as by requesting copies of ID documents or by email address verification. However, previous research has shown that this is associated with various security and privacy risks and that identifying DSs can be a non-trivial task. In this paper, we review different authentication schemes and propose an architecture that enables DCs to authenticate DSs with the help of independent Identity Providers in a secure and privacy-preserving manner by utilizing attribute-based credentials and eIDs. Our work contributes to a more standardized and privacy-preserving way of authenticating DSs, which will benefit both DCs and DSs.

## V.1 Introduction

Since the GDPR entered into force in May 2018 [18], citizens of the EU have certain data subject rights (DSR). Therefore, the data controller (DC) of any service that collects personal data is required to implement measures for letting the rightful owner of the data, the data subject (DS), exercise their DSRs. For this, a DC must reliably identify and authenticate the data owner to ensure that only legitimate DSs can access their data. However, the GDPR does not specify concrete, technical, or non-technical measures to do this. Consequently, there are currently different implementations of DS authentication used in practice. On most popular online services such as Google or Meta, users can access their data and exercise their DSRs by authenticating to their user accounts. Yet, there are many other use cases where user data is collected, but no user accounts exist. It has, therefore, become common practice to authenticate DSs by requesting a copy of their ID document to match the information with a data set or by verifying, e.g., an email address. However, these methods can be problematic as they might put the DS at risk and are often not compliant with the GDPR. For instance, in 2022, a media company in the Netherlands was fined for unnecessarily requesting full ID documents, forcing them to change their procedure to email verification [8].

The establishment of a digital identity is another vital endeavor driven by the European Commission to provide EU citizens with a unified way to authenticate to digital services, even across countries, in order to push forward digitalization. For that purpose, the first European regulation on electronic identification, authentication, and trust services (eIDAS) was proposed in 2014, specifying requirements for implementing eIDs [16]. Due to a reluctant adoption of eIDs by different countries and new requirements by different stakeholders, the regulation was updated in 2021 in that eIDs shall be based on the EU Digital Identity Wallet [13], following the paradigm of the self-sovereign identity (SSI) [32]. With this, European citizens will be able to manage their own attributes, such as personal information otherwise found on an ID document or other official documents like their driver's license on a digital wallet on their phone.

We identified that eIDs can also be helpful in the context of DSR enforcement. The fact that citizens can present only specific attributes that are issued by a trusted authority and are necessary to verify a DS's identity makes it a useful way of authentication. However, this requires a flexible architecture that can be used by DCs with relatively low effort to be practically feasible. In this paper, we describe in detail how DS authentication based on eIDs can be implemented. Furthermore, we discuss different scenarios to showcase how our approach can be used in practice. The main contributions of this paper can thereby be summarised as follows:

- A concept for using eIDs and SSI in the context of DSR.
- An architecture for secure and privacy-preserving authentication of DSR requests with eIDs.

- An analysis of the architecture in consideration of the EU Data Strategy.

The content of this paper is structured as follows. Section V.2 gives detailed background information on DSRs and eIDs. In Section V.3, the related work is summarised. After that, Section V.4 provides an overview of the different authentication models and discusses eID as a solution for authentication in the context of DSRs. Section V.5 describes the architecture of our solution as well as example scenarios and an analysis. In Section V.6, the proposed solution is discussed. Section V.7 summarizes our results and gives an outlook on future work.

## V.2 Background

The GDPR [18] grants several DSRs to European citizens. As an example, with the Right to Erasure, a DS can request the deletion of all their information by a DC, while the Right of Access (RoA) allows a DS to request a copy of all their data from a DC. The disclosure of such a RoA result holds great risks. Hence, secure authentication is a core element of the enforcement process for many DSRs.

In fact, DSR enforcement was shown to be prone to authentication attacks. For RoA requests, several studies have gained unauthorized access via the utilization of social engineering attacks, leveraging the requirement for human actions in the authorization process [5, 6, 31]. In an eID scenario, these weaknesses extend to possible abuse of DSRs by an authoritative government [26]. To get rid of these weaknesses, the need for an alternative authentication process for DSRs has been identified, especially in scenarios where the DC does not have any established authentication method [23].

This demand for secure DSR authentication becomes even greater in the face of the European data strategy [12]. An essential part of the strategy is the establishment of a single European data market, fostering data-sharing inside the EU. As a result of this strategy, the Data Act [15] facilitates data-sharing across different sectors. Further, the Data Governance Act [14] introduces Data Intermediaries (DI), acting as a sort of data broker between different DCs and DSs while fostering DSRs. These developments will lead to increased data flows between more actors. Consequently, the existence of secure and reliable DSRs gains more importance. Another important development regarding authentication in the European data strategy is the relevance of eIDs. Building upon eIDAS, a framework for interoperability of eIDs inside Europe, the European Commission has proposed the Digital Identity Regulation [13], looking to establish a European Digital Identity available for any citizen, resident, or business inside the EU.

An authentication method that has recently gained a lot of traction is attribute-based credentials (ABC) [34], with ENISA identifying privacy-enhanced ABCs as an important technology in the European data strategy [10]. While this solution is most practical in use cases where a DS only has to prove one specific attribute, such as its age or citizenship, it has to be considered if and how ABCs can be utilized in an authentication process in a DSR enforcement scenario.

## V.3 Related Work

Previous research has analyzed how DSRs can be performed in practice. For instance, Boniface et al. [4] found that in several cases where Service Providers (SP) do not offer user accounts, DSs had to send in a copy of their ID document to verify their identity and sometimes even only to check on the eligibility of RoA. This, however, contradicts the principle of data minimization and might allow a malicious actor to misuse the document for impersonation. They also highlight the challenge for third-party trackers to re-identify a DS and propose a cookie with a pseudonym derived from a DS's email address. Another approach is the concept of Data Subject Rights as a Service (DSRaaS) [22]. DSRaaS defines Identity Providers (IdP) for DSRs, e.g., embedded in the responsibilities of a DI, that act as IdPs for DCs lacking the resources to authenticate a DS themselves properly.

Similar studies observed that beyond ID documents, email addresses and phone numbers are often used by DCs to authenticate DSs. It was discovered that there are several ways for a malicious actor to access the data of another DS, e.g., by forging an ID document using open-source intelligence techniques or by requesting the data from a non-corresponding email address [6, 7]. Furthermore, in the study by Urban et al. [39], many companies did not respond to data requests within the prescribed period of 30 days or at all.

Other work focused on identifying a DS within a data set. It might be challenging to create a reliable mapping between DSs and the data that actually falls within their DSR, as it highly depends on what identifiers are present and how reliably their ownership can be verified. Therefore, different levels of identification are defined and discussed [33].

Most of the prior research on eID addresses the implementation challenges and compliance in different countries like the UK [38] or Estonia and the Netherlands [27] based on the first eIDAS proposal [16]. Furthermore, some work looked at the security of eIDAS implementations. For example, a security analysis on different eID implementations was conducted that discovered that many implementations were vulnerable to XML-based attacks [9]. Another survey conducted in 2021 [37] studied what authentication protocols and methods were used by the eID implementations of different countries, revealing that several countries only provided a low level of assurance, thus providing low security properties. Beyond that, much research on eIDAS has been done in the context of universities [1, 3, 20].

The research above discloses security and privacy issues of current DSR authentication deployments. Yet, to the best of our knowledge, there is no proposal for a standardized way to overcome these problems. Our work contributes by investigating a new use case for eIDs that addresses the challenges of identifying and authenticating DSs, as described in the work mentioned above.

Table V.1: Summary of advantages and disadvantages of the different identity models. Note that this list is non-exhaustive.

| Model | Advantages | Disadvantages |
|---|---|---|
| Centralized | • Service can be independent of a third-party<br>• No tracking of users | • Higher implementation and user management effort by the service<br>• Users need to keep track of multiple credentials |
| Federated | • Lower implementation and user management effort by the service<br>• Less credentials for the users to keep track of | • Service depends on third-party IdP<br>• Users may have low control about which attributes are shared<br>• IdP can track user activity |
| Decentralized | • Service can be independent of a third-party<br>• Control over attributes remains with the users<br>• Lower implementation and user management effort by the service | • Users need to take care of backups |

In particular, we propose a solution that mitigates the risk of impersonation and helps to minimize the data in a standardized manner.

## V.4 Review of Authentication Systems

Authentication systems differ in who or what is controlling the identities of users. Throughout the years, authentication systems have been implemented in various ways. The main distinction lies in the different models, which are *centralized identity management* (CIM), *federated identity management* (FIM), and *decentralized identity management* (DIM) [2]. The latest[1] proposal for the European eID is based on the latter model that aims to provide European citizens with a secure and privacy-preserving authentication solution for online services.

### V.4.1 Identity Models

The different identity models come with several advantages and disadvantages. Table V.1 provides a brief overview of these. In a CIM system, users have accounts specifically for a service or within one enterprise, as shown in Figure V.1. This makes the respective digital identity independent from any other service or outside an enterprise. Compared to a federated system, this is more privacy-friendly as users can not easily be tracked across different services. However, it

[1] At the time of writing

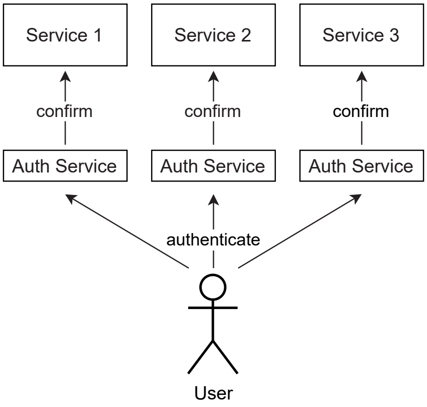


Figure V.1: Centralized identity model

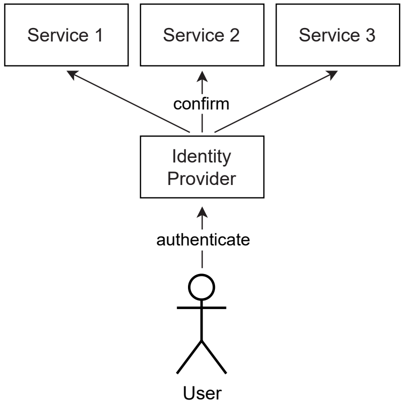


Figure V.2: Federated identity model

is less convenient for both the SP and the user. The SP needs to implement an authentication service and handle the user database. A user has to manage many different credentials for the distinct services, which often leads to the choice of less secure passwords [19]. Regarding DSRs, if a service with CIM provides sufficiently secure authentication methods such as multi-factor authentication, it can authenticate a DS for DSR requests without requiring additional data, thus not being affected by the risks described in Section V.3.

In FIM systems, the service, usually referred to as relying party (RP), outsources the identity management to an IdP. Users must authenticate against this IdP and grant access to certain account information to sign in to this RP. This is illustrated in Figure V.2. Depending on the IdP, a user might have limited control over how much information is shared between RP and IdP. Furthermore, the IdP can track the services accessed by a user, significantly impacting the user's privacy. However, with an FIM, the RP's implementation effort is relatively low since the IdP covers user management and authentication. The RP only needs to have a trusting relationship with the IdP and serve as a client. Commonly, this is implemented with widely-known, standardized protocols such as OAuth2 [24], OpenID Connect [35] or SAML [29]. Moreover, a user has fewer credentials to handle, as the same credentials are used for multiple services with the same IdP.

DIM systems aim to let users keep control of their identity attributes instead of IdPs. For that purpose, there are *attribute-based credentials* (ABCs) [34], which are a step towards a more user-centric approach. Using an architecture consisting of an *issuer*, *verifier*, *data holder*, it provides important privacy properties such as unlinkability and data minimization. The issuer is responsible for issuing valid attributes to the legitimate data holder who stores these credentials in their identity wallet. The data holder can then send attribute claims to a service that can verify these claims or use a third-party verification service to confirm the

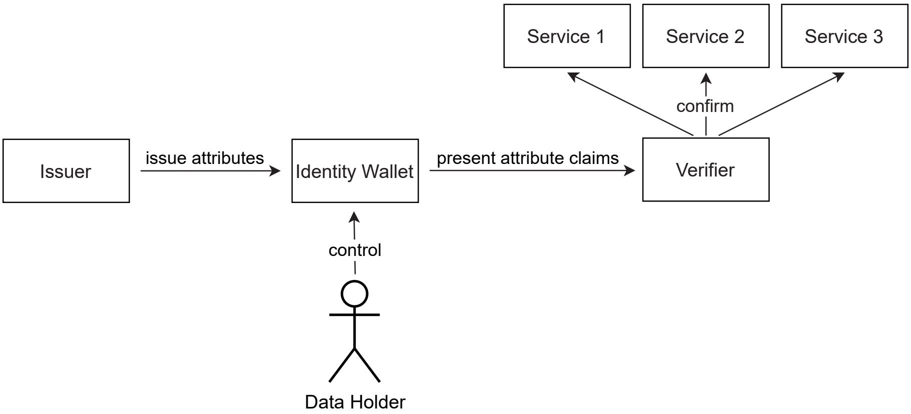


Figure V.3: Decentralized identity model based on ABCs

validity of the presented attributes. This is shown in Figure V.3. The paradigm behind this is referred to as *self-sovereign identity* (SSI) [32], which describes that users have complete control over their own attributes. It also often involves using decentralized architectures like distributed ledger technology (DLT) for public verifiability [28]. Since this is relatively new, users might have to get used to handling an identity wallet. Few studies on digital identity wallets have shown that users manage to use them to a certain degree but have problems creating and using a backup [25, 36]. However, one can assume that attributes can be re-issued in many cases. Also, it can be more convenient for users to have all their attributes and credentials in one place, which can be beneficial in the long run. In terms of security, it is crucial that the identity wallet is implemented with sufficient protection measures, e.g., using secure hardware and access control to prevent the credentials from being stolen. From an SP perspective, it can have similar benefits as for FIMs, as it is possible to combine it with FIM protocols like OpenID Connect to outsource the verification of credentials and thus does not require the implementation of any further authentication methods.

Importantly, SSI is considered in the latest eIDAS regulation [13] for the EU-wide implementation of eIDs. After the slow adoption of the first eIDAS regulation in 2014, based on a FIM [16], there has been an updated proposal (sometimes referred to as eIDAS 2.0). The new eID is based on SSI and builds on the EU Digital Identity (EUDI) Wallet.

### V.4.2 EU Digital Identity Wallet

According to the Digital Identity Regulation, “at least 80% of citizens should be able to use a digital ID solution to access key public services by 2030” [13]. Therefore, EU countries must ensure that an eID scheme that is compliant with this regulation is provided by 2030. A key requirement, already since the first eIDAS proposal, is to provide cross-border interoperability. In addition to the regulation, an architecture and reference framework has been created [17] to provide technical guidelines. The core element in the specified architecture is the EUDI Wallet instance, which is controlled by the user. Such a wallet can be either an application running locally on a mobile device or remotely as an online service. While the former case allows a user to protect their wallet physically, strong access control measures like MFA are required in the latter case to protect the wallet from being compromised. An identity wallet is meant to store different types of attributes, such as personal identifiable data (PID), which are attested and issued by trusted providers. These attributes can then be presented as verifiable claims to a relying party, which can verify the authenticity of these claims.

The reference framework also specifies which personal attributes are considered in the eIDAS framework. These attributes are basically those that can be found in ID documents. While the first name, last name, date of birth, and unique identifier are mandatory, other attributes are optional. These attributes are issued by specific PID providers, which must follow the issuing requirements for PID. Other types of attributes, e.g., driving licenses or digital payments, are issued by so-called (Non-)Qualified Electronic Attestation of Attributes ((Q)EAA) Providers [17].

### V.4.3 Using eID for Authenticating DSR Requests

The European Commission considers various use cases for their pilot implementation of the EUDI Wallet, such as access to electronic government services, opening a bank account, claiming prescriptions, and many more [11]. The use of eID in connection with DSRs has been suggested [23], but to the best of our knowledge, no concrete concept for this has yet been proposed.

As shown in Section V.3, prior research has revealed that the state-of-the-art for authenticating DSs for RoA requests has many issues concerning their security and privacy. We aim for an approach that neither suffers from weak authentication methods, such as passwords or email address verification nor violates data minimization, e.g., by requesting a copy of their complete ID document. Especially since the eIDAS regulation provides an EU-wide approach to authenticate citizens, this might be a particularly relevant authentication solution. The eIDs can help enforce DSRs in a more standardized, secure, and privacy-preserving manner.

We consider typical scenarios where services do not provide means of authentication, for instance, when using online retail as a guest user or advertising companies that collect information by cookies. In such cases, DCs need to put in place alternative methods to identify a DS in order to allow them to access their data. With the infrastructure of the EUDI Wallet, users can use the attributes stored in their identity wallet to present identity claims that overlap with the data held by the DC. This allows for matching between the data sets and their rightful owner. If a DC can verify the attributes presented by a DS, he can let them exercise their DSRs.

The required attribute claims will vary depending on the data sets. In many cases, a DC can request verifiable claims for the types of PID attributes that are included in the data set, e.g., first name, last name, and address. However, it is crucial that a DS can be identified reliably and that a sufficient number of attributes is verified. For example, verifying a DS's address is likely not sufficient. In cases where a data set is not sensitive, it might be acceptable to have lower requirements for the uniqueness of the requested attribute claims. For that purpose, the levels of identification discussed in [33] can be useful in deciding about such requirements. With eIDs, DCs have no reason to request complete ID documents from users. This mitigates the risk of their misuse and prevents the disclosure of irrelevant yet sensitive information. In addition, authentic ID information can only be obtained from a reliable and most likely governmental provider, making forgery significantly more difficult.

In cases where other means are used to authenticate a DS, such as email address or phone number verification, DCs should consider verifying other attributes that are included in the data set. Depending on the concrete use case, there might be verifiable attributes like the name, payment transactions, or other technical features. It is also conceivable that email providers themselves issue an attribute to verify the ownership of an email address, which might mitigate the risk of improper email validation. However, in that case, email providers still need to ensure secure authentication.

## V.5 Attribute-based eID Authentication for Data Subject Rights Enforcement

As stated in Table V.1, one of the advantages of a DIM is the control the user has over its credentials. This aligns with the DSRs' purpose of giving DSs control over their data. However, a pan-European solution also requires a uniform solution accessible to all DSs and DCs. Further, data minimization is a major concern during authentication. To meet these criteria, the proposed architecture combines an eID authentication framework with ABCs to provide a secure authentication scheme for DSR enforcement while reducing the amount of data disclosed during the process.

### V.5.1 Architecture

Inspired by Papadamou et al.'s privacy-preserving architecture for device-centric and attribute-based authentication [30], we introduce an architecture focused on assigning roles according to the European data strategy (see Figure V.4). It consists of a **(1) User Device**, an eID that can either be an RFID chip on a physical card or a standalone application on a device, such as the DSs smartphone. It is also possible for the application to be outsourced to a cloud. The User Device holds the credentials of the eID. A **(2) Service Provider** is a DC requiring authentication of a User Device, while a **(3) Identity Provider** is a trusted party, like a DI, receiving authentication requests for a User Device

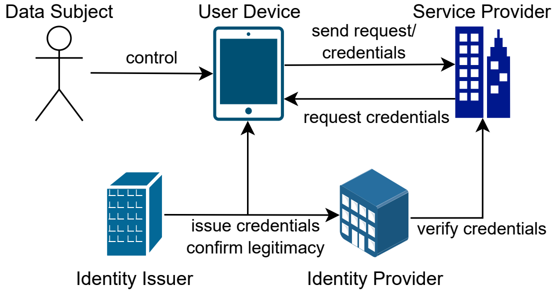


Figure V.4: Overview of Components in the Architecture

from the SP. For these requests, an ABC authentication scheme is used. Valid credentials depend on the relevant data sets for the DSR request, as well as the catalog of credentials included in the implementation of the eID. The IdP can either verify the credentials sent from the User Device to the SP or verify credentials for a specific DS itself. The **(4) Identity Issuer** is a neutral, third entity issuing the User Device's instance of the eID. For this purpose, they exchange information with the governmental entity that creates the national ID. Outsourcing the role of the Identity Issuer from the issuing government entity is recommended to combat the attack vector of a malicious government or entity. The Identity Issuer also maintains the instance of the User Device by performing updates or recoveries. Additionally, they confirm the expiration date of the eID to IdPs and the User Device. Additionally, they can possibly negotiate valid credentials. A DS might want to add or remove some attributes from the catalog of valid credentials or restrict certain attributes to specific use cases. While the Identity Issuer has an independent role in the architecture, they can also function as an IdP at the same time.

### V.5.2 Process Flow

Applying SSI to the architectural components leads to a process where the User Device holds all the credentials and the IdP only verifies them to the SP. However, an FIM solution may be preferred in specific circumstances, as will be discussed in Section V.5.3. In the FIM approach the process is mainly done between the SP and IdP. For both approaches, we assume that the Identity Issuer has issued a valid eID with a sufficient expiration date to the User Device and confirmed this to the IdP.

The SSI approach, see Figure V.5, starts with the DSR request by a DS for a certain data set held by a DC. So, the User Device sends a request with a payload to the SP. The payload includes information on which DSR shall be invoked and which data sets are subject to this request. This can be any single specific data set, all data sets with information about the DS the SP holds, or anything in between. The SP looks at which valid credentials can exist in their data sets, e.g., by taking the customer number for which the request was

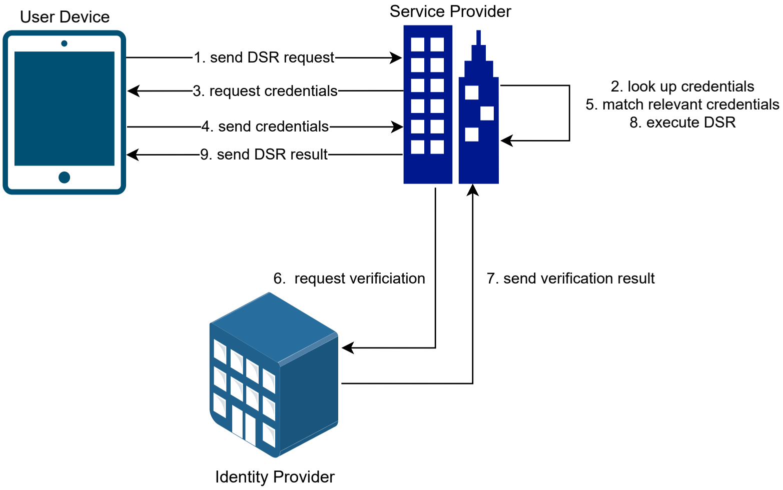


Figure V.5: Simplified flow of an SSI authentication process

issued and querying it over the catalog of valid credentials. The SP now sends a request for these credentials to the User Device. To prevent possible attack scenarios, this request is the same for any User Device contacting the SP. To see which credentials are relevant, the SP looks up which attributes exist in the corresponding data sets. After matching the stored credentials with the ones received in the response from the User Device, they are forwarded to the IdP, who verifies them. The following confirmation or decline of the response from the IdP does not contain any information about which credentials are correct or wrong. Finally, in case of successful verification, the DSR can be processed by the SP.

In the FIM approach, see Figure V.6, the request is initiated in the same way. The SP again looks for valid credentials in the relevant data sets. The SP then sends a request for authentication containing the required credentials to the IdP, which is doing the mapping in this scenario. The IdP then confirms the credentials. To prevent misuse of the authentication system by a malicious SP, the IdP notifies the User Device about the exchange with the SP. Again, the response to the SP does not contain any information about which attributes are correct or wrong. Alternatively, the DS could initiate the request in this approach over the IdP. The User Device would then send the parameters for the request to the IdP, which would forward them to the SP.

### V.5.3 Analysis

The main application of this model is for an attribute-based authentication of DSR requests. It allows the DC to determine a threshold of credentials to authenticate a DS while simultaneously verifying these credentials via a neutral, certified party. How to reach a sufficient threshold for an unambiguous

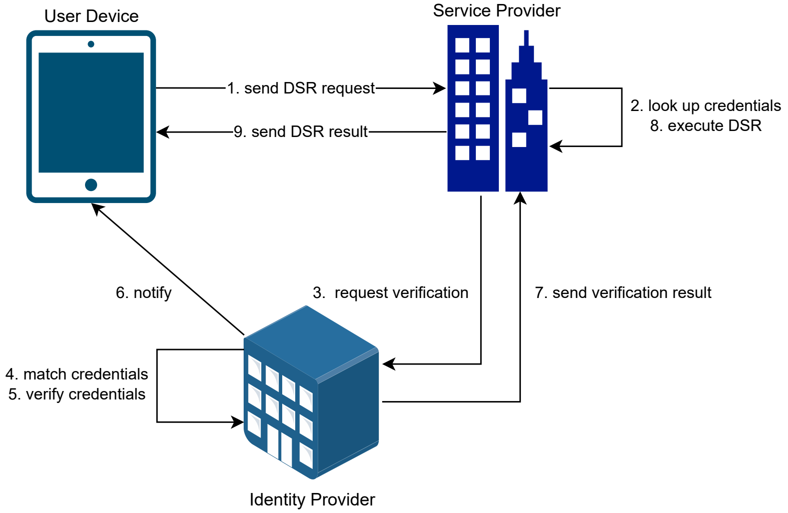


Figure V.6: Simplified flow of a FIM authentication process

authentication is highly complex and depends on the context and credentials at hand, therefore requiring further research. However, the architecture allows a variety of attributes to be included in this process. For example, the address and date of birth can be taken from the eID and mapped to an order number. This means that to authenticate a data set, it is only required to reveal a certain number of attributes instead of the entire catalog of credentials, as is often the case in a traditional authentication scenario. A precondition for this way of authentication is the existence of the credentials available in the eID solution in the data sets. Therefore, options for a meaningful expansion of these attributes, such as the (Q)EAA Providers in the EUDI Wallet, must be developed further. Using the architecture, a DS can also prove that a data set belongs to it based on specific credentials, even in cases where the data sets are not linked beforehand. This can also be utilized in scenarios where data sets belong to multiple DSs, e.g., in a shared streaming account. Since you do not need to reveal the user ID or name in the request, the DC can not learn which DS issued the request. This enables the possibility of an 'anonymous data subject right request'.

The process of searching the data sets for the credentials is greatly facilitated by a universal data model, as introduced in previous work [21]. By attaching metadata about the category of personal, such as a name, address, or date of birth, to the data sets, you can use these keywords to build your query.

A potential issue is a mismatched authentication due to incorrectly stored or guessed attributes or an impersonation attack based on credentials similar to the attacked DS. While the risk can not be completely mitigated, the IdP can see which eID issued the request. Hence, they can initiate adequate steps to address this problem as long as the eID itself is not compromised.

Besides their general characteristics, described in Section V.4.1, the two different approaches introduced come with their own advantages and disadvantages. In the SSI approach, the SP always requests all credentials that can theoretically exist in their data sets. This is done so an attacker cannot learn that a particular data set includes a specific attribute by instigating an intentionally wrong request. While the values can be hidden by, for example, matching only hashes instead of clear text, this nonetheless means that the SP could learn about the existence of a credential not included in the data set. For the FIM approach, a similar issue exists, as a malicious IdP can make the same attack based on the reduced catalog of only relevant credentials. However, with strong certification and auditing for the IdPs, this is easier to mitigate. The main difference between the two models comes in the distribution of competencies. The SSI flow gives most of the processing work to the SP, including the determination of the authentication threshold. Consequently, the IdP, as it acts as only a verifier, requires less implementation and acquires less information about the DS. Compared to a traditional approach, this still facilitates the implementation for the SP, as it can rely on the eID to verify the DS's claims. The FIM approach, on the other hand, puts a lot of responsibility on the IdP. This demands strong compliance control of the IdP. It is, however, advantageous in scenarios where the SP does not have the know-how or resources available to determine the authentication threshold reliably. For SPs that are hard to audit and assert compliance with, as may be the case with non-European DCs, or the DS does not trust the SP, this is also a preferred solution. Additionally, the FIM approach could also be realized for DSs without an eID. As the use of EUDI Wallets by citizens is not mandatory under the legislative proposal and their distribution has only started, this is an important factor to consider during this transition period.

Considering the direction the EU Data Strategy [12] is heading, the proposed architecture fills an important gap in ensuring secure and reliable authentication for DSRs. With the introduction of data spaces and the general increase in data sharing, it will become more prevalent that a DC will collect or receive personal data that is not linked to an existing authentication solution or other information, like e-mail addresses, that could be used to authenticate a DS easily. The DI, already at the center of this new data landscape, is a natural fit to fulfill the responsibilities of the IdP. Especially as one of the obligations of DIs is to facilitate DSR enforcement. For this purpose, the DI can serve the role of the Identity Issuer as well. Additionally, the involvement of the IdP and Issuer would mesh well with concepts like DSRaaS, where the DSRaaS Provider, or DI, already has a lot of competencies and responsibilities.

## V.6 Discussion and Open Issues

**Authentication threshold** As mentioned in Section V.5.3, it is important to determine a sufficient, at best generalized, threshold for a combination of credentials to identify a DS. However, this is a very challenging task as the threshold should be neither too high nor too low, and each data set might require a different set of attributes to achieve unambiguous identification. The threshold can also be impacted by the category of personal data in question. Sensitive data requires additional protection [18, Art. 9] and thus demands a stronger threshold. Further, the addition of new information to a context can always change this threshold. While privacy-enhancing technologies can be used to strengthen anonymization, they do not necessarily protect against re-identification through new information at a later point in time.

**Credential negotiation** A possibility to strengthen the authentication scheme is the addition of more credentials, which the EUDI Wallet addresses with the (Q)EAA Providers. This could possibly also be extended to technical identifiers. However, technical identifiers are not necessarily unique and will likely change over time, e.g., shared devices and IP addresses in public spaces. This makes them more unreliable. While a combination of these factors might reach a satisfactory threshold to identify a DS, regular exchange of information about these factors with the DS would be required to keep them up to date. Considering the privacy risks and contradiction of data minimization as well, it is highly questionable whether such an approach would improve the authentication process. Another possibility is the usage of derived attributes by the SP, such as the estimated address, gender, or age. This does hold potential, as an estimated age range or postal code can be used as a weak credential to reach a sufficient authentication threshold. However, it does include risks, as attributes might be derived incorrectly or based on outdated information. In the case of a rough estimate like an age range, it might also allow for too much leeway and lead to verifying a different person. Consequently, derived attributes should be labeled as those and only be used as a supplement to other credentials where only additional weak factors are required to reach a sufficient authentication threshold. Additionally, the accuracy of the underlying algorithm must be high enough so the usefulness of the additional credentials outweighs the risks it contains.

**Semantics** The semantics of data during the verification process is another issue that has to be considered during implementation. Lower- and upper-case writing, as well as localization of values with language-specific symbols, can lead to mistakes that would have to be specifically addressed by a process or manual intervention. Ideally, this should already be considered during the data creation by enforcing certain standardized, structured data formats. Already existing data sets will need to be normalized for that purpose.

**Non-European Data Subjects** Another question is how non-European residents can use the authentication architecture, as they do not possess a European eID. While it is possible to create a service to transform non-EU eIDs, it is not guaranteed that these residents even own an eID from their country. Therefore, a service to give an eID to non-EU is required anyway. This could be integrated into migration offices or the IdPs introduced here accordingly.

## V.7 Conclusion

This paper addresses the problem of identifying and authenticating data subjects in order to let them exercise their DSRs securely, reliably, and with respect to their privacy. To this end, various authentication schemes are first reviewed, and finally, an architecture based on eID and ABCs is proposed to provide a standardized authentication scheme. It includes User Devices belonging to the DSs, SPs, addressing the DCs, and IdPs and Issuers, which can find a meaningful representation in the European Data Strategy in the form of the DIs. The scheme introduces two different approaches, namely an SSI and an FIM approach, to meet different demands. This architecture can be used in particular by data controllers that do not offer their own authentication scheme, request copies of ID documents, or authenticate users by somewhat insecure methods like email address verification. Consequently, this architecture allows both DSs and DCs to rely on an eID solution for a secure authentication that only reveals the necessary attributes of the DS instead of its full identity.

In future work, an implementation of this model should be tested and improved. Furthermore, as the realization of the proposed architecture highly depends on the EU-wide adoption of eIDAS, further studies on the implementation of eIDAS should be conducted. While the architecture considers scenarios where eIDAS is not usable, this process must be more streamlined to not unnecessarily bloat the architecture. An open issue that must be solved before this scheme can be employed is the reliable construction of a sufficient authentication threshold that is both functional and secure. Finally, it has to be investigated further how this architecture can be used in conjunction with non-European DSs and DCs.

## References

[1] Alonso, Á. et al. “Enhancing university services by extending the eIDAS European specification with academic attributes”. In: *Sustainability* vol. 12, no. 3 (2020), p. 770.

[2] Avellaneda, O. et al. “Decentralized identity: Where did it come from and where is it going?” In: *IEEE Communications Standards Magazine* vol. 3, no. 4 (2019), pp. 10–13.

[3] Berbecaru, D., Lioy, A., and Cameroni, C. “Electronic identification for universities: Building cross-border services based on the eIDAS infrastructure”. In: *Information* vol. 10, no. 6 (2019), p. 210.

[4] Boniface, C. et al. “Security analysis of subject access request procedures: How to authenticate data subjects safely when they request for their data”. In: *Privacy Technologies and Policy: 7th Annual Privacy Forum, APF 2019, Rome, Italy, June 13–14, 2019, Proceedings 7*. Springer. 2019, pp. 182–209.

[5] Cagnazzo, M., Holz, T., and Pohlmann, N. “Gdpirated–stealing personal information on-and offline”. In: *European Symposium on Research in Computer Security*. Springer. 2019, pp. 367–386.

[6] Di Martino, M. et al. “Personal Information Leakage by Abusing the GDPR ’Right of Access’”. In: *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 2019.

[7] Di Martino, M. et al. “Revisiting identification issues in GDPR ‘Right Of Access’ policies: a technical and longitudinal analysis”. In: *Proceedings on Privacy Enhancing Technologies* vol. 2022, no. 2 (2022), pp. 95–113.

[8] EDBP. *Dutch SA fines DPG Media Magazines for unnecessarily requesting copies of identity documents | European Data Protection Board*. en. 2022. URL: https://edpb.europa.eu/news/national-news/2022/dutch-sa-fines-dpg-media-magazines-unnecessarily-requesting-copies-identity_en (visited on 02/21/2023).

[9] Engelbertz, N. et al. “Security Analysis of {eIDAS}–The {Cross-Country} Authentication Scheme in Europe”. In: *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. 2018.

[10] ENISA. *Engineering Personal Data Sharing*. en. URL: https://www.enisa.europa.eu/publications/engineering-personal-data-sharing (visited on 02/21/2023).

[11] European Commission. *EU Digital Identity Wallet Pilot implementation*. en. 2023. URL: https://digital-strategy.ec.europa.eu/en/policies/eudi-wallet-implementation (visited on 06/15/2023).

[12] European Commission. *European data strategy – Making the EU a role model for a society empowered by data*. https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy_en. 2022. (Visited on 04/14/2023).

[13] European Commission. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity*. SEC(2021) 228 final - SWD(2021) 124 final - SWD(2021) 125 final. June 3, 2021. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52021PC0281 (visited on 07/05/2023).

[14] European Commission. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on European data governance (Data Governance Act)*. COM/2020/767 final.

[15] European Commission. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on harmonised rules on fair access to and use of data (Data Act)*. SEC(2022) 81 final - SWD(2022) 34 final - SWD(2022) 35 final.

[16] European Commission. *Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.* July 23, 2014. URL: http://data.europa.eu/eli/reg/2014/910/oj (visited on 05/07/2023).

[17] European Commission. *The Common Union Toolbox for a Coordinated Approach Towards a European Digital Identity Framework.* en. Jan. 2023. URL: https://ec.europa.eu/newsroom/dae/redirection/document/93678 (visited on 06/22/2023).

[18] European Parliament and Council. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance).* May 4, 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679 (visited on 03/21/2023).

[19] Gaw, S. and Felten, E. W. “Password management strategies for online accounts”. In: *Proceedings of the second symposium on Usable privacy and security.* 2006, pp. 44–55.

[20] Gerakos, K. et al. “Electronic authentication for university transactions using eIDAS”. In: *E-Democracy–Privacy-Preserving, Secure, Intelligent E-Government Services: 7th International Conference, E-Democracy 2017, Athens, Greece, December 14-15, 2017, Proceedings 7.* Springer. 2017, pp. 187–195.

[21] Hansen, M., Gruschka, N., and Jensen, M. “A Universal Data Model for Data Sharing under the European Data Strategy”. In-press. 2023.

[22] Hansen, M., Gruschka, N., and Jensen, M. “Introducing the Concept of Data Subject Rights as a Service Under the GDPR”. In: *Privacy Symposium 2023.* Ed. by Schiffner, S., Ziegler, S., and Jensen, M. Cham: Springer International Publishing, 2023, pp. 17–31.

[23] Hansen, M. and Jensen, M. “A Generic Data Model for Implementing Right of Access Requests”. In: *Privacy Technologies and Policy: 10th Annual Privacy Forum, APF 2022, Warsaw, Poland, June 23–24, 2022, Proceedings.* Springer. 2022, pp. 3–22.

[24] Hardt, D. *The OAuth 2.0 Authorization Framework.* RFC 6749. Oct. 2012. DOI: 10.17487/RFC6749. URL: https://www.rfc-editor.org/info/rfc6749.

[25] Khayretdinova, A. et al. “Conducting a usability evaluation of decentralized identity management solutions”. In: *Selbstbestimmung, Privatheit und Datenschutz: Gestaltungsoptionen für einen europäischen Weg.* Springer Fachmedien Wiesbaden Wiesbaden, 2022, pp. 389–406.

[26] Lauradoux, C. "Can Authoritative Governments Abuse the Right to Access?" In: *Privacy Technologies and Policy: 10th Annual Privacy Forum, APF 2022, Warsaw, Poland, June 23–24, 2022, Proceedings*. Springer. 2022, pp. 23–33.

[27] Lips, S., Bharosa, N., and Draheim, D. "eIDAS implementation challenges: the case of Estonia and the Netherlands". In: *International conference on electronic governance and open society: challenges in Eurasia*. Springer. 2020, pp. 75–89.

[28] Mühle, A. et al. "A survey on essential components of a self-sovereign identity". In: *Computer Science Review* vol. 30 (2018), pp. 80–86.

[29] Organization for the Advancement of Structured Information Standards. *Security Assertion Markup Language (SAML) v2.0*. 2005.

[30] Papadamou, K. et al. "Killing the password and preserving privacy with device-centric and attribute-based authentication". In: *IEEE Transactions on Information Forensics and Security* vol. 15 (2019), pp. 2183–2193.

[31] Pavur, J. and Knerr, C. "Gdparrrrr: Using privacy laws to steal identities". In: *arXiv preprint arXiv:1912.00731* (2019).

[32] Preukschat, A. and Reed, D. *Self-sovereign identity*. Manning Publications, 2021.

[33] Purtova, N. "From knowing by name to targeting: the meaning of identification under the GDPR". In: *International Data Privacy Law* vol. 12, no. 3 (2022), pp. 163–183.

[34] Sabouri, A. and Rannenberg, K. "ABC4Trust: protecting privacy in identity management by bringing privacy-ABCs into real-life". In: *Privacy and Identity Management for the Future Internet in the Age of Globalisation: 9th IFIP WG 9.2, 9.5, 9.6/11.7, 11.4, 11.6/SIG 9.2. 2 International Summer School, Patras, Greece, September 7-12, 2014, Revised Selected Papers 9*. Springer. 2015, pp. 3–16.

[35] Sakimura, N., Bradley, J., and Jones, M. *Final: OpenID Connect Core 1.0 incorporating errata set 1*. en. 2014. URL: https://openid.net/specs/openid-connect-core-1_0.html (visited on 04/03/2023).

[36] Satybaldy, A. "Usability Evaluation of SSI Digital Wallets". In: *IFIP International Summer School on Privacy and Identity Management*. Springer, 2022, pp. 101–117.

[37] Sharif, A. et al. "The eIDAS Regulation: A Survey of Technological Trends for European Electronic Identity Schemes". In: *Applied Sciences* vol. 12, no. 24 (2022), p. 12679.

[38] Tsakalakis, N., O'hara, K., and Stalla-Bourdillon, S. "Identity Assurance in the UK: technical implementation and legal implications under the eIDAS Regulation". In: *Proceedings of the 8th ACM Conference on Web Science*. 2016, pp. 55–65.

[39] Urban, T. et al. “A study on subject data access in online advertising after the GDPR”. In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2019 International Workshops, DPM 2019 and CBT 2019, Luxembourg, September 26–27, 2019, Proceedings 14*. Springer. 2019, pp. 61–79.