Seminar report

Master’s Programme in Computer Science

Challenges with implementing multi-factor authentication

Riku Rauhala

April 9, 2024

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki, Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Study programme Master's Programme in Computer Science

Tekijä — Författare — Author

Riku Rauhala

Työn nimi — Arbetets titel — Title

Challenges with implementing multi-factor authentication

Ohjaajat — Handledare — Supervisors

Valtteri Niemi, Markku Kojo

Työn laji — Arbetets art — Level Seminar report	Aika — Datum — Month and year April 9, 2024	Sivumäärä — Sidoantal — Number of pages 15 pages, 2 appendix pages

Tiivistelmä — Referat — Abstract

Conventional password-based single-factor authentication methods frequently result in user accounts being compromised and valuable data being stolen. Passwords can be obtained by malicious actors due to vulnerable code, password reuse, social engineering tactics and malware.

Multi-factor authentication introduces an essential extra layer of security by requiring users to present two or more pieces of evidence called *factors* to verify their identity. With multi-factor authentication, unauthorised access can be effectively thwarted even if one of the factors is compromised.

Despite the numerous benefits that multi-factor authentication has to offer, not all software developers are enthusiastic about integrating it into their applications. Implementing multi-factor authentication comes at the cost of extra work and increased complexity.

This seminar report examines the possible challenges associated with implementing multi-factor authentication from a software developer's perspective. Practical and theoretical solutions are discussed to address these challenges.

In this paper, multi-factor authentication is mostly discussed in the context of web, mobile and desktop applications, but some other examples of implementation are also mentioned. Due to the limited scope of this paper, only some challenges can be inspected in detail. The paper's contribution is offering software developers an overview of the topic and a starting point for implementing multi-factor authentication.

ACM Computing Classification System (CCS)
Security and privacy → Security services → Authentication → Multi-factor authentication

Avainsanat — Nyckelord — Keywords

authentication, cyber security, multi-factor authentication, two-factor authentication

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Seminar: Advanced Topics in Networking and Security

Contents

1 Introduction

Authentication is a common challenge that software developers encounter in their line of work. Typically, an application requires a means to verify a user's identity before granting them access to a particular feature or piece of data. A conventional way to implement authentication is to request users to choose a password. While alternatives exist, passwords remain the most popular method (Zimmermann and Gerber, 2020).

However, authentication based on passwords only is often not secure enough. Users tend to choose weak, easily guessable passwords (Shen et al., 2016) or reuse the same password on multiple services (Zimmermann and Gerber, 2020). Passwords and their hashes are also frequently compromised and leaked to the Internet (AlSabah, Oligeri, and Riley, 2018). Even if a service is cryptographically secure and contains no vulnerabilities, user accounts protected by passwords can still become compromised if a user is targeted directly via social engineering (Gehl and Lawson, 2022), keylogger malware (Singh et al., 2021) or other malicious methods.

To address the shortcomings of password-based authentication, an additional layer of security can be introduced. *Multi-factor authentication* (or MFA for short) requires the user to verify their identity in more than one way (OWASP, 2024).

In this seminar report, the challenges of implementing multi-factor authentication are inspected from a software developer's perspective. *Implementation* refers to the process of adding support for MFA in the application under development.

The research process is guided by the following research questions.

> **Research question 1**: What challenges do software developers face when implementing multi-factor authentication?
>
> **Research question 2**: How can these challenges be overcome?

In chapter 2 a technical overview of the subject is provided and the details of multi-factor authentication are introduced. In chapter 3 the challenges of implementation are explored and solutions are presented. Chapter 3 provides answers to the research questions. Chapter 4 provides conclusions and summarises the findings of this seminar report.

2 Background

As mentioned in chapter 1, password-based authentication has several disadvantages. User accounts may become compromised due to vulnerable code or human error. Once the password has been obtained by an attacker with malicious intent, there is nothing left to prevent them from logging in with the victim's credentials, as authentication is based on the assumption that the password is only known by the individual who selected it. This issue is inherent in any form of *single-factor authentication* (Fenton, Grassi, and Garcia, 2017, p. 54) that relies on only one method for verifying a user's identity. This is where multi-factor authentication comes into play.

Instead of relying solely on a password (or a similar method such as a PIN number) for verifying a user's identity, multi-factor authentication is based on requiring users to provide more than one *factor* or type of evidence of identification.

As defined by OWASP (2024), there are five main types of factors:

1. **Something you know** - Passwords, PINs, security questions
2. **Something you have** - Security tokens, email accounts, phone numbers
3. **Something you are** - Facial features, fingerprints and other biometrics
4. **Something you do** - Typing patterns, mouse movements
5. **Somewhere you are** - Geolocation, IP addresses

The first three factor types are the most commonly used and the definition by the U.S. National Institute of Standards and Technology only covers the three of them (Fenton, Grassi, and Garcia, 2017, p. 49). The other two have niche use cases, e.g. making certain features only accessible from a company network in the case of factor 5.

Multi-factor authentication is commonly implemented as *two-factor authentication* (or 2FA for short) and the terms are often used interchangeably. Deciding to use more than two factors is possible but ultimately the choice depends on the context and on how much extra security is needed. The more factors are required, the more complicated the authentication process becomes.

The selection of factors depends on the specific application requirements. For example, in the case of a debit card the card itself is *something you have* and the PIN code is *something you know*. In online banking, the bank may require their customers to use the bank's own authenticator application to verify every login and transaction. In Finland, every major bank provides a separate application for this purpose (Danske Bank, 2024; Nordea, 2024; OP, 2024).

Multi-factor authentication adds a lot of complexity to an already challenging aspect of software development. There are many use cases for MFA and numerous ways to implement it. This can be overwhelming for developers. While there are challenges to overcome, there are also solutions.

3 Challenges

With additional security comes additional complexity. Deploying a conventional password-based authentication mechanism is relatively straightforward, but implementing multi-factor authentication comes with its own set of problems and challenges.

In this chapter, challenges related to the implementation of multi-factor authentication are explored.

3.1 User adoption

It can be difficult to get users to adopt multi-factor authentication in the first place (Golla et al., 2021). Some applications such as online banking platforms require MFA to be enabled for every user, but making it mandatory may not always be a practical solution. Some of the challenges discussed in this chapter could be thought of as sub-problems of user adoption: how do user experience, usability and security affect willingness of users to setup and utilise multi-factor authentication?

Most major websites and online services began offering multi-factor authentication in the early 2010s: Facebook and Google in 2011; Blizzard and Coinbase in 2012; Apple, GitHub, LinkedIn, Microsoft and Twitter in 2013 (Petsas et al., 2015, Fig. 1). However, even if multi-factor authentication is supported, not all users will take advantage of it. As modern web applications can have hundreds of millions, even billions of registered users, not everyone can be expected to have enough expertise in security and stay vigilant against attempts to compromise their accounts and steal their data.

While most notable websites have implemented some form of multi-factor authentication as of April 2024 (2factorauth, 2024), adoption rates among users remain low. Finding recent data on MFA adoption rates for each major website is difficult, as most companies do not make this information public. The share of users who have enabled multi-factor authentication appears to vary a lot depending on the service or industry. X Corp., formerly Twitter, Inc. (2022) reported that only 2.6% of active Twitter users had at least one 2FA method enabled in 2021. It was estimated by Petsas et al. (2015, Table 1) that

6.4% of Google accounts were protected by MFA in 2015. It should be noted that while X (formerly known as Twitter) is a microblogging platform, Google offers a large number of various services and collects more personal information about their users.

As the need for additional security has become more apparent, some websites outside of banking and finance have begun making multi-factor authentication mandatory for their users. In 2021, Google announced plans for automatically enabling MFA for 150 million users (Google, 2021). In 2023, GitHub started requiring all contributors to have at least one form of MFA enabled to keep using the site unrestricted (GitHub, 2024). This is a considerable change in policy from industry giants as MFA started seeing widespread, voluntary adoption only about a decade ago.

Users may not always clearly understand why enabling MFA is necessary. Providing users with an accurate model of how MFA works and why it is needed helps users to accept and enable multi-factor authentication (Golla et al., 2021, pp. 109–110). Reese et al. (2019) found that some users feel that their data is not worth the extra protection and the additional work that using multi-factor authentication requires. The importance of the data being protected affects the willingness to use MFA: users are more likely to enable multi-factor authentication for important accounts e.g. on financial or banking applications (Reese et al., 2019, p. 365), (Marky et al., 2022, p. 2) . Das et al. (2020, p. 5447) found that differences in computer expertise levels greatly affect user perception of multi-factor authentication: experienced users are more likely to understand the necessity of MFA while non-experts tend to view it as optional. Marky et al. (2022, p. 12) found similar results: experienced users favour MFA and understand the trade-off between usability and security.

Golla et al. (2021, pp. 109–110) found that using tailored messages to instil a sense of personal responsibility for security can be used to increase willingness for MFA adoption. They also found that a common marketing strategy of including the user's name in these messages would further increase the number of users enabling MFA.

Based on these findings, users can be educated to take better care of their own security. Being aware of the risks and common tactics used by online criminals could also help users become less likely to fall for phishing or social engineering.

Good communication about the security benefits of multi-factor authentication is critical to user adoption. Users are generally more accepting of more complicated and time-consuming authentication processes if the security benefits are made clear (Marky et al., 2022). It is therefore recommended to software developers to make sure that their users understand why they are required to setup and enable multi-factor authentication.

User adoption can also be boosted by paying attention to the other aspects of MFA implementation.

3.2 User experience

According to *Jakob's law of Internet user experience*, users expect things to work the way they are already familiar with (Yablonski, 2020). For this reason, most websites tend to resemble each other and consist of similar elements: headers, footers, navigation bars, clickable buttons and so on. Multi-factor authentication is not an exception: users expect familiarity and a consistent experience across different services. This presents a challenge for software developers as there are multiple ways to implement multi-factor authentication, each with a set of suitable use cases.

Issues with consistency begin with the term itself. In the context of this seminar report, the name *multi-factor* authentication is preferred to *two-factor* authentication. This is because two-factor authentication is always a form of multi-factor authentication but multi-factor authentication is not necessary implemented with just two factors. It is therefore more accurate to discuss the concept using the more broader term. Using another name could be more easily recognised by users and should be preferred in applications.

Based on a systematic study of 85 top-ranked websites, Lyastani, Backes, and Bugiel (2023, p. 10) found that the most commonly used names for referring to the same concept were two-factor authentication (49.4% of the websites) and *two-step verification* (28.2%). An additional 3.5% of the websites surveyed used the term *two-step authentication*. Only 4.7% used the term multi-factor authentication. The rest (14.1%) used another term.

To address the naming issue, developers should stick to a name that users recognise. Most of the top websites use the term two-factor authentication or two-step verification (Lyastani, Backes, and Bugiel, 2023, p. 10). While the most visited websites form only a tiny fraction of all websites on the Internet, they are visited by billions of users and design choices made by the large companies behind them lead the way for other developers. The term adopted and endorsed by the top websites has a huge impact on user perception and familiarity of the concept. Additionally, Google Trends (Google, 2024a; Google, 2024b) indicates a much higher search volume for 2FA compared to MFA in the past five years. If only two factors are used, developers should use the term two-factor authentication.

As inexperienced users occasionally have trouble finding relevant information, Das et al. (2020, p. 5447) recommend making the settings for MFA easy to discover. According to Lyastani, Backes, and Bugiel (2023, p. 10), an overwhelming majority (91.8%) of the top websites surveyed placed 2FA settings under either the *security* or the *account* category. For a consistent user experience, this is also a recommended practice unless there is a reason to decide otherwise.

Lyastani, Backes, and Bugiel (2023, p. 12) also found that overall there are inconsistencies with the user experience of websites, even among the industry giants such as Apple or Google. Their proposed solution is for official recommendations to be created and published by "influential industry associations and consortia". While such a standardised solution does not exist, software developers should attempt to identify and emulate current practices used on other websites. Overall, the significance of a consistent user experience should not be dismissed.

3.3 Usability

For a successful implementation of multi-factor authentication, usability must be taken into consideration. Typing a username and a password is relatively effortless due to the simplicity and popularity of the practice. Other more sophisticated methods could prove difficult for users to handle (Bonneau et al., 2012). The extra time and effort required is a common challenge as it makes authentication more laborious and time-consuming (Golla et al., 2021, p. 110). As multi-factor authentication adds a whole new level of complexity, it must be made as easy as possible to use. Users agree that usability is a crucial part in adopting MFA (Marky et al., 2022, p. 13).

Accessibility is part of usability and the needs of all users must be considered. People with disabilities should be taken into account, especially in the case of biometrics: fingerprint scanning may not be possible for people who have lost limbs and visually impaired people could be unable to use iris-based authentication (Ometov et al., 2018, p. 9). Alternative ways to authenticate should be offered as a single factor type will not always fit the needs of every user (Grassi et al., 2017, p. 51). The accessibility aspect can be improved by *personalisation* and *customisation* (Marky et al., 2022, p. 22).

Reese et al. (2019) conducted a usability study of several commonly used multi-factor authentication methods. The study found that if the setup process for MFA is properly implemented, users tend to find it easy to complete. Common usability issues identified in the study included failing to enter six-digit time-based one-time password (TOTP) codes before they expired and users not always having their MFA device readily available when needed. Das et al. (2020, p. 5448) also criticise the dependency on mobile devices. They suggest looking into alternative, more sophisticated ways that rely less on a specific physical device. Marky et al. (2022, p. 21) make a similar recommendation: for mobile approaches, the second factor should be independent of the device itself. This is also a security consideration: it should not be possible to compromise two factors at the same time.

For improved usability, Reese et al. (2019, p. 368) recommend instructing users during the setup phase so that they do not dismiss any important instructions. While they do not recommend any specific method for implementing MFA, push notifications and SMS messages received the highest scores from the survey participants.

Karim et al. (2024) conducted a comparative analysis of various MFA methods. Based on the results, Microsoft Authenticator and biometrics were found to be the most user-friendly options for factors. They recommend these options for situations where convenience should be prioritised over cost and security. While biometrics score high on usability, Karim et al. (2024, p. 208) also point out the additional risk of compromised biometric data. Once a malicious actor has obtained a person's fingerprints, they can be used for illegal purposes such as fraud or identity theft (Karim et al., 2024, p. 208).

Grassi et al. (2017, p. 50) recommend integrating usability into the software development process itself so that the application remains usable and secure. They also argue that conducting a usability evaluation on the selected factors is a critical step.

3.4 Security

Ultimately, the reason to use multi-factor authentication comes down to security and protecting valuable information. Despite its benefits, MFA is not a silver bullet and attention must be paid to properly securing each of the factors. Having multiple factors protecting user accounts makes the job more difficult for attackers, but not impossible.

While having two or more factors offers more protection, each of them can be compromised in various ways. Table 3.1 contains a summary of possible ways that some of the commonly used authentication methods can be compromised.

Factor	**Method**	**Threats**
Something you know	Password	Duplication, malware, phishing
	PIN	Eavesdropping, keylogging
Something you have	SMS (mobile device)	Malware, theft
	Security token	Cloning, theft
Something you are	Fingerprint scan	Forgery, replication
	Retinal scan	Replication
Something you do	Voice recognition	Voice imitation, replay attacks
	Gesture recognition	Observation, replay attacks
Somewhere you are	GPS	Spoofing
	IP	VPN, proxy servers

Table 3.1: Ways to compromise commonly used factors (Grassi et al., 2017, pp. 41–45)

Despite the potential vulnerabilities of each factor, software developers can attempt to mitigate the risks to their users. A critical implementation choice is to choose at least two different types of factors to be used in combination with each other so that they cannot be compromised the same way (Grassi et al., 2017; OWASP, 2024).

For instance, if the first factor is "something you know" and is implemented as a password, the second factor should be one of the other four options. A common solution is to choose the factor "something you have" and send a verification code to the user's phone number, email address, or a separate authenticator application installed on their device. This way, the person attempting to log in would have to not only know the correct password but also have access to the user's personal means of communication to receive the verification code.

To give a concrete example, let us discuss in more detail the security considerations of one of the common implementations of the ownership factor. When choosing which factors to support, developers may be tempted to go with sending verification codes as text messages (SMS). The advantages are obvious: most people already carry a mobile device in their pocket and are ready to receive messages without installing any additional software. With the rise of the smartphone in the past two decades, there is a high chance they are using the application on the same device to begin with. However, this could be a bad idea if security is a critical requirement for the application.

If a user's phone number is publicly available and a malicious actor is targeting them directly, the user may become susceptible to phishing attempts or social engineering (OWASP, 2024). An attacker may send a message masquerading as a legitimate MFA authentication request. Additionally, with the prevalence of mobile applications and browsing, the user may receive their MFA single-use code on the same device they are submitting their password on (OWASP, 2024). This is a problem because if the device is compromised, both factors (the password and the code delivered as a text message) can be obtained by an attacker at the same time.

With email, another popular method used as a *something you have* factor for MFA, the user is always in control of the factor unless an attacker manages to compromise the email account via the usual methods. With SMS however, there exists a way for impersonating the victim via a method called *SIM swapping*. If the victim's phone number is known by the attacker, they can contact the mobile operator and impersonate the victim with the goal of having the victim's phone number transferred to a SIM card they control (Jover, 2020). After taking over the victim's number, any MFA single-use code would then be sent to the attacker instead. Despite being a sophisticated and highly targeted attack method, Jover (2020, p. 16) argues that SIM swapping is the greatest threat against SMS as a factor for MFA. Karim et al. (2024, p. 208) also point out the issues with SMS and do not recommend using it for both security and usability reasons.

There is not much that for software developers to do about SIM swapping, other than to consider using some alternative factors such as email or a separate application. OWASP (2024) suggests considering to use a reliable third party service provider for MFA if the developer lacks resources for a proper implementation.

Some mitigation strategies have been deployed on a higher level and have been successful. In Mozambique, cellular operators allow banks to check for any possible occurrences of SIM swapping for a specific account (Jover, 2020, pp. 17–18).

Staying up to date with current recommendations is important for software developers. In the past, security question were a commonly used factor for account recovery. Registered users would provide answers to several personal questions. The questions were designed so that the answers would only be known to the user such as the city where their parents met or the name of their first pet. However, this practice has fallen out of use and it is no longer recommended to be used at all (OWASP, 2024) (Grassi et al., 2017). Security questions offer no additional benefits to passwords, yet the approach contains several flaws. The answers could potentially be known by people close to the user, be found on social media or obtained via phishing (OWASP, 2024). It is possible that in the future some of the currently recommended authentication methods are deemed too flawed or insecure to use as well. Other methods should be preferred, even in the case of account recovery.

When in doubt about the specific security considerations of each factor, software developers should listen to recommendations made by security experts. Two excellent documents containing useful recommendations and suggestions are the *Multifactor Authentication Cheat Sheet* by OWASP (2024) and *Digital Identity Guidelines: Authentication and Lifecycle Management* published by the National Institute of Standards and Technology (Grassi et al., 2017).

3.5 Account recovery

Software developers must consider the aspect of account recovery while implementing multi-factor authentication. What happens when a user loses access to one of their factors?

In the conventional approach to multi-factor authentication, all factors must be provided for a successful verification of identity. This is especially true in the case of two-factor authentication: accepting only one of the two would render the whole process meaningless. When it comes to traditional password-based single-factor authentication, most websites offer some sort of, usually fully automated mechanism for resetting a forgotten password. Multi-factor authentication is more complicated that, yet the need to recover lost factors also concerns MFA. How should developers handle lost or stolen mobile phones, changed phone numbers or other methods used as factors that have suddenly become unavailable?

As with other aspects of implementing multi-factor authentication, there is no general solution that could be applied to every single scenario. It is up to the software developer to choose the most suitable recovery methods for their specific use case and the needs of their application.

Offering account recovery options is mandatory as without it, users would get permanently locked out of their accounts in case any of their factors is suddenly inaccessible. The most simple option would be to resolve any problems via customer support, but as the number of users can potentially be in the hundreds of millions, some sort of automated solution would be preferable.

A common solution listed by OWASP (2024) is to provide the user with single-use recovery codes during the initial setup process. These codes can then be stored by the user in a secure location and used to regain access in case of any inaccessible factors. However, this approach trusts the user with an increased amount of responsibility: it would be up to the user to securely store the backup codes. An alternative way is to deliver the code to the user directly when needed, using their registered (and previously verified) email address.

In some situations a hardware backup solution could be considered, but they are generally used only in specific cases such as online banking or commercial settings due to higher cost (Golla et al., 2021, p. 110).

A novel solution proposed by Li et al. (2021) is a form of multi-factor authentication called *(t,n) threshold MFA* (or T-MFA for short). The idea is based on the observation that common implementations of MFA often require users to always provide every factor in order to be authenticated. If even one of the factors is missing or incorrect, authentication fails. This is also the case if the user no longer has access to one of their factors, e.g. their mobile phone.

With (t,n) threshold MFA, the user selects t factors out of n possible options, i.e. there are n factors to choose from but only t of them are required for successful authentication. In a practical example (Li et al., 2021, Fig. 1), a user could have a password (something you know, factor 1), a fingerprint (something you are, factor 3) and three separate devices (something you have, factor 2). The user could then authenticate using their password, their fingerprint and but only one of the three devices. In this case (where $t = 3$ and $n = 5$) the device they use would not matter, as any of them would be accepted by (t,n) threshold MFA. Even if the user loses access to their mobile phone or the device is compromised by a malicious actor, their account remains secure and they do not lose access.

Ometov et al. (2018, pp. 15–16) propose a similar solution for *vehicle-to-everything* (V2X) applications. A vehicle equipped with various sensors and devices, e.g. a weight sensor, a fingerprint scanner, face recognition hardware etc. (Ometov et al., 2018, Fig. 4), could offer multiple ways to grant access to the vehicle. This abundance could be leveraged as MFA factors and for example, only three out of four would be needed for successful verification. Given enough possible factors to choose from, a similar solution could be applied to software applications as well.

As losing access to a device is a common enough occurrence, reducing reliance on hardware should be taken as a preventive measure. Das et al. (2020, p. 5448) suggest a new recovery mechanism that would take into consideration a user's past behaviour and activities on the platform. However, novel solutions such as this lack precedent and could be difficult to implement.

4 Conclusion

Implementing multi-factor authentication is recommended for all applications that handle any confidential information. In conclusion, while MFA is a powerful additional security measure, various aspects must be considered for a successful implementation.

To summarise the findings and provide suggestions for software developers, the following steps are recommended.

1. **User adoption**: Unless multi-factor authentication is made mandatory, only a fraction of all users are likely to enable it. Adoption rates can be increased by improving usability and user experience as well as educating users on the necessity of MFA. Using tailored messages and personalised approaches are suggested.

2. **User experience**: Consistency and familiarity should not be overlooked. Users expect a consistent experience across different services and developers should adhere to commonly used terms. Unless there are more than two factors, the term *two-factor authentication* should be used. Making MFA settings easy to discover and providing clear instructions during the setup phase can enhance the user experience.

3. **Usability**: Usability should be taken into account. While typing a password is relatively easy, using more complex factors can pose challenges for users. Developers should ensure that MFA methods are easy to use and accessible, especially for people with disabilities. Usability affects willingness to adopt MFA.

4. **Security**: MFA adds an extra layer of security, but it is not completely foolproof. Each factor can be compromised in various ways, highlighting the importance of combining different types of factors to mitigate risks. The specific security measures should be considered for each factor selected.

5. **Account recovery**: Implementing an account recovery mechanism for MFA is mandatory. An automated solution such as providing single-use recovery codes is preferable. Security aspects of the chosen method must be considered.

Usage of AI tools

The following artificial intelligence services were utilised during the process of writing this seminar report. ChatGPT (OpenAI, 2024) was used to verify and explain LaTeX syntax. Keenious (Keenious, 2024) provided help with finding relevant source material.

Large language models were not used to generate any text for this seminar report.

Bibliography

2factorauth (2024). *2FA Directory*. URL: https://2fa.directory/int/ (visited on Apr. 4, 2024).

AlSabah, M., Oligeri, G., and Riley, R. (2018). "Your culture is in your password: An analysis of a demographically-diverse password dataset". In: *Computers & Security* 77, pp. 427–441. DOI: 10.1016/j.cose.2018.03.014.

Bonneau, J., Herley, C., Oorschot, P., and Stajano, F. (2012). "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes". In: *IEEE Symp. on Security and Privacy*, pp. 553–567. DOI: 10.1109/SP.2012.44.

Danske Bank (2024). *Danske ID -tunnistussovellus*. URL: https://danskebank.fi/sinulle/paivittaispalvelut/digitaaliset-pankkipalvelut/danske-id (visited on Mar. 9, 2024).

Das, S., Wang, B., Kim, A., and Camp, L. J. (2020). "MFA is A Necessary Chore!: Exploring User Mental Models of Multi-Factor Authentication Technologies". In: DOI: 10.24251/HICSS.2020.669.

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

Gehl, R. W. and Lawson, S. T. (2022). *Social Engineering: How Crowdmasters, Phreaks, Hackers, and Trolls Created a New Form of Manipulative Communication*. The MIT Press, p. 9. ISBN: 978-0-262-36892-6. DOI: 10.7551/mitpress/12984.001.0001.

GitHub (2024). *About mandatory two-factor authentication*. URL: https://docs.github.com/en/authentication/securing-your-account-with-two-factor-authentication-2fa/about-mandatory-two-factor-authentication (visited on Apr. 5, 2024).

Golla, M., Ho, G., Lohmus, M., Pulluri, M., and Redmiles, E. M. (2021). "Driving 2FA Adoption at Scale: Optimizing Two-Factor Authentication Notification Design Patterns". In: *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, pp. 109–126. ISBN: 978-1-939133-24-3.

Google (2021). *Making sign-in safer and more convenient*. URL: https://blog.google/technology/safety-security/making-sign-safer-and-more-convenient/ (visited on Apr. 5, 2024).

Google (2024a). *two factor authentication vs multi factor authentication.* URL: https://trends.google.com/trends/explore?date=today%205-y&q=two%20factor%20authentication,multi%20factor%20authentication&hl=en (visited on Mar. 20, 2024).

– (2024b). *two-factor authentication vs multi-factor authentication.* URL: https://trends.google.com/trends/explore?date=today%205-y&q=two-factor%20authentication,multi-factor%20authentication&hl=en (visited on Mar. 20, 2024).

Grassi, P. A., Fenton, J. L., Newton, E. M., Perlner, R. A., Regenscheid, A. R., Burr, W. E., Richer, J. P., Lefkovitz, N. B., Danker, J. M., Choong, Y.-Y., Greene, K. K., and Theofanos, M. F. (2017). *Digital identity guidelines: authentication and lifecycle management.* Tech. rep. National Institute of Standards and Technology. DOI: 10.6028/NIST.SP.800-63b.

Jover, R. P. (2020). "Security Analysis of SMS as a Second Factor of Authentication". In: *Queue* 18.4, pp. 37–60. ISSN: 1542-7730. DOI: 10.1145/3424302.3425909.

Karim, N. A., Kanaker, H., Abdulraheem, W. K., Ghaith, M. A., Alhroob, E., and Alali, A. M. F. (2024). "Choosing the right MFA method for online systems: A comparative analysis". In: *International journal of data and network science (Print)* 8.1, pp. 201–212. ISSN: 2561-8148.

Keenious (2024). *Keenious.* URL: https://keenious.com (visited on Mar. 5, 2024).

Li, W., Cheng, H., Wang, P., and Liang, K. (2021). "Practical Threshold Multi-Factor Authentication". In: *IEEE Transactions on Information Forensics and Security* 16, pp. 3573–3588. ISSN: 1556-6013, 1556-6021. DOI: 10.1109/TIFS.2021.3081263.

Lyastani, S. G., Backes, M., and Bugiel, S. (2023). "A Systematic Study of the Consistency of Two-Factor Authentication User Journeys on Top-Ranked Websites". In: *Proceedings 2023 Network and Distributed System Security Symposium.* DOI: 10.14722/ndss.2023.23362.

Marky, K., Ragozin, K., Chernyshov, G., Matviienko, A., Schmitz, M., Mühlhäuser, M., Eghtebas, C., and Kunze, K. (2022). ""Nah, it's just annoying!" A Deep Dive into User Perceptions of Two-Factor Authentication". In: *ACM transactions on computer-human interaction* 29.5, pp. 1–32. ISSN: 1073-0516.

Nordea (2024). *Activating the Nordea ID app.* URL: https://www.nordea.fi/en/personal/our-services/online-mobile-services/activating-nordea-ID-app.html (visited on Mar. 9, 2024).

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., and Koucheryavy, Y. (2018). “Multi-Factor Authentication: A Survey”. In: *Cryptography* 2.1. DOI: 10.3390/cryptography2010001.

OP (2024). *Mobile key.* URL: https://www.op.fi/private-customers/digital-services/mobile-key (visited on Mar. 9, 2024).

OpenAI (2024). *ChatGPT.* URL: https://chat.openai.com (visited on Mar. 5, 2024).

OWASP (2024). *Multifactor Authentication Cheat Sheet.* URL: https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html (visited on Feb. 27, 2024).

Petsas, T., Tsirantonakis, G., Athanasopoulos, E., and Ioannidis, S. (2015). “Two-factor authentication: is the world ready? Quantifying 2FA adoption”. In: *Proceedings of the Eighth European Workshop on System Security.* EuroSec ’15. Association for Computing Machinery. DOI: 10.1145/2751323.2751327.

Reese, K., Smith, T., Dutson, J., Armknecht, J., Cameron, J., and Seamons, K. (2019). “A usability study of five two-factor authentication methods”. In: *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security.* SOUPS’19. Santa Clara, CA, USA: USENIX Association, pp. 357–370. ISBN: 9781939133052.

Shen, C., Yu, T., Xu, H., Yang, G., and Guan, X. (2016). “User practice in password security: An empirical study of real-life passwords in the wild”. In: *Computers & Security* 61, pp. 130–141. DOI: 10.1016/j.cose.2016.05.007.

Singh, A., Choudhary, P., Singh, A. K., and Tyagi, D. K. (2021). “Keylogger Detection and Prevention”. In: *Journal of Physics: Conference Series* 2007.1, p. 1. DOI: 10.1088/1742-6596/2007/1/012005.

X Corp., formerly Twitter, Inc. (2022). *Account security.* URL: https://transparency.twitter.com/en/reports/account-security.html#2021-jul-dec (visited on Apr. 4, 2024).

Yablonski, J. (2020). *Laws of UX: Using Psychology to Design Better Products & Services.* O’Reilly Media, Inc. ISBN: 978-1-4920-5531-0.

Zimmermann, V. and Gerber, N. (2020). “The password is dead, long live the password – A laboratory study on user perceptions of authentication schemes”. In: *International Journal of Human-Computer Studies* 133, pp. 26–44. DOI: 10.1016/j.ijhcs.2019.08.006.