Multi-Factor Authentication Application Assessment: Risk Assessment of Expert-Recommended MFA Mobile Applications

Kevin Jensen, Faiza Tazi, Sanchari Das
University of Denver
{kevin.jensen42, faiza.tazi, sanchari.das}@du.edu

Abstract

The increased use of multi-factor authentication (MFA) has prompted the development of many competing MFA applications for secure authentication. Nevertheless, there is little research about the security vulnerabilities of these MFA mobile applications. To aid this, we conducted a thematic analysis on recent MFA-focused articles published in the year 2020 and performed security evaluation of 10 expert-recommended MFA mobile applications using RiskInDroid and Mobile Security Framework (MobSF). We found several code-based, permission-based, and cryptographic-based security violations of the applications which have severe vulnerability vectors. We conclude by providing actionable recommendations to fix any identified vulnerabilities and suggest stringent requirements for security-based applications to protect users from existing vulnerabilities.

1 Introduction

Robust authentication is a requisite security component for digital tools and technologies. According to a report by Microsoft, over 99.9% of breached systems did not use multi-factor authentication (MFA) to protect their accounts [1]. With news of several security breaches due to MFA adoption failure and recommendations by experts, we see a rise in the user adoption of MFA solutions to secure their accounts. Albeit, there is another school of thought, which discusses the problems in MFA adoption, due to its lack of focus on the usability component [8, 10–12, 25]. However, at the end of 2019, DUO Labs from CISCO found that more than 50% of interviewed users reported using some adoption of 2FA or MFA, compared to only 28% in 2017 [2]. Of the 50% who reported MFA use, over one-third of those users implemented a mobile MFA tool for authentication.

In the case of mobile MFA applications, users must login using their traditional credentials. Thereafter, the application prompts the user to input a code from the mobile MFA application [4, 6, 13, 20]. This is a trivial example of how MFA applications work, however, it is critical to understand the security of these applications which promise to deliver robust security controls for users [5, 7, 9]. To understand further, our study includes two phases: a thematic evaluation of recently published literature and an analysis of 10 MFA mobile applications. Our literature analysis included MFA-related articles published in the year 2020. Our analysis revealed that most of the analyzed MFA literature can be divided into four categories: 1) implementation; 2) emerging technology; 3) case studies; 4) MFA attack vectors and vulnerabilities. For our application analysis, we chose a sample of 10 popular and professionally-recommended mobile MFA applications to determine what code-based, cryptographic-based, and permission-based risks could be identified. We used two open source tools - RiskInDroid[3] and Mobile Security Framework (MobSF)[4] - to analyze and assess the risks of each MFA application in the study group.

Our paper provides a detailed insight on the current literature about MFA and MFA-focused mobile applications. Along with this, we also provide a detailed code-based, permission-based, and cryptographic-based risk analysis of these mobile MFA applications. Our research adds a new dimension to the literature while focusing towards understanding the risks and vulnerabilities of current MFA solutions.

[1] https://www.zdnet.com/article/microsoft-99-9%2Dof%2Dcompromised%2Daccounts%2Ddid%2Dnot%2Duse%2Dmulti%2Dfactor%2Dauthentication/

[2] https://duo.com/blog/the-2019-state-of-the-auth-report%2Dhas-2fa-hit-mainstream-yet

[3] https://github.com/ClaudiuGeorgiu/RiskInDroid

[4] https://github.com/MobSF/Mobile-Security-Framework-MobSF

Who Are You?! Adventures in Authentication (WAY) 2021.
August 7, 2021, Virtual Conference.

2 Methods: Thematic Evaluation Through Prior Literature

We selected articles from a variety of literature sources, published in the year 2020 using a set of search criteria. We began our data collection by retrieving a literature sample from three major databases: ACM, Google Scholar, and Science Direct. We then performed a quality assessment of the papers to ensure they met our inclusion criteria. Consequently, we conducted a thematic analysis of the remaining articles and categorized them.

Papers were selected based on the following criteria: 1) The paper was published in English; 2) The paper was published in a peer reviewed/academic journal/conference; 3) The paper focused on multi-factor authentication technology; 4) We also restricted our literature analysis to the year 2020 to ensure the most current topics were included and as a reference for what researchers consider important or relevant in the MFA field. Looking at the literature in this way allowed us to see if current research addresses any of the vulnerabilities found in our analysis.

We also excluded papers if: 1) The paper was not a scholarly article, such as book reviews, posters, and extended abstracts; 2) The text was not fully available by December 31, 2020; 3) The paper was only tangentially related to MFA and not the primary focus of the paper. We applied the first two search filters during the database collection. The third filter was applied during the data screening process. This process served as a form of quality assurance for our data collection and screening methods.

Our initial database collection yielded over 20,000 articles published in the three digital article databases. For the keyword-based search, we used the keywords "multi-factor authentication", "authentication", "MFA", and "multifactor authentication". However, our abstract and full-text screening revealed that most of the works discussed about authentication in general and mentioned MFA as a potential solution or a recommendation to create more robust security protocols. After the initial screening we then eliminated duplicate papers and began our data filtering process. The screening process refined our data set to the final 26 articles included in the literature analysis. We then categorized these 26 papers into four categories: 1) implementation; 2) emerging technology; 3) case studies; 4) MFA attack vectors and vulnerabilities.

3 Results: Thematic Evaluation Through Prior Literature

Through our literature analysis, we found that several researchers discuss the benefits of applying MFA for multiple emerging technologies. Out of these, drone technology, cloud computing, multi-server platforms, biometric-based tools, and the internet of things (Iot) are some of the most popular topics discussed in the current literature. In fact, 15 papers were related to emerging technologies [1, 2, 14–17, 19, 21–24, 26–28, 36]. For example, six articles explicitly discuss the importance and necessity of MFA in cloud technology, a relatively recent phenomena in the area of computer technologies. Prabha and Saraswathi address the necessity for new cryptographical methods for multi-factor authentication in cloud computing to preserve security and privacy [24]. Duncan recommends a multi-cloud security approach to secure cloud assets, which includes shoring up multi-factor authentication systems [14]. Kelf analyzes the risks, including authentication, of cloud migration, as more than half of UK businesses are expected to migrate to the cloud by the end of 2020 [19].

Out of the 26 papers analysed, six papers were related to MFA implementations on what are the suggested schemas for MFA application [3, 17, 18, 30, 31, 35]. In fact, biometrics is a growing solution of choice for MFA. Most hospital and healthcare systems have traditionally used personal identifying information to authenticate patients [31]. Additionally, Trung et al. propose the use of electroencephalogram data (records of brain activity – essentially a cerebral signature) in conjunction with a watermarking scheme to protect against spoofing [31]. Kele et al. posit the implementation of the Republic of Turkey Identity Cards (TCKK), which strengthen authentication with visual and electronic security components [18]. TCKK is an example of another biometric implementation to augment traditional multi-factor authentication – a security token in this case. Janik et al. also use biometrics to strengthen the popular pattern lock mechanism on mobile phones via behavioral biometrics [17]. Their biometric solution matches a user's behavior – how they swipe and interact with a device – with a login attempt.

On the other hand, only four papers were related to MFA attack vectors and vulnerabilities [25, 33, 34] where they discussed about MFA vulnerabilities and mitigation measures. For example, Wang et al. analysed five two-factor authentication schemes for multi-server environments, and uncover flaws in all applications, leading to critical attack vectors [32]. Yaacoub et al. evaluated the MFA solutions adopted in the banking sector while focusing on 30 banks across the world. Their evaluation was focused in terms of compliance with the law and best security practices, robustness against attacks, and complexity of MFA adoption and application [29].

4 Methods: Security Analysis

For the next phase of the study, we analyzed 10 expert-recommended MFA-based mobile applications. We selected the applications based on: 1) Recommended applications from technology-focused websites and online experts; 2) Most downloaded applications from the Android Play Store. Authy, Duo, Google, Microsoft, and SecureAuth applications were selected for their high download numbers on the An-

droid Play Store. Idaptive, Okta, PingID, Silverfort, and Symantec were selected because they were recommended by professionals on Google searches for the top MFA mobile applications All of the applications analyzed in this paper are recommended by websites that purportedly act as reliable repositories of information pertaining to cybersecurity and security tools [5]. Finally, we classified the applications in two categories: business and personal use. These categorizations were made by visiting the application's website and reading through it to determine what type of consumer the application markets itself to. This was done to determine if there were any immediate and noticeable differences between business and personal focused mobile MFA applications according to the scores derived from the static analysis.

RiskInDroid and MobSF tools were utilized to analyze and assess the various MFA applications included in this study. In tandem, both tools provide a holistic analysis of an application's permission-based risks, code-based risks, cryptographic risks, and more. Both tools are open source and publicly accessible through Git. RiskInDroid was chosen specifically for its analysis of permission-based risks. It reverse engineers a given application to assess all permissions used, requested or not. RiskInDroid provides a summary of the permissions requested by each application. A sample of the types of permissions requested has been detailed in table 1. MobSF is an automated analysis and security assessment tool for mobile applications that determines the code-based risks of a given application. MobSF aggregates each application's vulnerabilities and provides an average score using the Common Vulnerability Scoring System (CVSS) from the National Institute of Standards and Technology's (NIST) National Vulnerability Database (NVD). In addition, MobSF's reports provide a security score from zero to 100, which is the tool's cumulative assessment of the application. MobSF compiles its security score from an analysis of the application's: Signer Certificate, Permissions, Binary, Manifest, and Code. Both tools provide static analysis.

5 Results: Security Analysis

5.1 Permission-based Risk Assessment: RiskInDroid

The mean score from the analysis of the 10 MFA applications was 28.15, which is comparable to a low-to-moderate permission-based risk score according to RiskInDroid's metrics. Overall, this is a satisfactory score but could use improvement. The lowest score (and thus the highest rated) was the Microsoft Authenticator with a score of 10.10. The highest score (and thus the lowest rated) was the Idaptive Authenticator application with a score of 67.23, which is a moderate-to-high permission-based risk score. Additionally, the range of permission-based risk scores from these 10 applications is a staggering 57.14. On its own, a score of 57.14 would be considered a moderate-to-high permission-based risk. If we discard the two highest scores, the mean reduces to a 20.91. This indicates that, accounting for eight of the 10 applications, MFA applications are relatively safe permission-based applications, according to RiskInDroid.

A dichotomy also exists between business-oriented and non-business applications. For example, the highest scored application, Idaptive, is marketed exclusively for businesses. Aviation giant, Bombardier is a client of Idaptive, according to the latter's website [6]. Additionally, the lowest scored (thus highest rated) application was the Microsoft Authenticator app, which is not exclusive to business use. Furthermore, the mean score for the five business applications is 31.89, which is slightly higher than the mean of the 10 applications in aggregate. The business application mean is also significantly higher compared to the mean score (24.42) of the five personal or hybrid applications. Although this is only a preliminary analysis, it does suggest that, on average, business multi-factor authentication applications may pose a greater permission risk than non-business applications.

To detail more about the permission-based risks associated with these applications, commonly declared permissions included location, camera, and read-state permissions. The location permissions included fine (GPS) and coarse (network-based) definitions. Four of the 10 applications contain permission reference to a device's GPS location or its coarse location. One application also views the device's background location. Seven of the 10 applications require or access camera permissions in the static analysis. This was a surprisingly high amount of applications. Camera permissions could indicate the use of biometrics to authenticate users, but most authenticator applications utilize a persistent connection to avoid re-authenticating users every time they access the app. Meanwhile, only two applications write permissions to access the phone read state. The read state is helpful for applications tracking usage of their application, but it can be a source of concern in conjunction with other risky behaviors; such as apps that log sensitive data or permit external access by other applications. Finally, eight of the 10 applications inadvertently provide external access.

5.2 Code Feature-based and Cryptographic-based Risk Assessment: MobSF

MobSF's analysis of the MFA application sample produced some fascinating results. The mean score of the five applications categorized as personal, was 54. Out of them, Authy and Duo Mobile were the two applications that scored a five. The other three applications categorized as personal, Symantec

[5] https://www.expertinsights.com/insights/the-top-multi-factor-authentication-mfa-solutions-for-business/

[6] https://tracxn.com/d/companies/idaptive.com

Table 1: Permissions Requested by the 10 MFA Mobile Applications

Application	Authy	Duo	Google	Idaptive	Microsoft	Okta	PingID	SecureAuth	Silverfort	Symantec
Precise Location				X	X		X			
Approximate Location				X						
Camera Access		X	X	X	X		X	X		X
Other Applications	X	X			X		X	X	X	X
Read Phone State							X			X

VIP Access, Google Authenticator, and Microsoft Authenticator, scored 90, 70, and 100 respectively. The mean score of the five applications categorized as business, was a nearly identical 55. Out of these, Idaptive and PingID scored 10 in MobSF. Okta, SecureAuth, and Silverfort respectively scored 80, 90, and 85 respectively. Details and distribution of these MobSF evaluation scores can be seen in table 2.

We also report on the average common vulnerability scoring system (CVSS) score of the detected vulnerabilities in each application. The Microsoft Authenticator, which had a security score of 100, had an average CVSS score of zero. The remaining nine applications had an average CVSS score of 6.88. This indicates an average CVSS score of medium risk; although it is close to reaching the threshold (7.0) of a high risk vulnerability. Most interestingly, there was no correlation between an application's MobSF security score and its average CVSS score.

Interestingly, most applications shared very similar code-based risks. Eight of the 10 applications logged sensitive information, which is considered high-risk behavior, according to CVSS and scored 7.5 on their 10-point scale. Furthermore, six of the 10 applications used an insecure implementation of secure sockets layer (SSL) – improper certificate validation – vulnerable to man-in-the-middle attacks (MITM). Half of the sample utilized weak hashing algorithms, such as MD5 and SHA1, which are all known to have collision issues. NIST deprecated the use of SHA1 in 2012, stating it should no longer be used for digital signature generation [7].

Only two applications were found to read the device's state, which is a manifest permission. Reading phone state allows an app to identify a cell phone's number, network information, status of calls, and other details. Most applications use this permission to help with copy-protection and to track the number of users for an application usage, but it can be used for malicious purposes, if exploited. The malicious use of the read state permission can be a privacy concern for users depending on the information gathered and how it is used. This analysis did not focus on how the applications used and collected information from read state, but it is a potential area of concern regarding user privacy.

[7] https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions

6 Discussion

6.1 Code-based Risks

Unsurprisingly, several applications – such as the Google Authenticator and Microsoft Authenticator – were highly rated and feature few code-based risks. Nevertheless, RiskInDroid identified camera permission requests and access in seven of the 10 MFA applications – including the Google and Microsoft apps. MobSF identified eight MFA applications were accessible by external applications on the device where the MFA mobile application was installed. Code-based risks present a formidable challenge to MFA applications because they derive from the weakest link in cybersecurity: humans. MFA developers can perform constant penetration testing to detect any potential vulnerabilities in their applications. In addition, organization such as NIST can provide requirements, as opposed to recommendations, to encourage the development of more secure code. While it would be difficult to enforce secure software engineering to any extent - let alone punish offenders - initial steps could be taken for a type of universal secure software engineering best practices for software engineers. This removes the burden on developers as they would have a repository of information, libraries, and resources to develop more secure code.

6.2 Permission-based Risks

Permission-based risks are also important because of recent fissures in trust between users and tech companies. Interestingly, there were notable discrepancies between RiskInDroid and MobSF regarding some applications' permission-based risks. For example, PingID scored well in RiskInDroid with a 16.87 score - lower is better for RiskInDroid. MobSF, however, scored PingID 10/100 in its metrics - lower is worse for MobSF. Although MobSF also considers code-based and cryptographic-based risks, the disparity is still striking. This disparity demonstrates the need for a holistic analysis of multi-factor applications. Even if an application, like PingID, can score well in one metric, it may be found lacking in other areas. In addition, Authy and Duo both scored relatively well in RiskInDroid, but had the shared lowest score of all ten applications according to MobSF.This is vital to understand, as some users may prioritize certain issues above others.

Table 2: MobSF and Permission Risk Scores for Each MFA Mobile Applications

Application	Authy	Duo	Google	Idaptive	Microsoft	Okta	PingID	SecureAuth	Silverfort	Symantec
MobSF										
Security Score	5	5	70	10	100	80	10	90	85	90
Avg CVSS Score	7.5	6.6	6.8	6.2	0	7.0	6.8	7.5	7.5	7.5
Permission Risk										
Score	18.63	22.41	23.95	67.23	10.01	29.91	16.87	22.03	23.40	47.02
Declared	9	10	12	95	29	25	17	8	6	12
Exploited	4	2	3	16	10	8	8	2	3	3
Useless	5	8	9	79	19	17	9	6	3	9
Ghost	11	7	7	5	9	5	9	6	12	3

6.3 Cryptographic-based Risks

The code-based and cryptographic-based risks shared among the sample apps are another point of concern. For example, applications using the same insecure implementation of SSL mean these applications are vulnerable to MITM attacks. In addition, other applications implement hashing algorithms susceptible and vulnerable to collision. As a result, hackers may be able to replicate signatures or credentials. MFA applications, and other security focused implementations, should use the strongest available hashing algorithms, such as Argon2, in lieu of obsolescent algorithms like SHA1 and MD5. This requires NIST to update its own recommendations, which still include SHA256 as an acceptable hashing algorithm, despite its known weaknesses and vulnerabilities.

Two applications in particular, Idaptive and PingID, demonstrated significant cryptographical vulnerabilities. Idaptive was the only application analyzed that sent clear text traffic over the network with no cryptographic protocol. This is incredibly vulnerable to sniffing, MITM attacks, and other attacks. Idaptive also read and wrote to external storage devices, which compounds the issue of sending clear text traffic. Furthermore, Idaptive uses SHA1 as its hashing algorithm, which NIST formally deprecated in 2012 and disallowed as a digital signature. Okta and SecureAuth also employ SHA1. PingID employs the MD5 hashing algorithm. MD5 is a very vulnerable hash that researchers from Carnegie Mellon considers "broken and unsuitable for further use"[8]. Dougherty went even further and explicitly recommended that "software developers, Certification Authorities, website owners, and users should avoid using the MD5 algorithm in any capacity." These findings were published in 2008, and MD5 has been culpable for many notable attacks in the years since - including the well-known Skywiper malware in 2012.

7 Future Work and Limitation

We conducted a detailed two-phase study, to first determine the vulnerabilities of MFA analysed by previous researchers. Through the second-phase, we explore the current security vulnerabilities of existing expert-recommended MFA-focused applications. It is critical however, to acknowledge that we plan to address the limitation of this research by adding more research articles focused primarily on MFA applications and industry practices. Additionally, it is important to evaluate the MFA application strategies from the organizational perspective while also understanding the developers and security engineers' perspective. Thus, we plan to study how experts evaluate and implement security technologies such as MFA-applications for the users and how we can develop more secure applications. Another important aspect of this work is to understand the user trust for such security-focused tools and technologies.

8 Conclusion

MFA has become one of the primary tools for user authentication in recent years. The rapid implementation across platforms, applications, and services has increased user familiarity with MFA, but it is still difficult for users to accurately assess the security of MFA applications. This paper analyzes the code, cryptographic, and permission-based risks associated with mobile MFA applications and conduct a systematic literature review for articles published in 2020 and accessible prior to November 01, 2020. Ten MFA applications and their respective permission-based risks and code-based risks using the RiskInDroid and MobSF tools were analysed. The findings identified some worrying trends in several of these applications. Although some, such as the Microsoft Authenticator and Google Authenticator, were rated high by both tools, other applications had troubling privacy permission utilization and even more worrisome code-based risks that left the apps susceptible to a variety of exploits. Even more concerning is the fact that some of these applications are recommended by popular websites and experts. This means there are vulnerabilities in several of the applications analyzed in this paper that should trouble security professionals, given the increasing adoption of MFA applications over the past several years.

[8] https://www.kb.cert.org/vuls/id/836068

9 Acknowledgement

We would like to acknowledge the Security and Privacy Research Lab at the University of Denver and the students of the COMP 3705/4705 : Adv Topics: Human-Centered Data Security and Privacy for their initial feedback. Any opinions, findings, and conclusions or recommendations expressed in this material are solely those of the authors and do not necessarily reflect the views of the University of Denver.

References

[1] S. Banerjee, S. Roy, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. Rodrigues, and Y. Park. Multi-authority cp-abe-based user access control scheme with constant-size key and ciphertext for iot deployment. *Journal of Information Security and Applications*, 53:102503, 2020.

[2] N. Bhattacharya. Behavioural biometrics in action. *Biometric Technology Today*, 2020(10):8 – 11, 2020.

[3] C. Cimpanu. Securing your remote workforce against new phishing attacks. *Zdnet*, 2020.

[4] H. Crawford and E. Ahmadzadeh. Authentication on the go: Assessing the effect of movement on mobile device keystroke dynamics. In *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security*, SOUPS '17, page 163–173, USA, 2017. USENIX Association.

[5] S. Das. *A risk-reduction-based incentivization model for human-centered multi-factor authentication*. PhD thesis, Indiana University, 2020.

[6] S. Das, A. Kim, and L. J. Camp. Organizational security: Implementing a risk-reduction-based incentivization model for mfa adoption. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, 2021.

[7] S. Das, A. Kim, B. Jelen, J. Streiff, L. J. Camp, and L. Huber. Towards implementing inclusive authentication technologies for older adults. *Who Are You*, 2019.

[8] S. Das, A. Kim, B. Jelen, J. Streiff, L. J. Camp, and L. Huber. Why don't older adults adopt two-factor authentication? *Das, S., Kim, A., Jelen, B., Streiff, J., Camp, LJ, & Huber, L.(2020, April). Why Don't Older Adults Adopt Two-Factor Authentication*, 2020.

[9] S. Das, A. Kim, S. Mare, J. Streiff, and L. J. Camp. Security mandates are pervasive: An inter-school study on analyzing user authentication behavior. In *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*, pages 306–313. IEEE, 2019.

[10] S. Das, S. Mare, and L. J. Camp. Smart storytelling: Video and text risk communication to increase mfa acceptability. In *2020 IEEE Humans and Cyber Security Workshop (HACS 2020) in Association with the 6th IEEE International Conference on Collaboration and Internet Computing (CIC). IEEE*, 2020.

[11] S. Das, B. Wang, and L. J. Camp. Mfa is a waste of time! understanding negative connotation towards mfa applications via user generated content. In *Proceedings of the Thriteenth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2019)*, 2019.

[12] S. Das, B. Wang, A. Kim, and L. J. Camp. Mfa is a necessary chore!: exploring user mental models of multi-factor authentication technologies. 2020.

[13] S. Das, B. Wang, Z. Tingle, and L. J. Camp. Evaluating user perception of multi-factor authentication: A systematic review. In *Proceedings of the Thirteenth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2019)*, 2019.

[14] R. Duncan. A multi-cloud world requires a multi-cloud security approach. *Computer Fraud & Security*, 2020(5):11 – 12, 2020.

[15] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. Doostari. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care iot. *Computer Networks*, 177:107333, 2020.

[16] P. Gope. Pmake: Privacy-aware multi-factor authenticated key establishment scheme for advance metering infrastructure in smart grid. *Computer Communications*, 152:338 – 344, 2020.

[17] L. Janik, D. Chuda, and K. Burda. Sgfa: A two-factor smartphone authentication mechanism using touch behavioral biometrics. 2020.

[18] Kele. Republic of turkey identity card (tckk): Banking use case in authentication &; customer onboarding. In *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, ICCTA '20, page 35–40, New York, NY, USA, 2020. Association for Computing Machinery.

[19] S. Kelf. The security risks created by cloud migration and how to overcome them. *Network Security*, 2020(4):14 – 16, 2020.

[20] K. Krombholz, T. Hupperich, and T. Holz. Use the force: Evaluating force-sensitive authentication for mobile devices. In *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*, pages 207–219, 2016.

[21] H. Lee, D. Kang, J. Ryu, D. Won, H. Kim, and Y. Lee. A three-factor anonymous user authentication scheme for internet of things environments. *Journal of Information Security and Applications*, 52:102494, 2020.

[22] K. Mahmood, W. Akram, A. Shafiq, I. Altaf, M. A. Lodhi, and S. H. Islam. An enhanced and provably secure multi-factor authentication scheme for internet-of-multimedia-things environments. *Computers & Electrical Engineering*, 88:106888, 2020.

[23] J. Mohan and D. R. R. Enhancing home security through visual cryptography. *Microprocessors and Microsystems*, page 103355, 2020.

[24] K. Mohana Prabha and P. Vidhya Saraswathi. Suppressed k-anonymity multi-factor authentication based schmidt-samoa cryptography for privacy preserved data access in cloud computing. *Computer Communications*, 158:85 – 94, 2020.

[25] J. Reynolds, N. Samarin, J. Barnes, T. Judd, J. Mason, M. Bailey, and S. Egelman. Empirical measurement of systemic 2fa usability. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 127–143, 2020.

[26] A. Robles-González, J. Parra-Arnau, and J. Forné. A linddun-based framework for privacy threat analysis on identification and authentication processes. *Computers & Security*, 94:101755, 2020.

[27] M. L. Santos, J. C. Carneiro, A. M. Franco, F. A. Teixeira, M. A. Henriques, and L. B. Oliveira. Flat: Federated lightweight authentication for the internet of things. *Ad Hoc Networks*, 107:102253, 2020.

[28] M. Shuai, L. Xiong, C. Wang, and N. Yu. A secure authentication scheme with forward secrecy for industrial internet of things using rabin cryptosystem. *Computer Communications*, 160:215 – 227, 2020.

[29] F. Sinigaglia, R. Carbone, G. Costa, and N. Zannone. A survey on multi-factor authentication for online banking in the wild. *Computers & Security*, 95:101745, 2020.

[30] A. Storey. Where does biometrics sit in today's security ecosystem? *Biometric Technology Today*, 2020(7):9 – 11, 2020.

[31] P. D. Trung, N. N. Hai, and N. T. H. Ha. Secure eeg-based user authentication system integrated with robust watermarking. In *Proceedings of the Tenth International Symposium on Information and Communication Technology*, pages 242–247, 2019.

[32] C. Wang, Y. Wang, Y. Chen, H. Liu, and J. Liu. User authentication on mobile devices: Approaches, threats and trends. *Computer Networks*, 170:107118, 2020.

[33] D. Wang, X. Zhang, Z. Zhang, and P. Wang. Understanding security failures of multi-factor authentication schemes for multi-server environments. *Computers & Security*, 88:101619, 2020.

[34] J.-P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub, R. Couturier, and A. Chehab. Securing internet of medical things systems: Limitations, issues and recommendations. *Future Generation Computer Systems*, 105:581 – 606, 2020.

[35] J.-P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli. Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and Microsystems*, 77:103201, 2020.

[36] Z. Zhang, X. Chen, J. Ma, and J. Shen. Slds: Secure and location-sensitive data sharing scheme for cloud-assisted cyber-physical systems. *Future Generation Computer Systems*, 108:1338 – 1349, 2020.