

ANALYSIS OF REVERSE ENGINEERING AND CYBER ASSAULTS

Alibek Nurgaliyev

Thesis Submitted for the fulfillment of the requirements for the degree
Master of Research

Victoria University
Institute of Sustainable Industries and Liveable Cities

July 2023

Abstract

The rapid growth of cybercrime poses a significant threat in today's technological landscape, making cybersecurity a crucial concern for organizations and individuals alike. The shortage of cybersecurity professionals further exacerbates the risks faced by internal infrastructures worldwide. This study focuses on the domains of cyber security, reverse engineering, and encryption algorithms, aiming to address these pressing issues. By conducting a quantitative research method and employing meta-analysis techniques, the research provides an objective analysis and critical evaluation of the research problem. The primary objective of the study is to classify encryption algorithms based on various parameters, including flexibility, key expansion, possible attacks, entropy, and security vulnerabilities. This categorization is crucial for evaluating the effectiveness of different encryption algorithms in diverse applications. Another objective is to investigate methods for preventing information leakage in various fields of application. The study will showcase the process of reverse engineering and techniques to enhance application security. By thoroughly examining reverse engineering practices, vulnerabilities can be identified, and countermeasures can be developed against unauthorized data extraction and information theft. The third objective involves conducting in-depth research on common forms of online criminal activity and analyzing the findings to establish preventive measures. By comprehending these criminal activities and their consequences, effective strategies can be formulated to prevent or mitigate such crimes. This study contributes to the field of cybersecurity by investigating different systems, proposing strategies to prevent information leakage, and exploring encryption algorithms. Additionally, analyzing reverse engineering techniques helps identify vulnerabilities that can be exploited and empowers developers and security experts to fortify their applications. The outcomes of this research have been disseminated through publications and conferences, ensuring that the findings reach a wide audience of experts and researchers. By bridging the gap between theoretical research and practical application, this study enhances the security infrastructure of industries and contributes to creating a more secure and resilient digital ecosystem.

Declaration by author

“I, Alibek Nurgaliyev, declare that the Master of Research thesis entitled Analysis of reverse engineering and cyber assaults is no more than 50,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work”. “I have conducted my research in alignment with the Australian Code for the Responsible Conduct of Research and Victoria University’s Higher Degree by Research Policy and Procedures”.

Signature



Date

12/07/2023

Acknowledgements

I would like to express my sincere gratitude to my principal supervisor, Professor Hua Wang, for his invaluable guidance, unwavering support, and expert advice throughout my research journey. Professor Wang's deep knowledge, insightful perspectives, and continuous encouragement have been instrumental in shaping the direction of my work. I am truly grateful for the opportunity to work under his supervision.

I would also like to extend my thanks to the entire team led by Professor Hua Wang. The stimulating environment and collaborative spirit within the team have fostered a rich learning experience for me. I am grateful for the knowledge, expertise, and shared insights that I have gained from my interactions with team members. Their dedication to excellence and commitment to advancing research in our field have inspired me greatly.

Furthermore, I am indebted to my wife, Anara Nurgaliyeva, for her unwavering support and understanding throughout this challenging journey. Her constant encouragement, patience, and belief in my abilities have been a source of strength for me. I am deeply grateful for her sacrifices, as she stood by my side during the demanding times when I juggled two jobs, providing both emotional and practical assistance. Her love and unwavering belief in me have been my anchor, and I am truly fortunate to have her as my partner in life.

Finally, I would like to express my appreciation to all the individuals who have supported me in various ways during the course of my research. Your guidance, constructive feedback, and encouragement have been invaluable in shaping the outcomes of this work.

I am truly grateful for the support and contributions of all those mentioned above, as well as the many others who have played a part in my academic journey. Without their help, this research would not have been possible.

Publications included in this thesis

Some of the concepts, data, results, and figures presented in this thesis have been published in journals or conference proceedings during my Master of Research candidature. I, Alibek Nurgaliyev, certify that I was the primary contributor and author of each of these works. The publisher has granted the author permission to reproduce the contents of these publications for academic purposes. A complete list of the aforementioned publications is provided below.

1. A. Nurgaliyev and H. Wang, "Comparative study of symmetric cryptographic algorithms," 2021 International Conference on Networking and Network Applications (NaNA), Lijiang City, China, 2021, pp. 107-112, doi: 10.1109/NaNA53684.2021.00026.
2. A. Nurgaliyev and H. Wang, "Analysis of reverse engineering," 2023 15th International Conference on Advanced Computational Intelligence (ICACI), Seoul, Korea, Republic of, 2023, pp. 1-7, doi: 10.1109/ICACI58115.2023.10146175.

Table of contents

Contents

1	Introduction	8
1.1	Background	8
1.2	Overall aims	9
1.3	Contribution to knowledge and statement of significance	10
2	Literature Review	13
2.1	Overview on encryption algorithms	13
2.2	Information security overview	16
3	Methodology and Conceptual Framework	25
3.1	Data Encryption Standard	27
3.2	Triple Data Encryption Algorithm	28
3.3	The Advanced Encryption Standard (AES) Algorithm	30
3.4	The BLOWFISH Algorithm	32
3.5	The MARS Algorithm	34
3.6	Classification of the Symmetric Encryption Algorithms	36
4	Preventing Informational Leaks: strategies and tools	41
4.1	Reverse engineering. Common description	44
4.2	Practicing reverse engineering	49
4.2.1	colors.xml file	54
4.2.2	styles.xml file	55
4.2.3	Smali files	57
4.3	Obfuscation	58
5	Cybercrime Analysis	66

5.1	History of cybercrimes	67
5.2	Analysis of cybercrimes	71
5.3	Types of cyber crimes	72
5.4	Impact of cybercrime on society	96
6	Conclusion	101
6.1	Summary and conclusions of the thesis	101
6.2	Future work	102

1 Introduction

1.1 Background

In the current stage of societal evolution, the value of a company's information is on par with its goods and services, making it a valuable asset. Many businesses now commonly employ electronic storage and processing methods for their vast amounts of information. This practice not only enhances usability and interaction speed but also enables the automation of business processes. However, the potential benefits of these practices come hand in hand with the associated risks, particularly in terms of breaches of information confidentiality, integrity, and availability.

Exfiltration, also referred to as data theft, involves the unauthorized copying, transmission, or reception of data from unsuspecting victims' computers or servers. Exfiltration can occur over the Internet or within a local network. To prevent detection, hackers typically compress and encrypt the stolen data before transmitting it. Command and control servers and other transmission channels are commonly utilized by attackers to remove data from the targeted system.

One potential approach to address these security concerns is the reinforcement of cryptographic protocols. Cryptography involves the process of encoding and transmitting data in a manner that allows only authorized individuals to access it. It transforms data into an encrypted code, ensuring secure transmission over public networks. Encryption technology has facilitated numerous innovative developments and applications. This study aims to provide an unbiased comparison of well-known and widely used data encryption algorithms while identifying potential flaws and information leakage risks [1, 2, 3, 4].

The primary distinguishing characteristics of encryption algorithms include the time required for data encryption, the efficiency of the encryption process, and the level of protection they offer against various types of attacks [5]. The objective of this project is to evaluate these algorithms using a variety of parameters, thus considering their behavior and performance under different data loads. By conducting this comparison, the study

aims to shed light on the strengths and weaknesses of these algorithms, aiding in the selection of appropriate encryption methods based on specific security requirements.

Through comprehensive analysis and evaluation, this research contributes to enhancing the understanding of data encryption algorithms and their suitability for safeguarding sensitive information. The findings will support decision-making processes in choosing effective encryption approaches and optimizing data protection measures. Ultimately, this study aims to strengthen information security practices and reduce the risk of data breaches in an increasingly digitized world.

1.2 Overall aims

The primary objective of this study is, as indicated by the research question, to provide a number of suggestions for how the safety of information systems can be improved. In order to accomplish this general objective, the following subgoals have been outlined:

- Aim 1: To categorize the primary uses of encryption algorithms according to parameters like flexibility, key expansion, possible attacks, entropy, and security vulnerabilities of algorithms. These parameters are what determine how effective a cryptosystem is, so classifying the main uses of encryption algorithms is important.
- Aim 2: To discover means whereby the information can be prevented from being leaked in various fields of application. The process of reverse engineering should be demonstrated, and methods should be provided that will strengthen the security of applications.
- Aim 3: To conduct in-depth research into common forms of online criminal activity, analyze the results of those investigations, and reach conclusions about how to prevent or stop those crimes.

Aim 1 and Aim 2 exhibit a certain degree of interconnectedness since the information categorized within Aim 1, which pertains to the security vulnerabilities of encryption

algorithms, can have direct relevance to the objective of preventing information leaks outlined in Aim 2. An understanding of encryption algorithm strengths and weaknesses, as explored in Aim 1, plays a pivotal role in bolstering the security of applications, as delineated in Aim 2.

When it comes to Aim 3, which centers on the investigation of online criminal activities, it may not be directly sequential in its relationship with Aim 1 or Aim 2. It appears to be a distinct research objective that does not necessarily hinge on the fulfillment of the other aims. Nonetheless, the findings derived from Aim 1 and Aim 2 could potentially offer insights or contribute to our comprehension of online criminal activities in some capacity. In summary, while these aims do not strictly follow a sequential order, they do possess a certain level of interrelation. Aim 1 and Aim 2 exhibit more direct connections owing to their shared emphasis on security and encryption, while Aim 3 represents a somewhat separate research objective associated with online criminal activities.

1.3 Contribution to knowledge and statement of significance

The inadvertent exposure of confidential information presents a substantial risk for numerous companies. Such incidents can occur either due to deliberate actions by third parties or the negligent behavior of employees. Deliberate disclosures can be driven by two primary objectives: the first being to inflict harm on the state, society, or a specific business, which is often associated with cyberterrorism manifestations, while the second objective is to gain a competitive advantage over other businesses. On the other hand, employee negligence within an organization stands as the most common cause of accidental leaks, which can also result in severe unintended consequences and significant adverse outcomes.

In essence, safeguarding the confidentiality of information revolves around the prevention of information leakage. Disclosing confidential information leads to direct financial losses, intellectual property compromise, damage to the organization's reputation, and diminished trust from customers and partners [6, 7, 8].

Addressing the issue of confidentiality violation and mitigating the risks of leakage necessitate an integrated approach that combines technical and organizational measures. It is essential to recognize that solely relying on technical means or organizational methods alone is insufficient in most situations. This research endeavors to identify methods for preventing information leakage across various domains and to assess the strengths and weaknesses of encryption algorithms in specific scenarios [9, 10, 11].

This study is expected to make valuable contributions to the field of information security. Firstly, it will bridge the talent gap prevalent in many countries, particularly in the realm of information security. By providing a comprehensive and in-depth understanding of cryptography, this research will benefit both newcomers and experienced professionals in the field. Secondly, the study aims to offer a comprehensive comparison of various encryption algorithms currently available. This will enable businesses to transition from traditional algorithms to those better suited to their specific needs, ultimately reducing the reliance on cryptographic software across organizations. Thirdly, conducting an analysis of cybercrimes will assist in identifying vulnerabilities in existing information systems. Even if a company has not yet fallen victim to a cyber attack, it does not imply that its defenses are foolproof. By conducting a thorough analysis, this research aims to shed light on potential problem areas, strengthening the overall security posture of organizations.

In the specific context of cryptographic algorithms and information security, reverse engineering can be connected as follows:

- **Cryptanalysis:** Reverse engineering techniques can be applied to cryptographic algorithms to analyze their inner workings, discover weaknesses, and identify potential vulnerabilities. This is essential for ensuring that encryption algorithms remain secure against attacks.
- **Key Management:** Reverse engineering can help experts understand how encryption keys are generated, stored, and managed in software and hardware systems. This is crucial for ensuring the confidentiality of cryptographic keys.

- **Security Protocols:** Reverse engineering can be used to evaluate the implementation of security protocols, such as SSL/TLS, in software applications. This helps identify security issues and improve the overall security of data transmission.

In summary, reverse engineering is a significant process in the fields of information security and cryptography because it enables experts to analyze, assess, and improve the security of systems, including cryptographic algorithms and their implementations. It plays a vital role in identifying vulnerabilities, understanding proprietary systems, and strengthening the security of various components in the information security ecosystem.

By uncovering novel insights and providing practical recommendations, this study seeks to contribute to the advancement of information security practices and promote proactive measures in combating information leakage and cyber threats. Ultimately, the findings of this research can inform the development and implementation of robust strategies to protect confidential information and enhance the overall resilience of organizations in an ever-evolving threat landscape.

2 Literature Review

2.1 Overview on encryption algorithms

Qadir and Varol (2019) have written a review article on cryptography in which they familiarize the reader with the history and development of cryptography [12]. The authors conducted a thorough literature review, during which they read articles not only about cryptographic algorithms, but also about digital data transmission, network security, and the significance of secure cloud computing. After that, they explained the fundamentals of cryptography. This research also provides an overview of the development of algorithms, from the Caesar cipher, which is considered to be one of the first algorithms, to more recent algorithms that use stream or block ciphers. In addition to this, the article provides an explanation of hash functions and possible conflicts, as well as setting out the requirements and principles of the digital signature [13, 14, 15]. RSA is by far the most widely used cryptographic algorithm for performing this kind of authentication due to the fact that the digital signature is a public key algorithm. The Rivest–Shamir–Adleman (RSA) public-key cryptosystem is one of the most popular methods for ensuring the safety of data transmission. Having said that, there is a significant lack of research. The authors of this article only provide a brief summary of the underlying working principle of a few cryptographic procedures; more specific examples, names of algorithms, or even the implementation of the algorithms themselves are not provided. There is only a small amount of information provided about public key algorithms (asymmetric algorithms), but there is no information provided about private key algorithms (symmetric algorithms).

Ebrahim and his colleagues [16] conducted a study that compared and analyzed a number of symmetric algorithms that are widely used and have a solid track record. The evaluation is carried out with respect to the primary characteristics of the algorithms, which are as follows: security, scalability, reliability, flexibility, and architecture. The first section of the research offers an introduction to a variety of symmetric algorithms, including DES, 3DES, BLOWFISH, IDEA, TEA, CAST, Rijndael, RC6, Serpent, Twofish, and

MARS, among others. If we examine these algorithms from an architectural perspective, we will notice that all of them, with the exception of IDEA, have a Feistel structure, and the data is displayed in the form of a table [17]. The chapter titled "Security" provides a list of the possible attacks and demonstrates the strength of the cryptographic protocol. Memory utilization and the speed at which encryption algorithms operate are both aspects of the scalability feature. This quality is very significant; however, it is challenging to design a cipher that is scalable across all different kinds of platforms [18, 19, 20]. The capacity to adjust one's behavior in response to shifting personal priorities is one definition of the quality known as flexibility. Finally, the factor that is of the utmost importance is safety. A few of these algorithms suffer from significant deficiencies or confinements. The selection of an algorithm for a variety of information systems is the responsibility of this parameter. Based on the findings of the research, the Rijndael (AES) algorithm is the most secure, in addition to being the quickest and having no significant drawbacks. Although other algorithms have also proven to be effective, the majority of them have a compromise between the amount of memory they use and how well they encrypt data, and several algorithms have been found to be vulnerable [21, 22].

"Masram et al. (2014) present an analysis and comparison of several symmetric key cryptographic ciphers based on encryption time with the variation of various file features such as different data types, data size, data density, and key sizes [23]." The symmetrical algorithms of RC4, AES, Blowfish, RC2, DES, Skipjack, and Triple DES are the focus of this work. In this paper, information regarding the various symmetric key cryptographic algorithms that are to be analyzed for performance evaluation is presented, and the paper also displays the results that were obtained from a variety of sources. Comparisons of cryptographic algorithms have been made for the purpose of determining which one offers superior performance in terms of throughput, CPU Memory utilization, energy consumption, attacks, Encryption time, Decryption time, and other relevant metrics. It came to the attention of the authors, while they were conducting a review of previously published material, that none of the works did a very detailed analysis of the performance of various

symmetric algorithms on various parameters on the various types of files, particularly the files that are used for medical health-related data[24, 25, 26].

The variety of data types was decided to be the first criterion to consider. It was decided to use a variety of data files, including audio, image, textual, and video, totaling close to 50 megabytes in size. The findings that were presented in the tables and diagrams demonstrated that the amount of time required for encryption does not change depending on the type of data [?]. In order to ensure once more that the observations obtained in the first case were accurate, data files of the same type but of varying sizes were collected. As a consequence of this, the amount of time required to encrypt data grows proportionally with the size of the file in multiples of the data size. In order to determine whether or not the encryption is dependent on the data density, a test was run using files of varying amounts of data storage space. A sparse file with 69 megabytes and a dense file with 58.5 megabytes are both used to conduct an analysis of the encryption rate for the respective data density. The amount of data packed into a file has no bearing on how long it takes to encrypt it. The authors were able to determine the impact that varying the size of the encryption key had on the amount of time required for encryption by conducting experiments with encryption algorithms that featured keys of varying sizes. It was decided to use a BMP file that was 50.5 megabytes in size, and then various cipher algorithms were run on it using the ECB mode and the PKCS#5 padding scheme. Each algorithm was run on a different size of key that was supported by the cipher. The results of the execution demonstrated that the amount of time required for encryption shifts in response to variations in the size of the key for all cipher algorithms [27, 28, 29].

An exhaustive study of symmetric algorithms, which was based on a large number of experiments involving different types of files and criteria, revealed that encryption is solely dependent on the number of bytes that are present in the file. It was also discovered that the amount of time required for encryption and the size of the data are proportional to one another. On the other hand, this research was carried out solely within symmetric algorithms [30, 31, 32].

Maqsood et al. (2017) provide a comparison of various cryptography algorithms [33]. This study conducts a literature review on previously published works that concentrate solely on symmetric or asymmetric encryption techniques. The performance of various symmetric and asymmetric algorithms was evaluated by the authors by covering multiple parameters, such as the amount of time required for encryption and decryption, the amount of time required to generate keys, and the size of the files. Only the most common parameters are included in a criterion for comparing different performances. In addition, the analysis was based on simulations that were run in a sample environment in which multiple cryptography algorithms were compared to one another. In order to conduct an investigation into symmetric algorithms, the DES (Data Encryption Standard), 3DES (Triple Data Encryption Standard), and AES (Advanced Encryption Standard) algorithms were selected [34]. The RSA (Rivest, Shamir, and Adleman) algorithm, Elgamal, and ECC are examples of asymmetric algorithms (Elliptic Curve Cryptography). The primary purpose of this study is to provide an evaluation of the performance of cryptographic schemes, such as symmetric and asymmetric algorithms. The accuracy of the analysis was improved as a result of the authors' testing of the performance of various cryptographic systems on files of varying sizes. When compared to symmetric algorithms, the results that have been visualized indicate that the use of asymmetric algorithms for encryption and decryption requires a significant amount of additional time.

2.2 Information security overview

Mobile devices have become an indispensable component of our everyday lives; for example, we keep personal information, official documents, and photos on our mobile devices and even conduct online banking with them. Intruders and hackers are gaining an increasingly easier time gaining access to this sensitive personal information with each passing day [35, 36, 37, 38]. Work that can be done remotely or from home has seen a surge in demand since the pandemic began. At the same time, employers should raise the level of data security awareness among employees, as more and more people are work-

ing on their own personal devices and transmitting sensitive information through instant messengers and the mail. While using a mobile device at home and at work, fundamental security principles are required in order to protect the organization from suffering the consequences of a data breach. In the well-known attack known as cybercrime, the perpetrators of the crime gain access to employee mobile apps and other sources, and then use this information to infiltrate businesses and claim their requirements. Application development should be audited to restrict the availability of unknown apps in their store, and the establishment of a secure connection is necessary for mobile security [39, 40, 41]. For instance, VPN protocols should be configured when using mobile devices, and TLS/SSL connections should be checked whenever a web browser is used to access websites. In addition, blocking any strange SMS messages can be helpful in protecting against phishing attacks. There are a lot of articles and studies that talk about the features of mobile devices as well as the dangers of using mobile devices. Social engineering, which refers to the practice of manipulating users psychologically in order to steal data, has emerged as a primary concern in the realm of information security as a means of addressing the aforementioned issues. It is imperative that stringent data protection rules, such as the General Data Protection Regulation, be adhered to in order to ensure that only authorized parties have access to sensitive information [36, 42, 43]. The articles that follow both describe and offer solutions to various issues regarding device security awareness.

In their research work [44], Venkata Mukesh Marabathuni and Viktors Gopejenko described the various types of attacks that can occur in mobile security and how to defend against them. They also provided solutions. They investigated what kinds of defense tools and strategies could be used, as well as the steps involved in downloading an application and granting permissions for it. According to the findings of the article, the majority of regular mobile users are not aware when their data is being stolen. They have problems with the security of their mobile devices because they are careless and unaware. If you download apps from untrusted sources or give them permissions they do not need, you run the risk of downloading viruses or having personal information leaked. Uploading one's

personal files to a cloud service that can be trusted will protect them from the possibility of the device being formatted. Additionally, system updates of a high-quality should be provided to the device.

Universalizing the global anti-phishing mechanism and the technologies that are related to it is the primary objective of the research project titled "Design and Implementation of Mobile Phone Information Security System based on Android" [45]. Additionally, to create a mobile application that can detect phishing when it is performed on a device. Phishing attacks involve the practice of sending fraudulent messages that give the impression that they came from a reliable source. The objective is to obtain sensitive data from the victim's computer or mobile device, such as credit card and login information, or to install malware on either of those devices. The authors of this article have developed software management features with the goals of improving product usability and enhancing user safety. This has been implemented on the basis of the many different types of security products that are currently available on the market, with the protection of communications and processes serving as the primary focus. Every single person who uses an Android device should give serious consideration to Android security, not just researchers and IT experts [46]. The authors of this study created a mobile phone security guard based on Android, and they used Android Development Tools, embedded database SQLite, and Java as the development language. The mobile phone security guard is designed to protect smartphones. The authors state that the system has been debugged and put through its paces in terms of testing to ensure that it is ready to be released to the public and used by actual customers and users. However, there are some flaws in the design and development of the system, such as the absence of automatic system fault repair and the fact that packet intercept has not yet been achieved, among other things. In the future, the authors intend to keep working toward the goal of developing a more functional and reliable Android protection system that can be sold on the Android market. Additionally, they will continue to enhance the functions of the mobile phone information security guard [47]. Nevertheless, it is important to note that the [National Innovation and Entrepreneurship

Education Program for Students 2019] was the source of funding for this particular research project that was awarded the grant.

The article titled "Secure Payment With NFC Mobile Phones In The Smart Touch Project" [48] is of utmost significance because it discusses and investigates the various security concerns that arise when payments are processed using NFC technology. These days, people use their mobile phones for everything, including making payments when they want to buy something. The vast majority of users aren't concerned about security, but the programmers and information technology specialists put in a lot of work to ensure that the solutions and implementations they create for NFC payment applications are safe and accurate technically. In the first section of the paper, the definitions and the technical realization for this pilot project that is only applicable to a portion of the system are discussed. This paper demonstrates, in the second section, that the primary concern is the various security concepts. The study demonstrates that payments made using NFC should have the same level of security as a standard payment made using an EMV transaction. The methodology that pertains to the payment application and its communication with the payment terminal through an NFC link is discussed in the concluding section of this article. This study, which was carried out as part of the ITEA SmartTouch project, draws its inspiration from a pilot project that was carried out in Strasbourg, which is located in France, in the year 2007. This was the very first prototype of an NFC-based payment application, and it fully supported both the international standard and the PayPass program. This was the first attempt at doing so. In the final section of this article, a variety of solutions to the issues that have arisen are discussed from a variety of perspectives [49, 50, 51].

The authors of the project called ITEA12 SmartTouch have allotted one year for the completion of the work package security necessary to finish the application of the Common Criteria method on NFC mobile payment. On the other hand, there were two significant problems found with the overall criteria used in the study, both of which could potentially limit the generalizability of safety studies:

- One concerns the Common Criteria method application
- The second concerns point two of the Protection Profile definition.

The safety conditions that prevailed throughout the lifetime of the SIM card. When it comes to the application of the Common Criteria method, the difficulty for a single organization to ensure "state of the art" coverage of the whole range of attacks on multi chips and complex systems like NFC mobile payment can be circumvented in two different ways. The first option is to establish a research consortium, similar to the partnership that exists between the security lab of Gemalto and the public laboratory of Greyc. The formation of specialized businesses is the second course of action that can be taken. Regarding the second point of the definition of the Protection Profile, which is "the security environment during the life cycle of the SIM and particularly its phase of personalization," it is necessary to provide a more comprehensive description of the issue [52, 53, 54].

The NFC mobile payment has a lot of positive potential benefits, including: being simple to use, giving the impression of being secure, being able to benefit from two marketing networks at once, namely those of telecom operators and banks, making mobile payments more convenient for cardholders, being quick and easy to use, and being compatible with both the existing payment networks and the MasterCard and VISA PayPass standard. The difficulty lies in demonstrating that the level of security offered by an NFC mobile phone, which is used to complete a payment transaction, is equivalent to that offered by a smart-card. Even though putting it into practice can be difficult, using the Common Criteria method is a good way to compare and evaluate different security solutions [55, 56, 57].

The problem of technology-enabled abuse that is faced by survivors of intimate partner violence (IPV), whose accounts and devices may be physically or remotely accessed by an abuser who may have knowledge about the survivor's personal details, is what motivated the research that was conducted about mobile security strategies and usability problems in IPV and stalking contexts [58]. The IPV technology abuse threat model is distinguished by the presence of UI-bound adversaries, or abusers who are constrained in their actions by the capabilities provided by a system's user interface (UI). Given the dire

need for usable tools that are immediately accessible to IPV victims, the research focuses on the usability of existing interfaces and strategies to counter unsophisticated threats. Specifically, the research investigates whether or not these can be used. Interviews with nine participants, some of which were semi-structured, were conducted by the authors to discuss rooting and jailbreaking, as well as location tracking, compromised accounts, and spyware apps.

The research questions:

1. How informed are participants about these potential threats to their safety?
2. How knowledgeable are participants about the ways in which these security problems can be solved?
3. Are the participants capable of successfully navigating the interfaces in order to find solutions?
4. How straightforward is the procedure for resolving the issues, in the participants' opinions?

In the research paper [58], it was discovered that although the participants had a passing familiarity with the security risks and concepts being discussed, many of them, and sometimes all of them, struggled to find solutions to security problems in which the risks being discussed were simulated. The interviewers recruited participants through the Computer Gigs section of Craigslist for Pittsburgh and Los Angeles, and then screened them based on the following criteria: they needed to be at least 18 years old, located in the United States, proficient in English, and have access to a device that can connect to the internet to run Zoom. Additionally, they needed to use an iPhone. The questionnaire featured a general demographic section, but its sole purpose was to facilitate purposive sampling and ensure that the sample was representative of all relevant demographic groups.

During the interview, the participants were asked questions regarding the safety of their mobile phones, and they were given four fictitious examples of mobile security

breaches. In order to simulate the various security scenarios, an iPhone that had been specially formatted for this study was used, and the screen was remotely shared with the participants. During the interview, the participants were asked to guide us through their strategy for assisting their "friend" or "coworker" in resolving the security issues. The security threats were projected onto a fictional persona that represented the participant and their "friend" or "coworker," respectively. A gift code for \$20 can be found in each participant's Amazon account. The authors conducted a qualitative thematic analysis of the interviews by first coding the transcripts of the interviews, then collecting keywords and ideas from the interviews, and finally identifying common themes. The group recorded the results of the interviews in a code book, and each transcript was individually coded by a different member of the research team. It was also recorded whether or not the participants correctly identified the issue with a scenario, as well as the hints that were provided to them for each scenario. The authors came to a conclusion and detailed their choices after examining each scenario. On the other hand, the paper had a few flaws, such as a restricted number of people who took part in the interview and a restricted number of potentially malicious apps for certain scenarios. Because there were only 9 people involved in the study, the data are insufficient to generalize the findings. The two iPhones that were used in the experiments each had three apps that were not installed by default. These apps were Zoom, TeamViewer, and Google Maps. It is likely that the average iPhone user has a greater number of apps installed on their device, which would make it more difficult to identify spyware on the device. This paper demonstrates that even if a person is relatively familiar with some security risks, they are unable to articulate them in detail even if they try. Many of the participants found that finding solutions to their security issues was challenging or counterintuitive. The use of android smartphones has grown exponentially in recent years. Additionally, this growth has led to an increase in the number of malware attacks targeting Android devices. Malware attacks vary in complexity and are often targeted towards known weaknesses in the system. Therefore, a comprehensive framework for analyzing malware on android systems is essential. The article,

“Framework for malware analysis in Android” by Urcuqui, Christian., & Navarro, Andres, proposes a framework that is specifically designed for this purpose. This literature review aims to critique and provide an in-depth analysis of the article.

Critical evaluation: The article presents an overview of the existing techniques and frameworks used in analyzing malware in android systems. Urcuqui and Navarro highlight the limitations of current malware analysis frameworks, which fail to test behavior-based analysis and extract useful results for developers. The proposed framework integrates the best features of existing frameworks to provide a comprehensive solution for malware analysis. The authors emphasize the importance of analyzing not only the code but also the behavior of malware. The framework presented in this paper addresses the problem of behavior-based analysis by allowing the testing of malware in a controlled environment.

The authors present a four-step systematic process for the detection and analysis of malware in android devices. The process includes data collection, preprocessing, analysis, and post-processing. Data collection involves the gathering of information about the malware, such as static and dynamic analysis data. Preprocessing includes the cleaning of data and the preparation of the data for analysis. Analysis of the data tests the malware’s behavior by executing it in a controlled environment. Finally, post-processing involves the generation of a report on the malware’s behavior and characteristics.

The primary strength of the authors’ framework is its ability to conduct behavior-based analysis and provide useful information for developers. The framework enables the identification of malware’s behavior and characteristics, making it easier to understand the malware and develop countermeasures against it. The authors also address the issue of analyzing the latest malware, which may be more complex and difficult to analyze than previous malware. The proposed system analyzes Android applications in a controlled laboratory environment and has the capability to handle advanced malware attacks.

However, the framework may have some limitations. First, the framework may not be able to analyze attacks that occur in real-time, which requires the development of a more

sophisticated analysis system. Second, the framework does not address the issue of mobile device security, which may impact the successful implementation of this framework.

In conclusion, the article "Framework for malware analysis in Android" by Urcuqui, Christian., & Navarro, Andres, provides an essential contribution to the field of malware analysis in android systems. The framework proposed in this paper is efficient and effective in identifying malware's behavior and characteristics. The primary strength of the authors' framework is that it conducts behavior-based analysis, which provides useful data for developers. However, the framework has some limitations, including its inability to deliver real-time analysis and its failure to address mobile device security. Overall, the paper provides a valuable contribution to the field and highlights the importance of developing comprehensive malware analysis frameworks.

3 Methodology and Conceptual Framework

In today's digital age, the need for secure communication and data protection is of paramount importance. Encryption algorithms play a crucial role in ensuring the confidentiality, integrity, and authenticity of sensitive information. This chapter provides an overview of encryption algorithms, their purpose, and their significance in modern communication systems.

Encryption algorithms are mathematical procedures that transform plaintext into ciphertext, making it unreadable to unauthorized individuals. The primary objectives of encryption algorithms are to protect data from unauthorized access, prevent data tampering, and ensure secure communication channels. Encryption algorithms use keys to control the encryption and decryption processes, providing a means for authorized parties to access the original data.

Symmetric encryption algorithms, also known as secret-key encryption algorithms, use the same key for both encryption and decryption processes. They are generally faster and more efficient than asymmetric encryption algorithms, making them suitable for bulk data encryption. Popular symmetric encryption algorithms include the Data Encryption Standard (DES), Advanced Encryption Standard (AES), and the Triple Data Encryption Algorithm (3DES).

Asymmetric encryption algorithms, also known as public-key encryption algorithms, use a pair of keys: a public key for encryption and a private key for decryption. This approach provides a higher level of security and enables key exchange between parties without requiring a secure channel. Well-known asymmetric encryption algorithms include the Rivest-Shamir-Adleman (RSA) algorithm and the Elliptic Curve Cryptography (ECC) algorithm.

Hash functions are cryptographic algorithms that transform data of any size into fixed-size hash values. They are commonly used in conjunction with encryption algorithms to ensure data integrity. Hash functions have various applications, including digital signatures, password storage, and data verification. Examples of popular hash functions are the

Secure Hash Algorithm (SHA) family and the Message Digest Algorithm (MD5).

Effective key management is crucial for the security of encryption algorithms. It involves key generation, distribution, storage, and revocation. Key management systems employ various techniques such as key escrow, key rotation, and key recovery to ensure the confidentiality and availability of keys. Additionally, key lengths and complexity play a significant role in the strength of encryption algorithms.

Cryptanalysis refers to the study of cryptographic systems with the aim of breaking their security. It involves analyzing encryption algorithms and attempting to find weaknesses that can be exploited. Cryptanalysis techniques include brute-force attacks, differential cryptanalysis, and side-channel attacks. The process of cryptanalysis is essential for evaluating the strength of encryption algorithms and identifying potential vulnerabilities.

Encryption algorithms find extensive applications in various domains. They are crucial for secure communication over the internet, including email encryption, virtual private networks (VPNs), and secure web browsing. Encryption algorithms are also used to protect sensitive data stored on computer systems, such as database encryption, file encryption, and disk encryption. Additionally, encryption plays a vital role in securing financial transactions and digital signatures.

Encryption algorithms are indispensable tools in ensuring secure communication and data protection in the digital world. They provide the means to safeguard sensitive information, prevent unauthorized access, and establish trust in electronic systems. This chapter has provided an overview of encryption algorithms, including symmetric and asymmetric encryption algorithms, hash functions, key management, cryptanalysis, and their applications. In the following chapters, we will delve deeper into specific encryption algorithms, their underlying principles, and their practical implementations in various scenarios.

3.1 Data Encryption Standard

The Data Encryption Standard (DES) is a symmetric encryption algorithm that has been widely used for secure communication and data protection since its introduction in the 1970s. This chapter provides an in-depth exploration of the DES algorithm, its history, key features, and its role in modern cryptographic systems.

The development of DES can be traced back to the early 1970s when the need for a standardized encryption algorithm arose. The National Bureau of Standards (now the National Institute of Standards and Technology) initiated a project to develop a secure and efficient encryption algorithm. The result was the creation of the Data Encryption Standard (DES), which was adopted as a federal standard in the United States in 1977.

The DES algorithm operates on 64-bit blocks of data and uses a 56-bit key for encryption and decryption. The key is derived from a user-provided 64-bit key, where 8 bits are used for parity and not directly involved in the encryption process. The algorithm consists of several rounds of complex operations, including permutation, substitution, and XOR operations, which provide confusion and diffusion to achieve security.

DES supports various modes of operation to handle different encryption scenarios. The Electronic Codebook (ECB) mode is the simplest and most straightforward, where each block is encrypted independently. The Cipher Block Chaining (CBC) mode adds feedback from the previous block to the encryption process, providing better security for consecutive blocks. Other modes include Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) mode, each with its unique characteristics and security properties.

DES was designed to provide a high level of security, but over time, its effectiveness has diminished due to advances in computing power and cryptanalysis techniques. The primary strength of DES lies in its simplicity, efficiency, and widespread adoption, which contributed to its popularity for several decades. However, its key length of 56 bits has become a significant weakness, as it is susceptible to brute-force attacks. In 1999, a successful demonstration of breaking DES encryption using a distributed computing network

highlighted the need for stronger encryption algorithms.

To address the limitations of DES, the Triple Data Encryption Algorithm (3DES) was introduced. 3DES applies the DES algorithm three times using different keys, making it significantly more secure than the original DES. It provides backward compatibility with DES while increasing the effective key length to 168 bits. 3DES is widely used in legacy systems and as an intermediate solution during the transition to more advanced encryption algorithms.

Despite its declining security, DES continues to have a presence in certain applications. It is used in legacy systems where compatibility and interoperability with older technologies are required. Additionally, DES serves as a building block in more complex cryptographic systems, such as the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols, where it is used in combination with other encryption algorithms.

The Data Encryption Standard (DES) algorithm has played a significant role in the field of cryptography for several decades. It was the first widely adopted encryption standard, providing a foundation for secure communication and data protection. While its original 56-bit key length is now considered inadequate for modern security requirements, DES still has relevance in certain applications and continues to influence the development of newer encryption algorithms. This chapter has provided an overview of the DES algorithm, including its history, key features, modes of operation, strengths, weaknesses, and its role in modern cryptography. In the following chapters, we will explore more advanced encryption algorithms and their practical implementations.

3.2 Triple Data Encryption Algorithm

The Triple Data Encryption Algorithm (3DES) is an enhanced version of the Data Encryption Standard (DES) algorithm. It addresses the limitations of DES by applying the algorithm multiple times, using different keys. This chapter provides an in-depth analysis of the 3DES algorithm, its key features, modes of operation, and its significance in modern cryptographic systems.

With the decline in the security of DES due to its short key length, the need for a stronger encryption algorithm became apparent. 3DES emerged as a solution that enhanced the security of DES while maintaining compatibility with existing systems. It was first introduced in the 1990s as an interim solution before the transition to more advanced encryption algorithms.

3DES operates on 64-bit blocks of data and employs a key length of either 112 bits or 168 bits. It applies the DES algorithm three times in a cascade manner. There are two keying options for 3DES: 2-key 3DES (112-bit key) and 3-key 3DES (168-bit key). The algorithm uses a combination of encryption, decryption, and re-encryption operations to ensure security and compatibility with legacy systems.

Similar to DES, 3DES supports various modes of operation to accommodate different encryption requirements. The most commonly used mode with 3DES is the Cipher Block Chaining (CBC) mode. Other modes, such as Electronic Codebook (ECB), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) modes, can also be used with 3DES. These modes provide different levels of security and suitability for specific applications.

3DES offers a significant improvement in security compared to DES. The use of multiple encryption rounds and a longer key length make 3DES highly resistant to brute-force attacks. It provides backward compatibility with DES, enabling seamless integration with existing systems. However, 3DES suffers from slower performance due to the increased computational overhead of applying the algorithm three times. The transition to more efficient and secure encryption algorithms, such as the Advanced Encryption Standard (AES), has limited the use of 3DES in modern applications.

Although 3DES is more secure than DES, its security level is not as high as newer encryption algorithms. The increasing computational power of modern computers has reduced the time required to execute brute-force attacks on 3DES. As a result, the National Institute of Standards and Technology (NIST) has recommended transitioning to more advanced algorithms, such as AES, for stronger security.

3DES has been widely used in various domains where compatibility with legacy systems is essential. It has found applications in financial transactions, secure communications, virtual private networks (VPNs), and storage systems. Despite its declining popularity, 3DES is still utilized in certain industries and government sectors that have specific regulatory requirements or rely on older technologies.

The Triple Data Encryption Algorithm (3DES) has served as an important transitional encryption algorithm in the evolution of secure communication and data protection. By applying the DES algorithm three times using different keys, 3DES enhanced the security of DES and maintained compatibility with existing systems. However, with the advent of more advanced encryption algorithms, the use of 3DES has diminished. This chapter has provided an in-depth examination of the 3DES algorithm, including its key features, modes of operation, strengths, weaknesses, and applications. In the following chapters, we will explore more advanced encryption algorithms and their practical implementations [59].

3.3 The Advanced Encryption Standard (AES) Algorithm

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm that has become the de facto standard for secure communication and data protection. It replaced the Data Encryption Standard (DES) algorithm as the recommended encryption standard by the National Institute of Standards and Technology (NIST) in 2001. This chapter provides an in-depth exploration of the AES algorithm, its key features, modes of operation, and its significance in modern cryptographic systems.

The need for a stronger encryption algorithm to replace DES became evident as computational power increased. In response, NIST initiated a competition in 1997 to select a new encryption standard. After a rigorous evaluation process involving various candidates, the Rijndael algorithm, developed by Joan Daemen and Vincent Rijmen, was chosen as the AES algorithm in 2001.

The AES algorithm is a block cipher that operates on fixed-size blocks of data. It

supports block sizes of 128 bits, with key sizes of 128, 192, or 256 bits. AES employs a substitution-permutation network (SPN) structure and consists of several rounds of operations, including substitution, permutation, and mixing operations. These operations provide confusion and diffusion to ensure security.

AES supports various modes of operation to handle different encryption scenarios. The most commonly used mode is the Cipher Block Chaining (CBC) mode, which provides confidentiality and integrity by XORing the previous ciphertext block with the current plaintext block before encryption. Other modes, such as Electronic Codebook (ECB), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) modes, offer different security properties and suitability for specific applications.

AES is widely regarded as a highly secure encryption algorithm. Its strength lies in its resistance to various cryptanalysis techniques, including differential and linear attacks. The algorithm's design and rigorous evaluation process have contributed to its robustness. The use of longer key lengths, up to 256 bits, significantly increases the security level. AES has been adopted and extensively used in a wide range of applications requiring secure communication and data protection.

One of the key advantages of AES is its efficient and fast execution on modern computing platforms. AES has been optimized for various hardware and software implementations, enabling high-performance encryption and decryption operations. The algorithm's simplicity and regularity facilitate parallel processing, making it suitable for applications with high throughput requirements.

AES provides different security levels based on the chosen key length. AES-128 offers a high level of security and is considered secure against all known practical attacks. AES-192 and AES-256 provide even higher security levels and are recommended for applications that require enhanced protection. The selection of the appropriate key length depends on the desired level of security and the specific requirements of the application.

AES has found widespread applications in various domains requiring secure communication and data protection. It is used in secure communication protocols, such as

Transport Layer Security (TLS) and Secure Sockets Layer (SSL). AES is also employed in file and disk encryption, database encryption, virtual private networks (VPNs), and wireless communication security. Its versatility and robustness have made it a fundamental component of modern cryptographic systems.

The Advanced Encryption Standard (AES) algorithm has established itself as the most widely adopted symmetric encryption algorithm. Its security, efficiency, and versatility have made it a cornerstone of secure communication and data protection. This chapter has provided an in-depth analysis of the AES algorithm, including its key features, modes of operation, strengths, security levels, and applications. In the following chapters, we will delve into other advanced encryption algorithms and their practical implementations [60].

3.4 The BLOWFISH Algorithm

The BLOWFISH algorithm is a symmetric block cipher designed by Bruce Schneier in 1993. It is known for its simplicity, speed, and flexibility. BLOWFISH operates on variable-length blocks and supports key lengths ranging from 32 to 448 bits. This chapter provides a comprehensive exploration of the BLOWFISH algorithm, its key features, modes of operation, and its significance in modern cryptographic systems [61].

In the early 1990s, there was a growing need for a replacement for the aging Data Encryption Standard (DES) algorithm. Bruce Schneier developed BLOWFISH as a freely available alternative, intending it to be fast, secure, and adaptable. BLOWFISH gained popularity for its simplicity and excellent performance in both software and hardware implementations.

The BLOWFISH algorithm operates on 64-bit blocks and utilizes a variable-length key. It consists of two phases: key expansion and data encryption. During the key expansion phase, the original key is processed to generate a set of subkeys, known as P-boxes and S-boxes. These subkeys are used in the data encryption phase, where the plaintext is transformed into ciphertext.

BLOWFISH can be used with various modes of operation to meet different encryption requirements. The Electronic Codebook (ECB) mode, Cipher Block Chaining (CBC) mode, and Cipher Feedback (CFB) mode are commonly used with BLOWFISH. These modes provide different levels of security and can be selected based on the specific application and security needs.

BLOWFISH is designed to be highly secure and resistant to known cryptanalytic attacks. Its key length flexibility allows users to choose a key size suitable for their security requirements. BLOWFISH's use of the Feistel network structure, combined with its key-dependent S-boxes, provides confusion and diffusion, enhancing its security. However, it is important to note that BLOWFISH has not undergone the same level of scrutiny as some other encryption algorithms, such as AES.

One of the main advantages of BLOWFISH is its speed and efficiency. It is particularly well-suited for software implementations and environments with limited computational resources. BLOWFISH's simplicity and straightforward design contribute to its fast encryption and decryption operations, making it a popular choice for applications that require high-performance encryption [62].

BLOWFISH has been widely used in various applications that require secure communication and data protection [63]. It has found applications in network security, file and disk encryption, password storage, virtual private networks (VPNs), and secure communications protocols. BLOWFISH's speed, flexibility, and strong security make it a valuable tool in these domains.

While BLOWFISH has its strengths, it is essential to consider its security and performance in comparison to other encryption algorithms. AES, for example, has become the industry standard and offers a higher level of security due to extensive scrutiny and analysis. However, BLOWFISH may still be a suitable choice for applications that prioritize speed, simplicity, and flexibility.

The BLOWFISH algorithm has provided a valuable alternative to the Data Encryption Standard (DES) algorithm, offering speed, simplicity, and flexibility. It has found

extensive applications in various domains that require secure communication and data protection. This chapter has provided an in-depth analysis of the BLOWFISH algorithm, including its key features, modes of operation, strengths, security considerations, performance, and applications. In the following chapters, we will explore other advanced encryption algorithms and their practical implementations.

3.5 The MARS Algorithm

The MARS (Matrix-ARithmetic-based-Substitution) algorithm is a symmetric encryption algorithm designed by Don Coppersmith and Matthew Lepinski. It was selected as a finalist in the Advanced Encryption Standard (AES) competition organized by the National Institute of Standards and Technology (NIST) in 1999. This chapter provides a comprehensive exploration of the MARS algorithm, its key features, operation, and its significance in modern cryptographic systems[64].

The MARS algorithm was developed as a candidate for the AES competition, which sought a successor to the aging Data Encryption Standard (DES) algorithm. While MARS did not ultimately become the selected AES algorithm, it gained recognition for its innovative approach and strong security features. MARS was designed to provide a high level of security while remaining efficient and suitable for various applications.

MARS operates on fixed-size blocks of 128 bits and supports key sizes ranging from 128 to 448 bits. It employs a combination of substitution, permutation, and arithmetic operations to achieve strong security. The algorithm utilizes a matrix-based design, with the encryption process involving multiple rounds of operations, including key expansion, substitution, permutation, and mixing [65].

Similar to other block ciphers, MARS can be used with various modes of operation to accommodate different encryption requirements. The most commonly used modes include Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB). These modes offer different levels of security and are selected based on the specific application and security needs.

MARS was designed to provide a high level of security against various attacks. It offers resistance against known cryptanalytic techniques, such as differential and linear attacks. The algorithm's reliance on matrix operations and its complex design contribute to its strength. However, it is important to note that MARS has not undergone the same level of scrutiny and analysis as some other encryption algorithms like AES.

MARS strives to strike a balance between security and performance. While it may not match the efficiency of some other encryption algorithms like AES, MARS is designed to be computationally efficient. The algorithm's matrix-based design allows for parallel processing, enhancing its performance in hardware and software implementations.

MARS has found applications in various domains that require secure communication and data protection. Its strong security features make it suitable for applications where data confidentiality and integrity are critical, such as secure communications, storage systems, and financial transactions. MARS can be used in both software and hardware implementations, making it versatile for different environments.

While MARS offers strong security features, it is important to consider its comparison to other encryption algorithms. AES, which became the selected AES algorithm, has gained widespread adoption and undergone extensive analysis. AES offers a higher level of confidence and security due to its rigorous evaluation process. However, MARS may still be a viable choice for applications that require specific security features or have compatibility requirements with legacy systems.

The MARS algorithm is a robust symmetric encryption algorithm that offers strong security features and computational efficiency. While it did not become the selected AES algorithm, MARS has gained recognition for its innovative design and approach to encryption. This chapter has provided an in-depth analysis of the MARS algorithm, including its key features, operation, strengths, performance considerations, applications, and comparison to other algorithms. In the following chapters, we will explore additional advanced encryption algorithms and their practical implementations.

In order to accomplish the goals that have been outlined, the research work will be

segmented into three primary components.

3.6 Classification of the Symmetric Encryption Algorithms

When attempting to categorize encryption algorithms, it is necessary to take into account a number of parameters that have an effect on the algorithm's level of complexity. In this stage of the work, a quantitative analysis of symmetric encryption algorithms will be carried out to gain a deeper understanding of their characteristics and performance. Among the algorithms falling into this category are DES, 3DES, Blowfish, MARS, and AES, which serve as notable examples. In today's cryptographic landscape, DES, 3DES, Blowfish, MARS, and AES remain significant algorithms, each with its unique attributes and applications. DES and 3DES, although no longer suitable for modern security standards, still find use in legacy systems. Blowfish is valued for its simplicity and efficiency in various network security and encryption applications. MARS played a crucial role as a candidate in the AES selection process, contributing to the evolution of modern encryption standards. AES, the Advanced Encryption Standard, stands as the preeminent encryption algorithm, widely accepted, highly secure, and adaptable for various industries, cementing its position as the most common and useful encryption algorithm today. These algorithms and their variations are the most common symmetric encryption algorithms today.[66].

The comparative analysis will encompass several criteria, including flexibility, reliability, scalability, and entropy. These factors are crucial in assessing the overall effectiveness and suitability of each algorithm for different applications. By evaluating these aspects, we can determine the strengths and weaknesses of the algorithms and make informed decisions about their implementation.

In addition to the comprehensive comparative analysis of the aforementioned criteria, we will delve into the cryptographic strength of the algorithms by employing a brute force attack. This approach involves systematically attempting all possible combinations of the algorithm's key until the correct one is found. By subjecting the algorithms to such

attacks, we can measure their resilience against brute force techniques and evaluate their robustness in real-world scenarios.

To gather the necessary data for this analysis, we will extensively study the technical documentation related to each algorithm. This documentation will provide insights into the design principles, key management, and security mechanisms employed by the algorithms. Furthermore, we will leverage a variety of computer programs specifically designed for analyzing encryption algorithms to aid in the selection of an appropriate key for each algorithm.

In addition to the brute force attack, our investigation will also encompass the exploration of other vulnerabilities, such as the Side-Channel Attack, Dictionary attack, and Cryptanalysis [67, 68]. These techniques, described in Table 1, represent alternative avenues through which an attacker may attempt to exploit weaknesses in encryption algorithms. By thoroughly examining these vulnerabilities, we can gain a comprehensive understanding of the potential risks associated with each algorithm and identify any potential mitigations that need to be implemented.

Overall, this comprehensive analysis of symmetric encryption algorithms aims to provide a thorough evaluation of their performance and security characteristics. By considering various parameters, conducting brute force attacks, and investigating other vulnerabilities, we can make informed decisions regarding the selection and implementation of encryption algorithms in different contexts.

Algorithm	Structure	Initial Vector Size	Flexibility and Modification	Known Attacks
DES	Feistel network	64 bits	The structure of DES does not support any modifications	Brute-Force Attack
3DES	Feistel network	64 bits	Could be extended from 56 up to 168 bits	Brute-Force Attack, Chosen Plaintext, Known Plaintext
AES	Substitution & Permutation	128 bits	Could be modified with a condition: 256 key length in multiples of 64	Side Channel Attack
BLOWFISH	Feistel network	64 bits	Could be modified with a condition: 64-448 key length in multiples of 32	Dictionary Attack
MARS	Feistel network	128 bits	Key size can range from 128 to 448 bits (in 32-bit increments)	Brute-Force Attack

Table 1: Comparative study of symmetric cryptographic algorithms

These data will be used to categorize algorithms according to the importance of the tasks they perform. On the other hand, a comparative analysis of asymmetric algorithms will also be carried out, in addition to the analysis of symmetric algorithms.

OFFICE FOR RESEARCH TRAINING, QUALITY AND INTEGRITY

DECLARATION OF CO-AUTHORSHIP AND CO-CONTRIBUTION: PAPERS INCORPORATED IN THESIS

This declaration is to be completed for each conjointly authored publication and placed at the beginning of the thesis chapter in which the publication appears.

1. PUBLICATION DETAILS (to be completed by the candidate)

Title of
Paper/Journal/Book:

Comparative study of symmetric cryptographic algorithms

Surname: Nurgaliyev

First name: Alibek

Institute: Institute for Sustainable Industries and Liveat

Candidate's Contribution (%): 70%

Status:

Accepted and in press:

☐

Date:

Published:

☒

Date:

31.10.2021

2. CANDIDATE DECLARATION

I declare that the publication above meets the requirements to be included in the thesis as outlined in the HDR Policy and related Procedures – policy.vu.edu.au.

[Redacted Signature]

Signature

13/07/2023

Date

3. CO-AUTHOR(S) DECLARATION

In the case of the above publication, the following authors contributed to the work as follows:

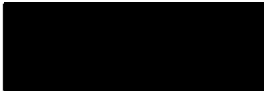
The undersigned certify that:

1. They meet criteria for authorship in that they have participated in the conception, execution or interpretation of at least that part of the publication in their field of expertise;
2. They take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;



3. There are no other authors of the publication according to these criteria;
4. Potential conflicts of interest have been disclosed to a) granting bodies, b) the editor or publisher of journals or other publications, and c) the head of the responsible academic unit; and
5. The original data will be held for at least five years from the date indicated below and is stored at the following **location(s)**:

Not applicable because I'm using public data.

Name(s) of Co-Author(s)	Contribution (%)	Nature of Contribution	Signature	Date
Professor Hua Wang	30%	Supervision and editing		13/07/23

Updated: September 2019

4 Preventing Informational Leaks: strategies and tools

In order to safeguard both personal and corporate information, it is crucial to implement various preventive measures to address any potential vulnerabilities, whether they stem from internal or external sources. These measures encompass restricting access to data, establishing non-disclosure agreements (NDAs), deploying data loss prevention (DLP) systems, and employing encryption techniques to protect sensitive data, among others. To achieve this objective, a methodology called meta-analysis will be employed, allowing for an investigation of findings from previous studies that have examined specific instances of information disclosure. This approach enables the formulation of general principles for safeguarding sensitive data by synthesizing information obtained from diverse incidents involving data leaks.

However, it is important to be aware of certain limitations associated with this research method. Firstly, the sheer volume of data necessitates a considerable amount of time to collect and analyze. Secondly, as the field of cryptography is continuously evolving, many of the analysis results may already be outdated, emphasizing the need to reassess the analysis using contemporary sources of information.

Despite these challenges, meta-analysis provides a valuable tool for comprehensively assessing and consolidating knowledge on information disclosure incidents. By synthesizing data from multiple studies, researchers can derive more robust and generalizable insights into effective data protection strategies. This allows for the identification of common vulnerabilities and the development of proactive measures to mitigate potential risks.

In light of the dynamic nature of information security, it is essential to continuously update and refine research findings [69, 70, 71]. This ensures that the conclusions drawn from the meta-analysis remain relevant and applicable to contemporary cryptographic practices. By combining the strengths of meta-analysis with ongoing research efforts, we can enhance our understanding of information protection and contribute to the development of robust and adaptable security measures.

Meta-analysis is a statistical technique used in research to combine and analyze the

results of multiple studies on a specific topic. It's a powerful method for synthesizing data and drawing more robust conclusions from a collection of individual studies. Here's how meta-analysis is typically used:

- **Research Question Formulation:** The process begins with the formulation of a research question or hypothesis. The question should be specific and well-defined, and it should relate to a topic for which there are multiple individual studies available.
- **Literature Search:** Researchers conduct a comprehensive literature search to identify all relevant studies on the chosen topic. This typically involves searching academic databases, journals, and other sources for published research.
- **Inclusion and Exclusion Criteria:** Studies that meet predefined inclusion criteria are selected for inclusion in the meta-analysis. These criteria might include study design, sample size, publication date, and other factors that are relevant to the research question.
- **Data Extraction:** Data from the selected studies are extracted. This includes relevant outcome measures, effect sizes, and other statistical information that will be used in the analysis.
- **Effect Size Calculation:** Effect sizes are calculated for each individual study. The choice of effect size depends on the nature of the data, but it is typically a measure of the magnitude of the effect being studied (e.g., the standardized mean difference, odds ratio, correlation coefficient, etc.).
- **Statistical Analysis:** Meta-analysis involves the application of statistical techniques to pool the effect sizes from the selected studies. Common statistical methods include calculating weighted averages, conducting subgroup analyses, and assessing heterogeneity (variation) among the results of individual studies.

- **Publication Bias Analysis:** Researchers often assess for publication bias, which is the tendency for studies with significant results to be published while studies with non-significant results are less likely to be published. This can be done using funnel plots and statistical tests.
- **Interpretation and Conclusion:**** After analyzing the combined data, researchers draw conclusions regarding the overall effect size and its significance. They may also investigate sources of variation, such as differences in study populations or methodologies.
- **Reporting:** The results of the meta-analysis are typically reported in a research paper or report. The report should include details about the search strategy, inclusion criteria, effect size calculations, statistical methods, and conclusions.
- **Implications:** The findings of the meta-analysis are used to make evidence-based recommendations or inform policy decisions. Depending on the field, these recommendations may be used in clinical practice, public health, education, or other areas.

Meta-analysis is widely used in various fields, including medicine, psychology, education, and social sciences, to summarize and synthesize research findings. It can provide a more comprehensive and reliable overview of a specific topic than individual studies and is a valuable tool for evidence-based decision-making.

In the subsequent sections, we will delve into the intricacies of meta-analysis and explore its application in the context of information disclosure incidents. Additionally, we will discuss the latest advancements in cryptography and the implications they hold for data protection strategies. Through a comprehensive analysis of existing literature and current developments, we aim to provide valuable insights into the protection of sensitive information in today's rapidly evolving digital landscape [72].

4.1 Reverse engineering. Common description

During our childhoods, many of us have attempted to take apart a toy in order to better comprehend how it is constructed on the inside. Some people maintained this pattern throughout their lives, applying their natural curiosity in the line of work they were already in. Continuing with the previous illustration, programmers will attempt to "disassemble" the structure of the program in order to either improve it or correct errors within it [73, 74, 75].

The process of analyzing an application in order to determine its functional characteristics, internal architecture, and, in fact, its operation — including its modules, functions, and algorithms — is referred to as "reverse engineering." IT professionals use reverse engineering for a variety of reasons, including the following: improving the functionality of an application in situations in which the company that developed it no longer exists or in which it is impossible to contact the company; analyzing viruses, worms, and Trojans in order to extract their signatures and create protection tools (anti-virus software); decrypting file formats in order to improve compatibility (file formats of paid popular applications for Windows that do not have Linux counterparts, such as Open Office or Gimp); learning the code and more [76, 77, 78].

Reverse engineering is important for several reasons:

- Understanding how a product works: Reverse engineering allows you to take apart a product and understand how it works. This can be helpful for troubleshooting, maintenance, and repair.
- Improving a product: By analyzing a product's features and specifications through reverse engineering, you can identify areas for improvement and develop new ideas for enhancing the product.
- Recovering lost or damaged data: Reverse engineering can be used to recover lost or damaged data from a product, which can be critical for businesses that rely on that data for their operations.

- Creating interoperability: Reverse engineering can help create interoperability between different systems, making it possible to use products that were not originally designed to work together.
- Protecting intellectual property: Reverse engineering can be used to identify potential security vulnerabilities in a product and develop measures to protect against intellectual property theft and cyberattacks [76].

Overall, reverse engineering is a valuable tool for improving products, enhancing security, and developing new ideas for innovation.

However, reverse engineering is frequently used "for other purposes." After all, once you have studied the architecture of the application or received the source code, you are able to make changes to it and use it for your own "selfish" purposes. Reverse engineering is frequently used "for other purposes." Here are some examples:

- Endless use of app trials. Suppose we have access to a product that we can try out for free for the next 30 days. When the application is started up, the installation date is compared to the present day's date in order to determine if it was successful. The application will continue to operate in trial mode indefinitely if this check is disabled or if it is replaced with a function that will consistently produce the desired output.
- The stealing of data or computer code. It's possible that an attacker won't go after the application itself, but rather a module or component of it. This strategy is applicable to companies that compete with one another in the software development industry.
- Get around the technical safeguards for intellectual property. The objective of the hacker is to circumvent the copy protection on audio and video files, computer games, and electronic books in order to make them freely available afterwards.

Applications designed for "desktops" as well as those designed specifically for mobile devices are both susceptible to being attacked. Because hacking methods depend more on

the programming language and the protection mechanisms that are actually implemented, it makes no difference at all in the context of reverse engineering whether the application is written to work on a smartphone or on a personal computer, because reverse engineering does not even exist. When all is said and done, a mobile application is nothing more than an archive that contains compiled code files, configuration files, and library files. This is the case regardless of how you look at it. As a consequence of this, the strategies utilized to "hack" mobile and desktop applications will, in general, be identical.

Due to the fact that obtaining the source code is a reverse compilation process, the procedure varies depending on the platform and programming language used. Take, for instance, the applications that were developed in the .Net framework are initially compiled into the Common Intermediate Language (CIL), and then the Common Language Runtime (CLR) transforms them into their respective native code. The compilation process for Java and Python applications is very similar: first, high-level code is converted into a low-level intermediate language known as bytecode, and then a just-in-time compiler converts the bytecode into machine code.

This company not only offers support for multiple operating systems but also makes it possible to compose individual components of an application in a variety of languages while retaining a consistent framework. In contrast to this, information regarding classes, structures, interfaces, and other such things can be obtained from an intermediate language from the perspective of reverse engineering (both CIL and bytecode). as well as restoring the building to its original design. There are utilities that are pre-made for this purpose, such as. For .Net applications, you can use Net Reflector, MSIL Disassembler, ILSpy, and dotPeek; for restoring Java from bytecode, you can use Javap, JAD, and DJ; and for working with Python applications, you can use pyREtic, pycdc, and Uncompyle2.

If an attacker has sufficient knowledge of CIL or bytecode, then sooner or later he will be able to modify it, recompile it, and make it work for his own purposes. This will allow the attacker to circumvent any security measures that have been put in place [79].

The process of reverse engineering applications written in traditional programming

languages (like C, C++, or Objective-C) is more difficult. Applications written in them are immediately compiled into executable machine code, which does not store any information about the structure of the original application. This includes information such as class names, names of functions or variables, and so on. The fact that the low-level representation does not contain branching constructs (such as if, for, and so on) presents an additional challenge. In order to restore these constructs, it is necessary to construct a flow graph that contains program control constructs [80]. This consumes a sizeable chunk of one's available time. However, this does not in and of itself guarantee that the source code of applications is secure. If you have a solid understanding of Assembler and are skilled at programming, the task of restoring the source code (or creating an identical version in terms of functionality) will be a simple matter of time.

The question now is, how do you protect your application? Or at the very least make the task of an attacker more difficult? The following are some common methods:

- The process of making computer code more difficult to read and understand while preserving its original purpose is known as code obfuscation. The process of reverse engineering is made significantly more difficult by obfuscation because even if an attacker obtains the source code, it is very difficult to understand what it is that they are doing with it. Mutation is one of the forms of obfuscation that is known to be particularly effective. This indicates that the application is continuously modifying its source code while it is being run, which makes the process of reverse engineering very challenging. Nevertheless, there are issues to be concerned about here. Not only is the code that has been obfuscated "unreadable" for the attacker, but it is also "unreadable" for the developer [81]. Adding unnecessary code branches can also lead to a decrease in performance and even introduce errors in the program. Obfuscation, on the other hand, does not guarantee a high level of security in the event that an adversary obtains the source code; this is true even if the code is difficult to comprehend. This is perhaps the most significant disadvantage. After all, the objective in this scenario is a particular segment of the program's source code;

consequently, it is not essential to deconstruct the functioning of the application as a whole in order to eliminate, for example, copy protection or license verification [82].

- A confirmation that the code has not been altered is what is meant by an integrity check. In order to accomplish this, the checksums of various sections of the application code are calculated, and in the event that there is a discrepancy between the calculated value and the value that was specified, the application will cease functioning. However, this presents its own set of challenges, as an adversary who has gained access to an application's source code may be able to disable the integrity check or replace it with a function that always produces the desired outcome if they are in a position to modify the code.
- Encryption of the program code serves the purpose of ensuring that only "legal" buyers are able to use the application. This is accomplished by ensuring that, in the absence of the encryption key, the program either cannot be used at all or will only function on trial branches [83]. Despite this, the integrity of the code cannot be guaranteed in any way, shape, or form because it is theoretically possible for the mechanism that generates keys to be exposed.

There are numerous additional methods of protection, such as watermarks, separating critical sections of code into individual modules, protected execution environments, and so on; however, none of these can guarantee complete safety. The strategy that is utilized to safeguard an application needs to be adapted on a case-by-case basis.

For instance, code obfuscation is not only a feature that enhances security, but it can also improve performance in certain circumstances. Therefore, writing code in a single line or replacing variable names with ones that are shorter and less intuitive leads to a reduction in the size of the assembly, which in turn leads to an increase in the performance of the application. Obfuscations, on the other hand, like adding code branches or aliases, can make things move more slowly [84].

Therefore, when selecting methods to protect the code, the first thing you need to do is be guided by the threat model. Specifically, you need to ask yourself what needs to be protected within the application and what ways an attacker can try to get it. If this is a change to the code, then it is important to place an emphasis on the integrity check. If you are analyzing a portion of the application, then it is important to consider whether obfuscation or encryption would be beneficial. If you use the protection methods that have been described, you can make the task as difficult as it possibly can be for an attacker, despite the fact that there is no solution that is guaranteed to work.

4.2 Practicing reverse engineering

Reverse engineering serves not only as a means to comprehend existing applications but also as a way to customize them according to specific requirements. Android apps, with their versatility, provide an excellent platform to illustrate the process of reverse engineering. This chapter aims to delve deeper into the process by examining several different applications. As an initial example, let's consider an application designed for learning Morse code and explore the modification of its resource files, including elements such as colors, titles, and captions.

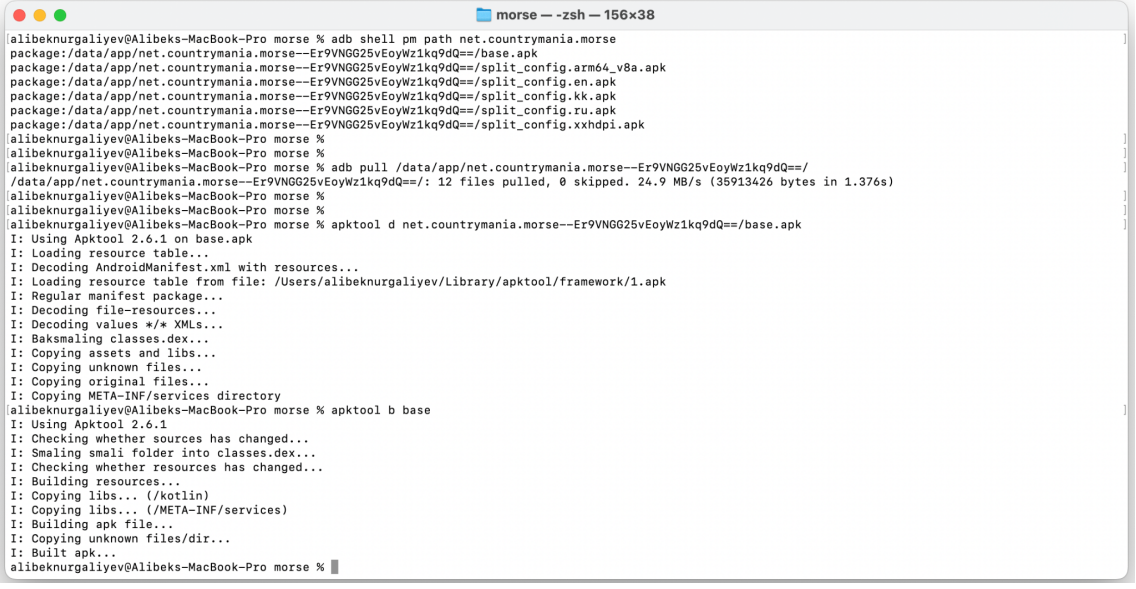
To begin, a crucial skill to acquire is the ability to disassemble and reassemble an application. For this purpose, we can utilize the apktool utility, which facilitates the extraction of all resources embedded within the application. This utility unpacks and repacks APK (Android Package Kit) files, which store Android application codes in a binary format and compressed state. Additionally, it disassembles the program code contained within these files.

The APK file format represents a complete archive of Android application codes. These components are combined into a single installation file, commonly referred to as the APK. Typically, only a single APK file is necessary for the successful installation of a fully functional application [85, 86].

Despite its name, the APK format is not limited to the Android platform alone. Applications available for Windows, Mac OS X, and Linux operating systems allow users to open and extract individual code files from APK archives.

In order to obtain the installation package, one can utilize the Android Debugging Bridge (ADB), which serves as a system for debugging programs on devices. On Linux-based operating systems, ADB can be automatically installed through a package manager. On Windows, it is included as part of Android Studio or the Android SDK Platform Tools [87].

The process involves downloading the desired app from the Google Play Store onto a mobile device, establishing a connection between the device and a computer using a USB cable, and using the Android Debugging Bridge to transfer the application package to the computer and extract its contents. To achieve this, a series of commands should be executed in the following order:



```
morse -- zsh -- 156x38
alibeknurgaliyev@Alibeks-MacBook-Pro morse % adb shell pm path net.countrymania.morse
package:/data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/base.apk
package:/data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/split_config.arm64_v8a.apk
package:/data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/split_config.en.apk
package:/data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/split_config.kk.apk
package:/data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/split_config.ru.apk
package:/data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/split_config.xxhdpi.apk
alibeknurgaliyev@Alibeks-MacBook-Pro morse %
alibeknurgaliyev@Alibeks-MacBook-Pro morse %
alibeknurgaliyev@Alibeks-MacBook-Pro morse % adb pull /data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/
/data/app/net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/: 12 files pulled, 0 skipped. 24.9 MB/s (35913426 bytes in 1.376s)
alibeknurgaliyev@Alibeks-MacBook-Pro morse %
alibeknurgaliyev@Alibeks-MacBook-Pro morse %
alibeknurgaliyev@Alibeks-MacBook-Pro morse % apktool d net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ==/base.apk
I: Using Apktool 2.6.1 on base.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/alibeknurgaliyev/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory
alibeknurgaliyev@Alibeks-MacBook-Pro morse % apktool b base
I: Using Apktool 2.6.1
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/kotlin)
I: Copying libs... (/META-INF/services)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
alibeknurgaliyev@Alibeks-MacBook-Pro morse %
```

Figure 1: android decompile commands

adb shell pm path net.countrymania.morse - using this command, you will be able to locate the path to the APK file that was installed. There have been multiple files sent to us with the.apk extension. These are configuration files that contain text resources for three

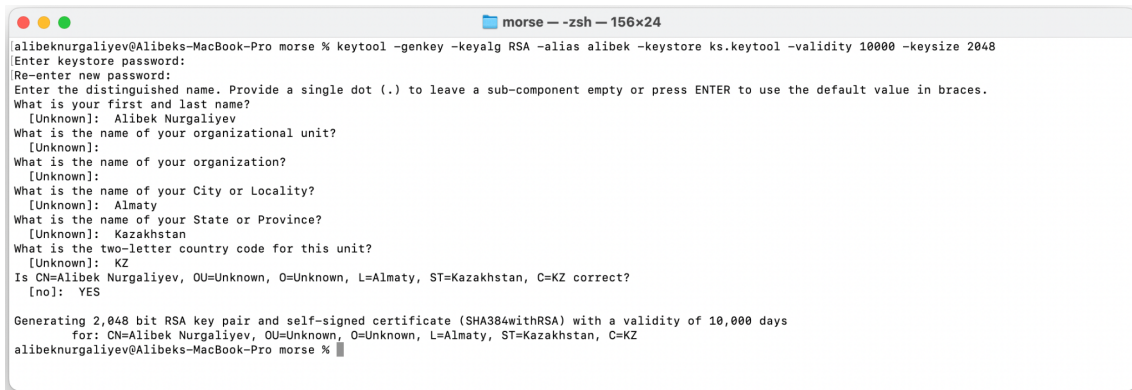
different languages: Kazakh, Russian, and English [88]. In addition to text resources, there is a configuration file that establishes the dimensions of the display screen on the mobile device and, for those smartphones that have a higher pixel density, offers images and pictures of a higher quality. There is also a file called `base.apk`, which is directly the installation file of the application and which we require because it contains the source code. You can find this file in the same location as the application. On the other hand, it is essential to copy all of the apk files because. When it comes time to install the application on a mobile device, we will require their assistance.

`adb pull /data/app/net.countrymania.morse-Er9VNGG25vEoyWzlkq9dQ==/` - this command downloads all of the apk files in the directory to our personal computer.

`apktool d net.countrymania.morse-Er9VNGG25vEoyWzlkq9dQ==/base.apk` - we can unzip the apk file into a folder with the same name by using the apktool utility.

`apktool b base` - the following command creates an apk file that contains all of the files currently contained within the base folder.

At this point in the process, we have taken apart the previously installed mobile application and put it back together again. Within the newly created dist folder, we can locate a new .apk file to work with. I gave it the name `new_morse.apk` and renamed it. However, until we sign the apk file with an encryption key, the Android security policy will not permit us to install the file. We are going to build our own cryptographic key, and then use it to sign the application that we have put together. In order to accomplish this, we will make use of the utility known as `keytool`, which is included with the Java Development Kit.

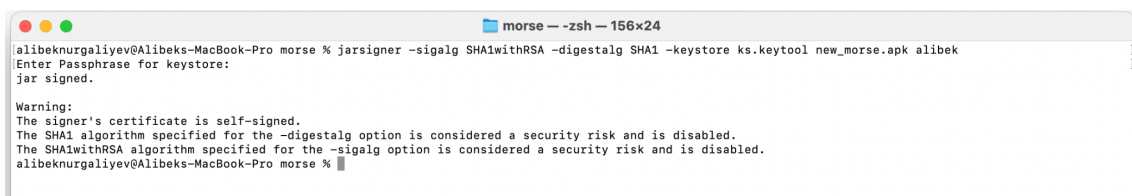


```
morse -- -zsh -- 156x24
alibeknurgaliyev@Alibeks-MacBook-Pro morse % keytool -genkey -keyalg RSA -alias alibek -keystore ks.keytool -validity 10000 -keysize 2048
Enter keystore password:
Re-enter new password:
Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default value in braces.
What is your first and last name?
[Unknown]: Alibek Nurgaliyev
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]: Almaty
What is the name of your State or Province?
[Unknown]: Kazakhstan
What is the two-letter country code for this unit?
[Unknown]: KZ
Is CN=Alibek Nurgaliyev, OU=Unknown, O=Unknown, L=Almaty, ST=Kazakhstan, C=KZ correct?
[no]: YES

Generating 2,048 bit RSA key pair and self-signed certificate (SHA384withRSA) with a validity of 10,000 days
for: CN=Alibek Nurgaliyev, OU=Unknown, O=Unknown, L=Almaty, ST=Kazakhstan, C=KZ
alibeknurgaliyev@Alibeks-MacBook-Pro morse %
```

Figure 2: creating signature

Now that our key has been generated, all that is left to do is use the jarsigner utility, which is included as another component of the Java Development Kit, to sign the compiled application as well as the configuration files [83]. However, before you can do that, you will need to open the application configuration files and delete the /META-INF folder along with all of its contents. This is necessary because the folder contains data pertaining to the signature of the previous certificate. It is imperative that this folder be deleted so that there are no collisions with the cryptographic key that we use. Using any archiver, you should be able to open the.apk file.



```
morse -- -zsh -- 156x24
alibeknurgaliyev@Alibeks-MacBook-Pro morse % jarsigner -sigalg SHA1withRSA -digestalg SHA1 -keystore ks.keytool new_morse.apk alibek
Enter Passphrase for keystore:
jar signed.

Warning:
The signer's certificate is self-signed.
The SHA1 algorithm specified for the -digestalg option is considered a security risk and is disabled.
The SHA1withRSA algorithm specified for the -sigalg option is considered a security risk and is disabled.
alibeknurgaliyev@Alibeks-MacBook-Pro morse %
```

Figure 3: apk file signing process

It is necessary for us to repeat the same process with each configuration file. Following this step, you will need to uninstall the application that we downloaded from the Play Market and place it on your phone, followed by the installation of the application that we compiled and signed with a cryptographic key ourselves. You can accomplish this by utilizing the command to install multiple apk files all at once.

```
net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ== -- mc - zsh - 202x16
alibeknurgaliyev@Alibeks-MacBook-Pro net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ== % adb install-multiple new_morse.apk split_config.arm64_v8a.apk split_config.en.apk split_config.kk.apk split_config.ru.apk split_config.xxhdp1.apk
Success
alibeknurgaliyev@Alibeks-MacBook-Pro net.countrymania.morse--Er9VNGG25vEoyWz1kq9dQ== %
```

Figure 4: android install apk commands

Our application, which is self-signed, installed without any problems, and we were able to run it. We were successful in disassembling the application, and then reassembling it with our own key before signing it.

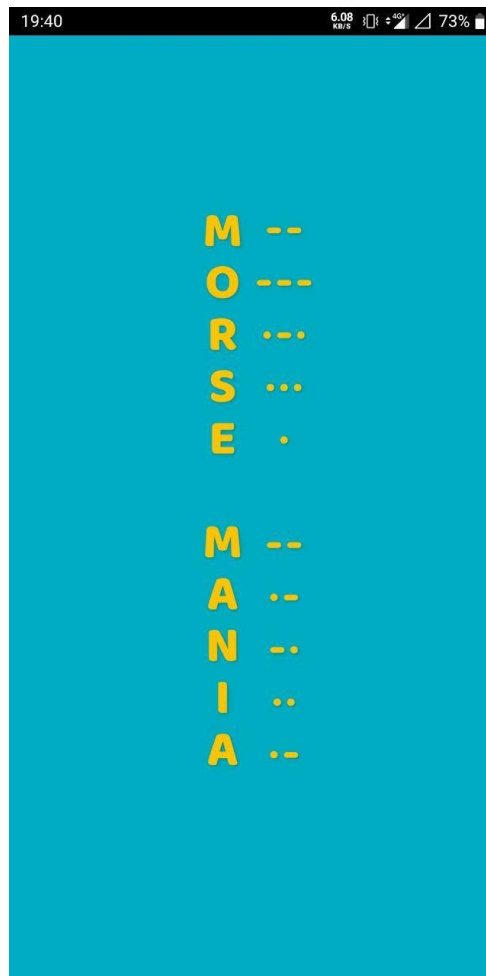


Figure 5: initial screen of the application

The modifications to the mobile application's resources will take place in the following step. This includes things like titles and color schemes, among other things.

The application allows for a number of different color customization options, including the following:

- Make use of the predefined colors, such as `@android:color/white`;
- create a palette of user-defined colors in the `colors.xml` file, and then use those colors as the `@color` and text primary values;
- You can set the color with a hexadecimal code, but the most important thing is to remember that the colors in the app are written in the format `#AARRGGBB`, and transparency comes first;
- When writing code in Java or Kotlin, you should make use of the `setTextColor` and `setBackgroundColor` parameters.

All of these strategies are common knowledge and are routinely implemented.

4.2.1 `colors.xml` file

The color scheme for the application can be found in the file that is located at `morse/base/res/values/colors.xml`.

This is how the structure of the file looks:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="abc_decor_view_status_guard">#ff000000</color>
    <color name="abc_decor_view_status_guard_light">#ffffffff</color>
    ....
    <color name="tooltip_background_dark">#e6616161</color>
    <color name="tooltip_background_light">#e6ffffff</color>
</resources>
```

This is where the information for each color is stored, which will later be used in the application's static screens. You can get a general idea of which screen the color is used

for as well as which component it belongs to by looking at the names of the colors. The following format should be used when writing about colors:

A hex color code is a six-character string that can be broken down into as many as three parts that are each two symbols long. Each of the elements that consists of two symbols can express a color value that ranges from 0 to 255:

Element 1: Red value

Element 2: Green value

Element 3: Blue value

A formula is used to write the code, and it is this formula's job to convert each value into a distinct two-digit alphanumeric code. As an illustration, the hexadecimal representation of the RGB code (0, 172, 193) is 00ACC1.

For instance, all we did was change the color of the application's start screen, which previously was cyan but is now blue, see Figure 6.

4.2.2 styles.xml file

The file `morse/base/res/values/styles.xml` contains some color specifications, but not all of them. When developers of an application want to create multiple color schemes for that application, they use this file. In the event that these color schemes are not available, the process of changing colors will take significantly more time.

Because the colors of some text and the background are already set, any modifications to them must be made in the same manner as in the `colors.xml` file.

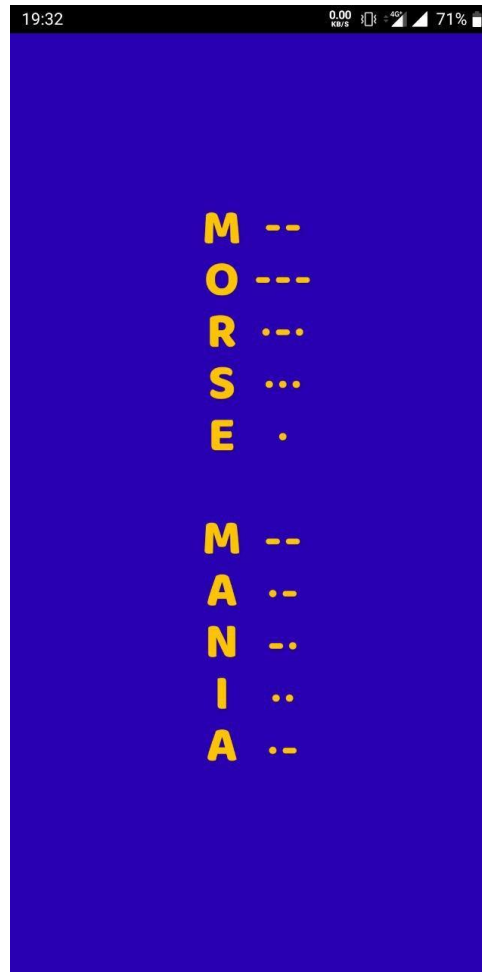


Figure 6: modified screen of the application

3) Layout files

The descriptions of the application screens can be found within the `morse/base/res/` folder.

- The universal screen storage for all mobile devices is located at `morse/base/res/layout/`. These resources will be utilized, unless another course of action is specified;
- When working with the most advanced UI elements, the `morse/base/res/layout-v"XX` (where XX is the smartphone SDK version, which varies depending on the version of Android installed on the device) file format is required;
- The remainder of `morse/base/res/layout-...` define application-specific screens for things like device orientation and resolution.

You can find and change the colors of certain application elements, most frequently the background, in this section.

But we can't wrap this up just yet because there are some screens whose background and text colors aren't set in.xml files like the rest of the application; rather, they are set directly in the application's executable code. Only the application code itself allows us to make changes to them.

4.2.3 Smali files

Within Android applications, they use their very own file format called.dex, which stands for Dalvik EXecutable, as well as their very own virtual machine called Dalvik in order to run these files.

There is a bytecode for.dex, and it's called smali. It's human-readable and easy to understand at a glance, just like the bytecodes for other compiled languages.

In contrast to the JVM, the Dalvik machine relies on registers rather than stacks to store information. Registers do not have types, so they can store anything, including numbers, strings, and instances of classes. Simultaneously, the language of Swazi is highly characterized by its typology.

Use the smali decompiler in Java known as Jadx if you are having trouble understanding what the meaning of this or that register is or what the purpose of this or that function is. It is easy to use Jadx; all that is required is to install it from the open repository, enter the command into the jadx-gui terminal, select the file to open, and then click the Open button. The source code will then be displayed automatically after it has been decompiled.

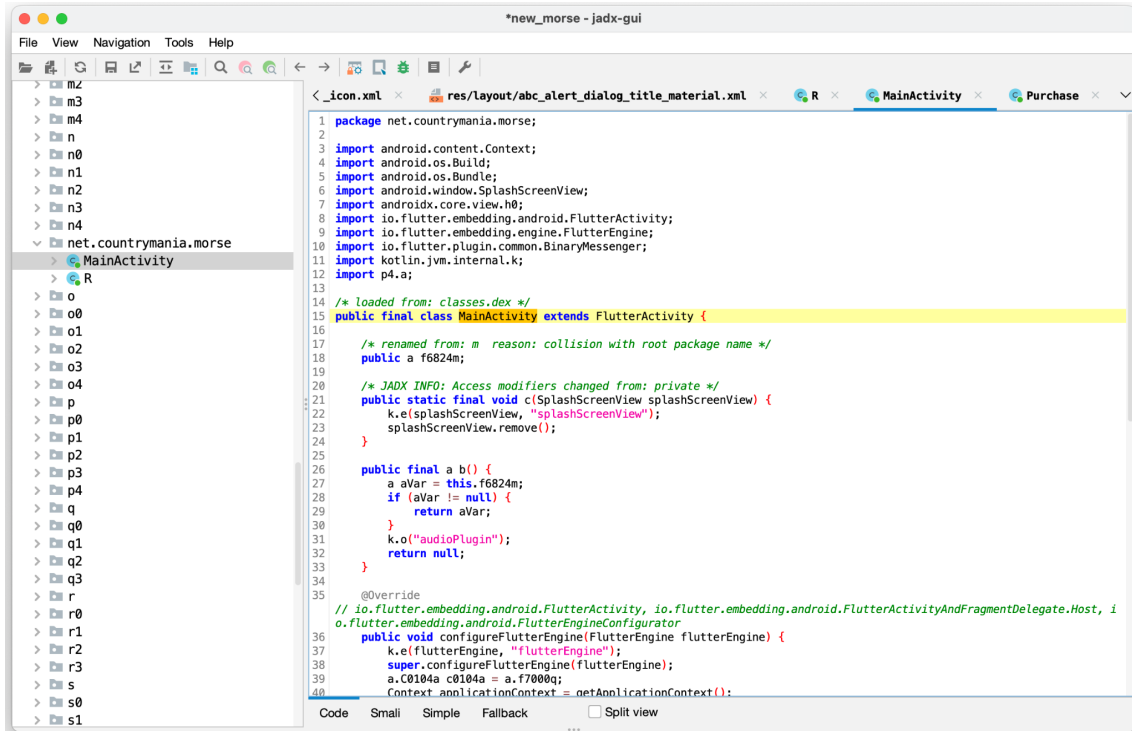


Figure 7: jadx-gui application

4.3 Obfuscation

Obfuscation of code refers to the process of purposefully concealing program code through the use of obfuscation while keeping the product's functionality intact [89].

Either the process is carried out manually, which takes a significant amount of time and makes it challenging to "deobfuscate" the information, or it is carried out automatically, which occurs much more quickly and is carried out by special programs called "obfuscators" that have a "deobfuscation" function. The work is carried out by the programmer with the intention that no other programmers will be able to read the computer code and interpret the obfuscator techniques [90].

To begin, the purpose of obfuscating the source code is to increase the software's overall level of safety [91]. The developer may also have the option of pursuing economic aims, such as protecting the competitive advantage against forgeries or concealing values and logic. The outcome is either software that has been compressed (unused classes,

attributes, and methods are deleted) or software that has been optimized (operators are checked and rebuilt). Obfuscated code is utilized on platforms such as Android and Java, for instance (optimizer example: R8 for Android; ProGuard for Java and Android) [92].

Evaluation of the practicability and efficiency of obfuscation is performed throughout the management of development [90]. To put it another way, what aspects should a management take into consideration when hiring a performer or, more generally, when assigning such a task?

- **Secretiveness**

Determine the level of secrecy that is maintained by the program control algorithms. For example, is it a good idea to do control-flow obfuscation (which is commonly used in iOS apps) when changing the control variable is monitored and the transition to the dispatcher node is replaced by transitions to the next block, which corresponds to the new value of the control variable?

- **Cost**

The obfuscation method has associated costs and potential profits. In order to ensure that the selected approach is suitable for extensive use in a variety of contexts that are analogous to one another, it is essential to conduct an analysis of whether or not the expenses are reasonable [93, 94].

- **Protection**

How much more difficult it is to read the modified code in comparison to the original version. Different kinds of security precautions are defined by the software complexity metrics. For instance, the total number of predicates it possesses, the depth of the hierarchy, the number of nesting levels, and many other such aspects. Obfuscation is done with the intention of making something as difficult as possible, while the purpose of excellent software design is to reduce complexity based on these characteristics as much as feasible [95].

- Stability

Finds out how effectively the converted code can defend itself against attacks from automatic deobfuscation tools. A change that can only occur in one direction and cannot be undone by the deobfuscator has the maximum degree of stability. Obfuscation, for instance, will get rid of information such as the formatting of the source code.

Obfuscation need to be a component of the development process whenever a programmer or a development business delivers useful closed-source software (like an app for iOS or Android), especially if the product is closed source. Because of this, it is more difficult for an outside party to crack, analyze, and debug the code [96].

- Code obfuscation examples

- 1) Rename Obfuscation

When you rename anything, the names of the associated methods and variables are changed. It makes the source code that has been decompiled more difficult for a human to understand, but it does not affect how the program is actually run. Different naming conventions, such as "a," "b," and "c," as well as numerals, characters that cannot be printed, and invisible characters, may be used for the new names. In addition, there is no limit to the number of names that may be used provided each one has a unique scope. The majority of obfuscators for.NET (C#, etc.), iOS, Java, and Android make advantage of a fundamental transformation known as name obfuscation.

Original Source Code Before Rename Obfuscation	Reverse-Engineered Source Code After Rename Obfuscation
<pre> private void CalculatePayroll (SpecialList employee- Group) { while (employeeGroup.HasMore()) { employee = employeeGroup.GetNext(true); employee.UpdateSalary(); Distribute Check(employee); } } </pre>	<pre> private void a(a b) { while (b.a()) { a = b.a(true); a.a(); a(a); } } </pre>

Figure 8: Rename Obfuscation

2) String Encryption

Even though String Encryption differs from regular obfuscation, it can still be considered a form of obfuscation. String encryption involves converting plaintext strings (such as text or data) into a format that is not easily human-readable or recognizable. Each and every string is easily discoverable and readable within an executable that has been controlled. Even when methods and variables are renamed, strings may be used to discover key code parts by searching for string references inside the binary. This is done by checking for string references within the binary. Messages that are presented to the user, particularly error messages, are included in this category. String encryption is used to conceal strings inside the executable and only returns them to their original value when it is absolutely necessary to do so. This helps to offer an effective defense against this kind of assault. In most cases, the decryption of strings at runtime results in a marginal decrease in performance during runtime.

Original Source Code Before String Encryption	Reverse-Engineered Source Code After String Encryption
<pre>... MessageBox.show("Invalid Authentication - Try Again") ...</pre>	<pre>... MessageBox.show(a.b("Σ;f5f5HJ•Q")) ...</pre>

Figure 9: String Encryption

3) Control Flow Obfuscation

When control flow is obfuscated, conditional, branching, and iterative constructs are synthesized. These constructs create legitimate executable logic, but when they are decompiled, they produce non-deterministic outputs for their semantics. To put it another way, it makes the decompiled code appear to be in the form of spaghetti logic, which is extremely difficult for a hacker to understand. The performance of a method at runtime might be impacted by these techniques.

Original Source Code Before Control Flow Obfuscation	Reverse-Engineered Source Code After Control Flow Obfuscation
<pre>public int CompareTo (Object o) { int n = occurrences - ((WordOccurrence)o).occurrences; if (n == 0) { n = String.Compare (word, ((WordOccurrence)o).word); } return (n); }</pre>	<pre>private virtual int _a(Object A+0) { int local0; int local1; local 10 = this.a - (c) A_0.a; if (local10 != 0) goto i0; while (true) { return local1; } i1: local10 = System.String.Compare(this.b, (c) A_0.b); goto i0; }</pre>

Figure 10: Control Flow Obfuscation

4) Insertion of the Dummy Code

Including code in the executable that, while it does not impact the logic of the program

itself, does either disrupt decompilers or make it far more difficult to evaluate code that has been reverse engineered.

5) Negative aspects of obfuscation

Unfortunately, code obfuscation may also be used by attackers for malicious purposes [97], and the following are just a few examples of this:

- Obfuscation is a technique that attackers employ to disguise or incorporate harmful code into software products that appear to be secure. Antivirus software is designed to identify potentially malicious behaviors in running applications in advance.
- The execution speed of the program is decreased because, as the program's code grows more complicated, a single instruction is occasionally substituted by three to five other commands. Because of this, the total number of instructions included inside the program grows, necessitating an increase in the amount of time required for the computer to process these instructions.
- It is extremely difficult to modify obfuscated code without having access to the original source code. If the programmer deletes the source code of the program after it has been converted by accident, it will be far simpler for him to rewrite the program than it will be for him to attempt to figure out the result.

In summary, while obfuscation and symmetric encryption can mitigate information leaks to a significant extent, a holistic security approach may involve additional techniques like data integrity checks, access controls, intrusion detection systems, and regular security audits. Obfuscation and symmetric encryption are valuable tools in the quest to prevent information leaks, they are key methods used to prevent leaks and protect data and intellectual property from unauthorized access, theft, or disclosure.

OFFICE FOR RESEARCH TRAINING, QUALITY AND INTEGRITY

DECLARATION OF CO-AUTHORSHIP AND CO-CONTRIBUTION: PAPERS INCORPORATED IN THESIS

This declaration is to be completed for each conjointly authored publication and placed at the beginning of the thesis chapter in which the publication appears.

1. PUBLICATION DETAILS (to be completed by the candidate)

Title of
Paper/Journal/Book:

Analysis of reverse engineering

Surname: Nurgaliyev

First name: Alibek

Institute: Institute for Sustainable Industries and Liveat

Candidate's Contribution (%): 70%

Status:

Accepted and in press:

☐

Date:

Published:

☒

Date:

07.05.2023

2. CANDIDATE DECLARATION

I declare that the publication above meets the requirements to be included in the thesis as outlined in the HDR Policy and related Procedures – policy.vu.edu.au.

	13/07/2023
---	------------

Signature

Date

3. CO-AUTHOR(S) DECLARATION

In the case of the above publication, the following authors contributed to the work as follows:


The undersigned certify that:

1. They meet criteria for authorship in that they have participated in the conception, execution or interpretation of at least that part of the publication in their field of expertise;
2. They take public responsibility for their part of the publication, except for the responsible author who accepts overall responsibility for the publication;



3. There are no other authors of the publication according to these criteria;
4. Potential conflicts of interest have been disclosed to a) granting bodies, b) the editor or publisher of journals or other publications, and c) the head of the responsible academic unit; and
5. The original data will be held for at least five years from the date indicated below and is stored at the following **location(s)**:

Not applicable because I'm using public data.

Name(s) of Co-Author(s)	Contribution (%)	Nature of Contribution	Signature	Date
Professor Hua Wang	30%	Supervision and editing		13/07/23

Updated: September 2019

5 Cybercrime Analysis

In order to gain a comprehensive understanding of how to effectively combat cybercrimes in the real world, it is crucial to investigate and analyze real-world examples and stories from a technical standpoint [98, 99, 100]. This section of the study aims to collect open-source data from various articles and papers that describe victims of cybercrimes, regardless of whether the crimes were the result of personal issues or flaws in a company's policy [101].

The collected data will be classified as technical data and will encompass information such as the encryption algorithm used by the victim, the type of data that was compromised or lost, the duration of time that has passed since the incident, and the method employed to perpetrate the cybercrime. However, a complete analysis necessitates the inclusion of non-technical data as well. This includes details such as the year the crime was committed, a profile of the perpetrator, and information about the security policies in place within the affected company, if applicable. By gathering both technical and non-technical data, we can gain a multidimensional perspective of the cases, particularly in relation to the handling of sensitive data across various domains. The comparative analysis of real-world data will play a significant role in determining the most effective strategies for reducing the prevalence of cybercrimes. While laboratory experiments offer higher internal validity, their external validity and practical applicability may be limited. Consequently, data obtained solely from laboratory experiments would not possess the same level of validity or practical utility. Therefore, to provide valuable insights and support efforts aimed at mitigating the impact of cybercrimes, the following section will offer recommendations for governments, social agencies, educational institutions, and researchers.

By synthesizing the real-world data and drawing upon its insights, stakeholders can develop strategies and policies that are informed by actual cybercrime incidents. This holistic approach will contribute to the collective effort in combatting cybercrimes and fostering a safer digital landscape [102].

5.1 History of cybercrimes

The development of the personal computer has simplified the lives of people; people at all levels, from individuals to major corporations, are now using computers for a wide variety of tasks all over the world. A computer can be defined, in its most basic form, as a machine that is able to store, manipulate, and process information or instructions based on the user's instructions. Since the 1980s, the vast majority of people who use computers do so for inappropriate reasons, whether it be for their own advantage or for the advantage of others. This gave rise to the concept of "Cyber Crime." This had resulted in the participation in actions that are considered unlawful by the society. We are able to define "Cyber Crime" since it refers to crimes that are perpetrated using computers or networks and typically take place in cyberspace, particularly on the web [103].

In 1995, Sussman and Heuston were the ones who came up with the term "Cyber Crime" for the first time. The term "cybercrime" cannot be reduced to a single, all-encompassing meaning; rather, it is more accurately understood as an umbrella term encompassing a variety of different activities or behaviors [104, 105]. These examples are dependent on the physical offender item that influences the information or frameworks that are stored in the computer. These are the unethical demonstrations in which a technologically advanced device or data architecture is either an apparatus or an objective, or it tends to be a mixture of both. Other names for the phenomenon that we refer to as cybercrime include electronic crimes, computer-related crimes, ecrime, high innovation crime, data age crime, and others [106].

To put it another way, we can say that "Cyber Crime" refers to any infractions or crimes that are committed through the use of electronic communications or data systems. These kinds of criminal activities are, essentially, criminal operations that involve the use of a computer and/or a network in some way. Because to the development of the internet, there has also been an increase in the volume of criminal activities that take place online. This is occurring for the reason that it is now possible to commit crimes without the real presence of the offender being required. The peculiar aspect of cybercrime is

that the individual in issue, as well as the perpetrator of the crime, may never come into direct contact with one another. In order to reduce the likelihood of being discovered and charged with a crime, cybercriminals frequently choose to operate their businesses from countries that either do not have any laws pertaining to cybercrime or have laws that are not very effective. People have the misconception that cybercrimes can only be conducted while using cyberspace or the internet, however this is not the case. It is not necessary for the cybercriminal to remain present online in order to commit a cybercrime; in point of fact, cybercrimes can be performed even when the perpetrator is not actively involved in the cyberspace in which they occur. The privacy of software is one example that can be taken into consideration [107].

Within the span of time around the year 1820, the first instance of cybercrime was documented. The earliest known examples of computers were discovered in Japan, China, and India around the year 3500 B.C.; nevertheless, the development of modern computers is generally regarded to have begun with Charles Babbage's analytical engine. A French textile manufacturer named Joseph-Marie Jacquard invented the loom in the year 1820. Jacquard's invention took place in France. This gadget made it possible for there to be a sequence of steps that occurred continuously during the process of weaving specialized textiles or materials. As a consequence of this, the employees of Jacquard were filled with irrational fear that their means of subsistence as well as their traditional employment were in jeopardy. As a result, they preferred to engage in acts of sabotage in order to dissuade Jacquard and ensure that the company would not be able to make use of the innovative technology in the foreseeable future.

In May 2002, the first cybercrime conviction occurred in India. Sony India Private Limited, operating the website www.sony-sambandh.com, catered to Non-Resident Indians (NRIs) who could purchase Sony products online for delivery in India. An individual, using the alias Barbara Campa, ordered a color television and a cordless headphone, providing credit card details for payment and requesting delivery to Arif Azim in Noida. The transaction was approved, and after due diligence, the company delivered the products.

However, the true owner of the card later denied the purchase, prompting a complaint from the company. The Central Bureau of Investigation (CBI) investigated the matter under Sections 418, 419, and 420 of the Indian Penal Code. The probe revealed that Arif Azim, employed at a call center in Noida, had accessed an American national's credit card information and used it on the company's website. The CBI recovered the products and presented strong evidence, leading to Azim's confession and conviction under Indian Penal Code sections 418, 419, and 420. Given Azim's age and first-time offense, the court showed leniency, releasing him from probation after a year. Additionally, Sections 67 and 70 of the Information Technology Act applied, addressing cases where hackers deface websites with explicit or slanderous content. Cybercrimes are a global concern, affecting nations worldwide.

To understand why cyber-crime in Africa is different from that in other parts of the world, one must first have an understanding of the state of information security in Africa, which is affected by factors such as the growth of user base, poor security awareness, lack of training for law enforcement, lack of regulations, and weak cross-border collaboration. However, in order to understand why cyber-crime in Africa is different from that in other parts of the world, one must first have an understanding of the state of information security in Africa [103].

In recent years, there has been a spectacular increase throughout the number of people in Africa who have access to the internet. The number of people coming online for the first time in Africa is growing at a faster rate than in any other region of the world, thanks to the widespread availability of broadband connections and the declining cost of online subscriptions. According to Internet World Stats, the percentage of people using the internet across Africa reached 2.3% of the total worldwide population in December of 2007. 4 When compared to the growth of 180.3 percent seen throughout the rest of the world from 2000 to 2007, Africa's internet usage climbed by a whopping 423.9 percent. This high number of users in Africa has made the Internet a popular means of communication in addition to opening up new opportunities for online business. Unfortunately, this high

number of users has also led to an increase in cyber-criminal activities, which has necessitated an increased effort across the region to strengthen the information infrastructure, educate users in security awareness, and develop cybercrime regulations. In point of fact, cybercrimes are a global concern, and there is no nation that is immune to their threat.

To understand why cyber-crime in Africa is different from that in other parts of the world, one must first have an understanding of the state of information security in Africa, which is affected by factors such as the growth of user base, poor security awareness, lack of training for law enforcement, lack of regulations, and weak cross-border collaboration. However, in order to understand why cyber-crime in Africa is different from that in other parts of the world, one must first have an understanding of the state of information security in Africa. In recent years, there has been a spectacular increase throughout the number of people in Africa who have access to the internet. The number of people coming online for the first time in Africa is growing at a faster rate than in any other region of the world, thanks to the widespread availability of broadband connections and the declining cost of online subscriptions. According to Internet World Stats, the percentage of people using the internet across Africa reached 2.3% of the total worldwide population in December of 2007. When compared to the increase of 180.3 percent seen throughout the rest of the world from 2000 to 2007, Africa's internet usage climbed by 423.9 percent. This high number of users in Africa has made the Internet a popular means of communication in addition to opening up new opportunities for online business [108]. Unfortunately, this high number of users has also led to an increase in cyber-criminal activities, which has necessitated an increased effort across the region to strengthen the information infrastructure, educate users in security awareness, and develop cybercrime regulations. The potential for misusing the internet is significantly higher in Africa and for everyone there. Many people who utilize the internet are falling victim to cybercrime attacks, and the number of successful attacks is expanding unchecked since there are no security awareness programs or specialized training for law enforcement authorities. It is pointless to revisit the vast body of literature on the topic because so much has already been written about the

staggering occurrences of cyber-based fraud emerging from African states [109].

5.2 Analysis of cybercrimes

The term "cybercrime" refers to any crime in which a computer is either the target of the crime or the primary instrument utilized in the committing of the crime. A cybercriminal could utilize a device to gain access to private information belonging to a user, sensitive company information, or government information, or they could disable a device of their choosing. Selling or getting any of the aforementioned information through the use of the internet is also considered a cybercrime [110].

Two distinct types of illegal activity can be classified as cybercrime:

- Criminal activity directed at networks or devices: Viruses, phishing emails, and other forms of malware
- Cyberstalking, denial of service attacks (DOS), and identity theft committed online are examples of crimes committed with the use of electronic technology.

Different types of criminal activity on the cybercrime.

Individual offenses, property offenses, and government offenses are the three primary divisions of cybercrime. The approaches taken, as well as the degree of challenge involved, change depending on the category.

- Possessional: Comparable to a situation that occurred in real life, a thief may have illegally stolen an individual's banking or credit card information. This scenario is similar. A hacker will steal a person's financial information in order to gain access to finances, to make online transactions, or to perform phishing scams, which are con games that deceive individuals into giving the hacker their personal information. They might also make use of malware in order to access a website that contains sensitive information.

- **Individual:** The individual is the focus of this type of cybercrime, which involves the victim (a human) in order to transmit harmful or illegal material on the internet. It's possible that the victim is participating in illicit activities like cyberstalking, the distribution of pornography, or even human trafficking.
- **Governmental:** Cybercrime against governments is the most serious form of the crime, despite being the least common form of the crime. Cyber terrorism can also refer to criminal activity directed at government institutions. Hacking into the websites of government and military entities, as well as spreading propaganda, are both examples of government cybercrime. These kinds of offenders are typically members of terrorist organizations or high-ranking government officials from hostile countries.

5.3 Types of cyber crimes

1. DDoS Attacks

With the increasing reliance on technology and the internet, Distributed Denial of Service (DDoS) attacks have become an increasingly severe threat to businesses, organizations, and individuals. DDoS attacks occur when cybercriminals flood a target's network or website with so much traffic that it crashes, effectively denying access to legitimate users. This paper will explore DDoS attacks in more detail, looking at their origin, types, impacts, prevention, and mitigation strategies [111].

- **Origin of DDoS Attacks**

DDoS attacks are not a new phenomenon, as they have existed since the early days of the internet. However, the rise of the Internet of Things (IoT) and the proliferation of internet-connected devices such as cameras, appliances, and thermostats have made it easier for cybercriminals to carry out DDoS attacks. These devices, referred to as botnets, can be remotely controlled by cybercriminals to send massive amounts of traffic to a target's website or network [112].

- Types of DDoS Attacks

DDoS attacks come in various forms, each with its unique characteristics and modus operandi. Some common types of DDoS attacks include:

1. Application layer attacks – these attacks target the web applications of a target, overwhelming them with requests that overload the servers and cause them to crash. Common examples include HTTP floods and Slowloris attacks.
2. Network layer attacks – these attacks target the network infrastructure of a target, overwhelming it with traffic that causes a network outage. Common examples include UDP floods and ICMP floods.
3. Protocol attacks – these attacks exploit weaknesses in the communication protocols used between computers, such as TCP and DNS. They can cause service outages by preventing computers from communicating with one another.
4. Volumetric attacks – these attacks are the most common and involve overwhelming the target's network with a flood of malicious traffic, thereby rendering it inaccessible. Common examples include amplification attacks and botnets.

- Impacts of DDoS attacks

The impacts of a DDoS attack can be devastating, both financially and reputationally, as they can cause significant business disruption and downtime. Some of the impacts include:

1. Loss of revenue – DDoS attacks can render a website or online service inaccessible, resulting in the loss of revenue. A business that relies on e-commerce could potentially lose thousands of dollars a minute.
2. Damage to reputation- a DDoS attack could damage an organization's online reputation, causing customers to lose confidence in their ability to provide secure and reliable services.
3. Data breaches- DDoS attacks can be used as a smokescreen to distract IT teams

while cybercriminals launch more sophisticated attacks such as injecting malware or stealing sensitive data.

4. Legal repercussions - Businesses may also face legal repercussions if data breaches occur due to DDoS attacks.

- Prevention of DDoS attacks

Preventing DDoS attacks requires a proactive approach that involves a combination of technical and non-technical solutions. Some of the prevention strategies include:

1. Network segmentation- Segmenting networks can help contain the impact of a DDoS attack by dividing it into smaller parts, reducing the overall damage caused by an attack.

2. Up-to-date software - Organizations should ensure that all software used in their networks is up-to-date with the latest patches and security updates. This helps to prevent known vulnerabilities that attackers could exploit.

3. Cloud-based DDoS protection – Cloud-based DDoS protection services can detect and mitigate any DDoS attack by monitoring network traffic and identifying malicious traffic in real-time.

4. Buffered capacity – Organizations should ensure that they have enough bandwidth and network capacity to handle unexpected surges in traffic.

5. Intrusion detection and prevention systems - By deploying such systems, it will be possible to keep track of the network's traffic and identify any unusual patterns that could signal a potential DDoS attack.

- Mitigating the Effects of DDoS attacks

Despite implementing prevention strategies, DDoS attacks may still occur. Therefore, it's vital to have effective mitigation strategies in place to mitigate the effects of an attack. Some of the mitigation strategies include:

1. Diverting traffic to Anycast IPs - Diverting traffic to anycast IPs can help absorb and deflect attack traffic without impacting the performance of legitimate traffic.
2. Blackholing - A network administrator can use blackholing to stop the flow of traffic from the DDoS attacker's IP address by directing all traffic to the firewall or blackhole router.
3. Using content distribution networks (CDNs) – CDNs are a network of proxy servers in different geographic locations that can help distribute traffic and reduce the impact of a DDoS attack.

DDoS attacks have been around for years, and as technology advances, so do the sophistication of these attacks. Businesses, organizations, and individuals who rely on their web presence to conduct business are especially prone to the impacts of DDoS attacks.

2. Phishing

With an exponential increase in the number of internet users, cyber threats have become a major concern for individuals, businesses and governments across the globe. One of the most prevalent types of cyber attacks is phishing, which has become a major tool for cyber criminals to steal sensitive information from individuals and organizations. Phishing is a type of social engineering attack where an attacker impersonates a trustworthy entity in order to obtain sensitive information from the target [113]. This research paper will explore the various aspects of phishing, including its types, techniques, impacts, and prevention measures.

- Types of Phishing

Phishing attacks can be classified into different types, each with their own unique characteristics. The following are the most common types of phishing attacks.

1. Email Phishing

The most common type of phishing attack is email phishing. This is where the

attacker sends an email to the victim that appears to be from a legitimate source such as a bank, social media platform or a website login page. The email usually contains a link that takes the victim to a fake website where they are asked to enter their login credentials. Once the victim enters their credentials, the attacker can use them to steal sensitive data.

2. Spear Phishing

Spear phishing is a more targeted version of email phishing. In this type of attack, the attacker targets a specific individual or organization. The attacker will research the target before creating a tailored email or message that appears to be from a trusted source. The goal of spear phishing is to steal sensitive data such as financial information or trade secrets.

3. Clone Phishing

Clone phishing is a type of attack where the attacker creates a replica of a legitimate email. The attacker will copy the content of a real email and replace the links or attachments with malicious ones. The victim will receive the email, which appears to be from a legitimate source, and open the attachment or click on the link. This allows the attacker to install malware on the victim's system or steal sensitive data.

4. Whaling

Whaling is a type of phishing attack that targets high-profile individuals such as executives or celebrities. The attacker will use social engineering to gain the target's trust before requesting sensitive information such as financial data or login credentials.

5. Phishing Techniques

Phishing attacks use a variety of techniques to trick the victim into providing sensitive information. The following are some of the most common techniques used in phishing attacks.

6. Social Engineering

Phishing attacks use social engineering tactics to manipulate the victim into believing that the attacker is a trustable entity. The attacker will use psychological pressure such as urgency or fear to make the victim act quickly. For instance, the attacker may threaten to close the victim's bank account if they don't verify their login credentials.

7. Malware

Phishing attacks often use malware to infect the victim's system. The attacker will send a file or attachment that contains a virus or worm that can steal sensitive data or take control of the victim's system. The malware may also be used to launch further attacks on the victim's network.

Spear phishing attacks may leverage personal or business details about the victim to appear more legitimate. For example, the attacker may send an email that appears to be from the victim's boss, containing specific details and requests.

- Impact of Phishing Attacks

Phishing attacks can have severe consequences for both individuals and organizations. The following are some of the impacts of phishing attacks.

1. Financial Losses

Phishing attacks can cause significant financial losses for individuals and organizations. Attackers can use stolen login credentials to access bank accounts or credit card information, leading to theft or financial loss.

2. Data Breaches

Phishing attacks can lead to data breaches, where attackers gain access to sensitive information such as customer data, trade secrets, or proprietary information. This can severely damage an organization's reputation and result in legal implications.

3. System Damage

Malware installed through phishing attacks can cause damage to the victim's system, leading to the loss or corruption of important data. This can result in downtime, operational costs, and a negative impact on business continuity.

- Prevention Measures

There are various prevention measures that individuals and organizations can take to avoid phishing attacks. The following are some of the prevention measures.

1. Employee Training

Educating employees on how to detect and report phishing attacks is an effective way to prevent attacks. Employees should be aware of the various types of phishing attacks and what to do if they suspect an attack.

2. Multi-Factor Authentication

Using multi-factor authentication, such as SMS or app-based codes, can provide an additional layer of security to prevent stolen credentials from being used. This may deter attackers who find it more difficult to penetrate secure systems.

3. Email Filters

Email filters can be set up to block suspicious emails from reaching the user's inbox. This can include blocking emails from known phishing domains and flagging those with suspicious attachments or links.

Phishing attacks have become increasingly sophisticated and pose a significant threat to individuals and organizations. Cybercriminals use a variety of techniques to trick victims into revealing sensitive information. However, there are various prevention measures that can be implemented to reduce the risk of falling victim to phishing attacks, including employee training and multi-factor authentication [101].

3. Botnets

Botnets are a network of computers infected with malicious software, also known as malware, that allow cybercriminals to remotely control them for illegal activities such

as theft, spamming, distributed denial of service (DDoS) attacks, and online fraud. The word "botnets" comes from the combination of the words "robot" and "networks" explaining how the system works. Cybercriminals use the compromised computers to form a network that allows them to launch attacks on websites, commit financial fraud, and steal sensitive information. The use of botnets has increased over the years, becoming a significant threat to online security.

Botnets are a significant and growing problem for online security. As the number and complexity of botnets have increased, the negative impact of these networks on the internet has grown. According to a report by the European Union Agency for Cybersecurity (ENISA), botnets are responsible for approximately 20% of all online crime. Additionally, cybersecurity experts predict that botnets will continue to grow and become more sophisticated, making it more challenging to detect and stop them.

Botnets typically consist of two components: a command-and-control (C&C) server and a network of infected computers known as bots. The C&C server is a computer that the cybercriminals use to control the botnet. It sends commands to the bots that are infected with malware installed on the target machine. The bots then receive the command and execute it, allowing the cybercriminal to control the computer.

There are various types of botnets, each with its unique characteristics. Zombie botnets are the most common type of botnets, and they are used in DDoS attacks. Cybercriminals use zombies to flood the target website with traffic, making it temporarily unavailable to users. Spam botnets, on the other hand, are used to send spam emails to a large number of email addresses. These emails often contain phishing scams, malicious attachments, and links that can install malware on the recipient's computer.

Botnets can be used for a wide range of illegal activities. They are often used to gain unauthorized access to a victim's computer, steal sensitive information such as credit card details, login credentials, and create unauthorized financial transactions. Botnets are also used in DDoS attacks to make a website unavailable by overwhelming it with traffic, which can lead to significant financial losses for businesses.

The methods used to infect computers with malware vary widely. The most common methods include phishing emails, downloading infected software, or clicking on a malicious link on a website. Cybercriminals often use social engineering tactics to trick users into downloading malicious software or visiting compromised websites. Once the malware is installed on a computer, it connects to the C&C server, which allows the cybercriminals to control the botnet and execute their illegal activities.

Detecting and mitigating botnets can be challenging because they are designed to operate covertly. Cybercriminals often use techniques such as encryption, multi-step command structures, and decentralized communication to prevent detection by antivirus software and other security measures.

Several techniques can be used to detect and mitigate botnets. One approach is to use signature-based detection, which matches a specific pattern of code to identify malware. However, this technique is limited to known threats and cannot detect new and emerging botnets.

Another technique is to use anomaly detection. This approach uses machine learning algorithms to identify unusual network traffic patterns that may indicate botnet activity. It is based on the fact that botnets tend to have a distinct pattern of network traffic that differs from normal network traffic.

To mitigate the effect of botnets, organizations can implement preventative measures such as keeping software up to date, using strong passwords, and using antivirus software. Companies can also use network traffic analysis and behavioral analysis to detect botnet activity immediately and take action to stop it.

In summary, botnets are a significant threat to online security, with cybercriminals using them to launch DDoS attacks, steal sensitive information, and commit financial fraud. Detecting and mitigating botnets can be challenging due to the methods used by cybercriminals to remain undetected. However, several techniques can be used to identify and prevent botnets, including signature-based detection, anomaly detection, and network traffic analysis. Implementing preventative measures can also help reduce the risk of bot-

net infection. It is essential for businesses and individuals to remain vigilant and take proactive measures to ensure their online security.

4. Identity theft committed online

Identity theft is a type of cybercrime that involves stealing someone's personal or financial information and using it for fraudulent purposes. Online identity theft is a growing problem, with millions of people falling victim to this crime every year.

Identity theft committed through online cybercrime involves criminals using the internet to gain access to personal information such as social security numbers, bank account numbers, credit card details, and other sensitive information. These details can then be used to make unauthorized purchases, open new credit accounts, apply for loans, and commit other frauds in the victim's name.

One of the most common methods used in online identity theft is phishing. Phishing is the fraudulent practice of sending email messages or creating fake websites that appear to be legitimate in order to steal sensitive information such as passwords, credit card numbers, and social security numbers. The emails or websites may contain links that, when clicked, install malware on the victim's device. This malware can then be used to steal the victim's information or monitor their online activities.

Another method of online identity theft is known as hacking. Hacking involves gaining unauthorized access to a computer, smartphone, or other device with the intention of stealing personal or financial information. Hackers can use a variety of techniques, including exploiting vulnerabilities in software or services, brute force attacks, or social engineering techniques that exploit human weaknesses.

Online identity theft is a serious problem that can have serious consequences for victims. Not only can it result in financial losses, but it can also damage a person's credit score and compromise their personal and professional reputation. Victims of online identity theft may also experience emotional distress and loss of trust in online services, which can have lasting effects.

In order to protect themselves against online identity theft, individuals should take a

variety of precautions. These include using strong passwords, regularly updating software and security features, avoiding clicking on suspicious links or downloading unknown attachments, and regularly monitoring their accounts for unauthorized activity.

Financial institutions and other businesses also have a role to play in protecting their customers from online identity theft. They should implement strong security measures, including two-factor authentication, encryption, and fraud detection systems. They should also educate their customers about ways to protect themselves from online identity theft.

Due to the growing threat of online identity theft, governments and law enforcement agencies around the world have also taken steps to tackle this issue [114]. Many countries have laws in place that criminalize identity theft and provide for stiff penalties for offenders. In addition, international collaboration between law enforcement agencies has increased in recent years, making it easier to track and prosecute cybercriminals who commit online identity theft [115].

In conclusion, identity theft committed through online cybercrime is a growing problem that affects millions of people each year. This crime involves stealing sensitive personal and financial information using a variety of techniques such as phishing and hacking. Individuals and businesses can take steps to protect themselves from online identity theft, while governments and law enforcement agencies can provide stronger laws and increased international collaboration to tackle this issue. By working together, we can help prevent and mitigate the effects of online identity theft.

5. Online scam

The internet age has brought about a lot of changes to the way people conduct business. Today, people buy and sell all manner of goods and services online. The ease and convenience of this mode of transaction have made it popular worldwide. However, this has also opened new opportunities for cybercriminals. Online scams have become a significant problem over the years, leading to significant financial losses to individuals, companies, and governments. This chapter provides an in-depth analysis of online scams.

An online scam is a type of cybercrime that involves the use of the internet or other

electronic channels to conduct fraudulent activities. It involves the use of fraudulent techniques to deceive and exploit vulnerable individuals or groups. Online scams take different forms, from phishing to identity theft.

1. Phishing Scams

Phishing scams involve the use of deception to gain access to people's personal and sensitive information. It involves sending fraudulent emails, text messages, or social media messages in the guise of a legitimate institution, such as a bank or a government agency. The intent is to trick the recipient into clicking on a link or downloading a file, leading to the installation of malware or the provision of sensitive information.

2. Identity Theft

Identity theft is a type of online scam that involves stealing someone's identity. It is a form of cybercrime that uses personal information such as name, date of birth, social security number, or credit card information to impersonate the victim. The perpetrator can then use this information to open bank accounts, make online purchases or obtain credit.

3. Online Shopping Scams

Online shopping scams are a form of online scam that involves the use of fraudulent online stores to deceive unsuspecting buyers. These stores offer goods and services at a bargain but fail to deliver once payment is made. In other words, the seller disappears, leaving the buyer stranded.

4. Investment Scams

Investment scams are a type of online scam that involves convincing unsuspecting individuals to invest in fraudulent schemes. These scams promise high returns on investments but are designed to defraud the investors.

5. Job Scams

Job scams target job seekers by posting attractive job offers on popular job boards.

These offers promise high-paying jobs but require individuals to pay a fee upfront or provide sensitive information, such as social security numbers or bank account details.

6. Social Media Scams

Social media scams involve the use of social media platforms to exploit unsuspecting individuals through different means such as click-baiting, spam accounts, or phishing links.

7. Effects of Online Scams

The effects of online scams are widespread and can be both personal and financial. The personal effects of online scams include loss of trust, emotional distress, and embarrassment. Victims of online scams may also be wary of conducting online transactions in the future. Financial effects include monetary losses due to the loss of personal or company funds. It can also lead to a decline in business reputation due to customers losing trust.

Online scams have become a significant problem in today's digital age. They have affected countless individuals and businesses, causing significant financial losses. It is thus important for individuals and businesses to take measures to prevent online scams. This can be achieved by being vigilant and adopting secure practices when conducting online transactions. It is also important to report such activities to appropriate authorities to help prevent future cases of online scams.

6. Social engineering

Social engineering cybercrime refers to a type of cybercrime that involves the manipulation of individuals to divulge confidential information or grant unauthorized access to computer systems. Cybercriminals employ various techniques to deceive unsuspecting individuals into revealing sensitive information such as passwords, personal identification numbers, and credit card details. Social engineering has become a significant threat to

cybersecurity and has led to significant financial losses for individuals, businesses, and governments. This chapter provides an overview of social engineering cybercrime, the techniques employed by cybercriminals, and the impact on individuals and organizations [116].

Social engineering cybercrime involves the use of psychological manipulation to deceive individuals into taking actions that aid cybercriminals in their nefarious activities. Cybercriminals typically use different strategies to deceive individuals, such as phishing scams, pretexting, baiting, and scareware attacks. Phishing scams involve the sending of an email or text message that appears to be from a legitimate source, such as a bank or social media platform, asking individuals to provide sensitive information [117]. Pretexting involves the use of false identities to gain the trust of individuals and extract confidential information. Baiting is a technique that involves offering something desirable, such as a free gift or prize, to lure individuals into providing sensitive information. Scareware attacks involve the use of malicious software to scare individuals into providing sensitive information.

Cybercriminals continue to refine their techniques to make them more effective and harder to detect. For instance, spear-phishing targets a specific individual or organization to gather sensitive information or gain unauthorized access. Cybercriminals gather personal information from social media platforms and other online sources to personalize their attacks to increase their chances of success. The use of AI and machine learning to automate social engineering attacks is also increasing, making it more challenging to detect and prevent such attacks.

Social engineering cybercrime has significant implications for individuals and organizations. Individuals who fall victim to social engineering attacks may suffer financial losses, identity theft, and psychological trauma. Organizations that fall victim to social engineering attacks may suffer significant financial losses, reputational damage, and legal liabilities. The theft of intellectual property and business secrets poses a significant risk to businesses, leading to loss of market share, reduced competitiveness, and damage to

innovation capabilities. Governments and critical infrastructure sectors are also vulnerable to social engineering cybercrime, which can potentially lead to significant disruption of essential services, intellectual property theft, and serious national security threats.

The prevention and mitigation of social engineering cybercrime require a collaborative approach among individuals, organizations, and governments. Individuals need to be aware of the risks of social engineering cybercrime and adopt preventive measures, such as updating their antivirus software, using strong and unique passwords, and being skeptical of unsolicited emails and text messages. Organizations need to implement robust cybersecurity measures that include employee training, regular software updates, and the adoption of multi-factor authentication systems. Governments need to invest in cybersecurity research and development, implement stringent regulations to protect individuals and organizations from cybercrime, and promote international cooperation to combat cybercrime.

In conclusion, social engineering cybercrime continues to pose a significant threat to individuals, organizations, and governments. Cybercriminals employ a wide range of techniques to deceive individuals into divulging sensitive information or granting unauthorized access to computer systems. The prevention and mitigation of social engineering cybercrime require a collaborative approach among individuals, organizations, and governments. The adoption of preventive measures, robust cybersecurity measures, and stringent regulations is essential in combating this pervasive threat to cybersecurity. It is vital to recognize the severity of social engineering cybercrime and take proactive steps to ensure that individuals, organizations, and governments are prepared to face this challenge.

7. Cyberstalking

Cyberstalking is becoming increasingly common. It is the act of threatening, harassing, or intimidating an individual using the internet and other digital communication methods. This type of harassment can have a significant impact on an individual's mental health, quality of life, and overall well-being. In this chapter, we will discuss the prevalence and

characteristics of cyberstalking, its effects on victims, and prevention and intervention strategies to confront this type of behavior.

- Prevalence and Characteristics

Cyberstalking is more common than we might think. According to a study conducted by the Pew Research Center, 15% of individuals who use the internet have been the victim of cyberstalking. This type of harassment often takes place through social media, text messages, email, and online forums. Cyberstalkers may use tactics such as threatening messages, posting personal information online, or secretly monitoring the victim's online activity.

Cyberstalkers can be anyone – strangers, acquaintances, or former partners. They may be motivated by a desire for power, revenge, or control. In some cases, cyberstalkers may also engage in physical stalking, which can make the victim feel even more threatened.

- Effects on Victims

Cyberstalking can have serious consequences for the victim's mental health and overall well-being. Victims may experience anxiety, depression, and post-traumatic stress disorder (PTSD). They may also feel a sense of helplessness, loss of privacy, and fear of leaving their home.

Furthermore, cyberstalking can affect the victim's employment and personal relationships. For example, if the perpetrator spreads false rumors about the victim on social media, it can damage their reputation and harm their job prospects. Additionally, if the victim's friends and family are contacted by the perpetrator, it can strain those relationships and create tension within the victim's support network.

- Prevention and Intervention Strategies

There are several prevention and intervention strategies that can be used to confront cyberstalking. First, it is important to educate individuals about the risks and dangers of online harassment. This can involve teaching young people about safe and

responsible social media use and encouraging them to speak up if they experience cyberstalking or any other kind of online abuse.

Second, social media platforms and other online communities can implement measures to prevent and address cyberstalking. For example, social media sites can offer reporting tools that allow users to report threatening or harassing messages. These platforms can also work with law enforcement to identify and prosecute cyberstalkers.

Third, it is essential to provide support and resources for victims of cyberstalking. This can involve connecting victims with mental health professionals, legal resources, and community support groups. It is also helpful to provide victims with information about self-defense and safety measures they can take to protect themselves.

In conclusion, cyberstalking is a serious and growing problem that can have devastating consequences for victims. It is important to recognize the prevalence and characteristics of cyberstalking, its effects on victims, and prevention and intervention strategies to confront this type of behavior. By educating individuals, implementing proactive measures to prevent cyberstalking, and providing support for victims, we can work to create a safer and more secure online environment for everyone.

8. PUP (potentially unwanted programs)

The rise of technology has opened up new possibilities for businesses, governments, and individuals to connect and conduct their activities online. However, technological advancements have also created new forms of criminal activity, including cybercrime. One such form of cybercrime is phishing, pharming, and malware attacks on PUP (potentially unwanted programs) targets. PUP includes browser hijackers, fake antivirus software, adware, etc. This chapter aims to discuss the main types of PUP cybercrime, their impact on society, and the measures that can be taken to prevent or reduce their occurrence.

- Types of PUP Cybercrime

1. Phishing Attacks

Phishing attacks are the most common form of cybercrime that target PUP users. Hackers send fake emails or messages that appear to come from trusted sources such as banks, e-commerce websites, or government agencies. These messages typically have a sense of urgency and prompt the user to click on a link or download an attachment. Once the user responds, they are redirected to a fake website created by the hacker to steal their personal information, including usernames and passwords to access crucial accounts such as bank accounts [118].

2. Malware Attacks

Malware attacks are another prevalent form of cybercrime targeting PUP users. Malware refers to malicious software created by hackers to access or damage a user's computer system and steal sensitive information. Malware can be introduced through emails, pop-up ads, or downloaded software from phishing websites or even social media. The impact of malware attacks can range from loss of data to total loss of control over the systems targeted.

3. Adware Attacks

Adware is a type of software that displays unwanted advertisements on the user's computer without their consent. Adware is usually downloaded as part of a larger program without the user's knowledge. These programs are designed to track the user's browsing habits and display ads that match their interests. The harm here is that users' private data can be collected by these adware applications while they are connected to the internet.

4. Browser Hijackers

Browser hijackers are programs that redirect users' web searches to unwanted websites or change their homepages. These programs can be introduced through bundled software downloads or downloaded from unsecured websites. The impact of browser hijackers is that they can redirect users to potentially harmful websites or

collect user data to be used for cybercriminal activities.

- **Impact of PUP Cybercrime on Society**

The impact of PUP cybercrime varies depending on the actions of the criminal actors. However, some of the common impacts include:

1. **Loss of Data:** PUP cybercrime can lead to a loss of personal and confidential information, including identity theft and financial loss.
2. **Financial Damage:** Prior to PUP cyberattacks, users can likely face financial losses, negative experiences with the services or products, etc.
3. **Reputational Damage:** When an organization experiences a PUP cybercrime incident, it could damage its reputation.
4. **Economic Impacts:** PUP cybercrime can result in significant economic loss, particularly since many PUP criminals look to extort ransoms from businesses and individuals.

- **Measures to Prevent and Reduce PUP Cybercrime**

The growing cases of PUP cybercrime have led to increased efforts to prevent and reduce these attacks. These methods include:

1. **Education:** Users must be educated on how to recognize PUP cybercrime and avoid it. Educating users on the importance of strong passwords, two-factor authentication, and the risks associated with downloading software from untrusted sources can be crucial to reducing PUP cybercrime.
2. **Security Measures:** Anti-virus software can be used to detect and prevent PUP cybercrime from infiltrating a computer system. These software applications can provide real-time protection against phishing, malware, and adware attacks.
3. **Regular Updating:** Regularly updating software can help prevent PUP cybercrime since criminals often try to exploit vulnerabilities in outdated software.

In conclusion, PUP cybercrime is a significant threat to businesses, individuals, and governments. Organizations and users must take steps to prevent or reduce the occurrence of these attacks. Educating users, implementing security measures, and regularly updating software are some of the ways to minimize the risks associated with PUP cybercrime. It is crucial that organizations and individuals remain vigilant in protecting themselves against these cybercrimes to avoid financial losses, reputational damage, and loss of confidential information.

9. Prohibited/Illegal Content

The internet has brought manifold benefits to society; it connects people, provides access to information, and facilitates worldwide communication. However, with increased access to the web, there has been an explosive growth in cybercrime. Cybercrime refers to any criminal activity that is committed through the use of computer systems. One of the most significant types of cybercrime is the dissemination of prohibited or illegal content.

Prohibited or illegal content includes materials that promote or facilitate criminal behaviors like human trafficking, drug trafficking, terrorism, or child pornography. The dissemination of these materials is a significant problem because it can have a damaging impact on society, especially on minors who may be exposed to the content. Therefore, this research paper will discuss prohibited/illegal content cybercrime, its effects, and the measures that can be taken to prevent it.

- Effects of Prohibited/Illegal Content Cybercrime

Prohibited/illegal content is a significant concern due to its detrimental effects on society, particularly young people. Children and teenagers are often the most vulnerable targets for cybercriminals disseminating this content. The dissemination of illicit content can lead to psychological and emotional trauma for the youth population. For instance, exposure to violent materials may lead to aggressive behavior while exposure to sexual material can lead to the sexualization of minors. In addition, this type of content can foster poor mental and emotional health, contribute to

higher rates of depression, and lower self-esteem.

Additionally, the dissemination of prohibited/illegal content can have an adverse impact on the community as a whole. For instance, it promotes behaviors that contravene societal norms and moral values. The distribution of illicit content can result in a higher rate of drug addiction, sexual promiscuity, human trafficking, and terrorism activities. Moreover, the existence of such content can foster the growth of criminal organizations involved in illegal activities, including money laundering and other financial crimes.

- **Measures to Prevent Prohibited/Illegal Content Cybercrime**

There are various measures that can be taken to prevent or reduce prohibited/illegal content cybercrime. A critical step is to enact laws that prohibit the dissemination of illicit content and impose strict penalties for its distribution. In most jurisdictions, the dissemination of child pornography and terrorist material is strictly prohibited, and those found guilty are punished severely. These laws have proven to be effective in curbing the spread of prohibited/illegal content.

Secondly, it is imperative to enforce these laws to ensure that those who violate them are brought to justice. Law enforcement agencies should be adequately trained to investigate and prosecute cybercriminals involved in the dissemination of illicit content. This includes tracking down individuals or groups responsible for the distribution and taking action to prevent them from disseminating any further.

Thirdly, internet service providers (ISPs) and social media platforms can play a critical role in preventing prohibited/illegal content cybercrime. They should establish policies and guidelines to counter the spread of illicit content. This includes implementing content-filtering mechanisms that prevent users from uploading or sharing prohibited content. Meanwhile, social media platforms can use technologies such as automated content monitoring and reporting systems that instantly flag any prohibited/illegal content.

Fourthly, parents, teachers, and guardians have a significant role to play in protecting minors from prohibited/illegal content. They should educate their children and young people about the dangers of illicit content. Additionally, they should install parental controls and monitoring software to track their children's online activities and prevent them from accessing prohibited content. Moreover, they should promote responsible online behavior among young people and create a safe internet environment for them.

Lastly, international collaboration is essential in preventing prohibited/illegal content cybercrime. Cybercrime is a global phenomenon that traverses borders and jurisdictions. Therefore, collaboration between nations is crucial for effective law enforcement and the exchange of intelligence and information concerning prohibited/illegal content.

In conclusion, prohibited/illegal content cybercrime is a significant challenge for society, particularly for children and young people. It has far-reaching and detrimental effects on individuals and the community as a whole. Therefore, strict laws, enforcement, ISP policies, parental responsibility, and international cooperation are critical measures for preventing prohibited/illegal content cybercrime. These measures must be executed with urgency to ensure the safety of minors and the community. The implementation of such measures is essential to prevent the spread of prohibited/illegal content and to mitigate the risks that it poses to the livelihood of individuals and society as a whole.

10. Exploit kits

Exploit kits are becoming the weapon of choice for cybercriminals to spread malware, steal sensitive data, and execute other attacks. Given their ease of use, affordability, and powerful capabilities, it is no wonder that exploit kits have become one of the biggest threats to businesses and individuals alike. This chapter will explore exploit kits in more detail, examining their history, workings, impact, and potential future.

- Background

The first exploit kit that made headlines was the notorious Blackhole, which went live in 2010. The Blackhole exploit kit was a commercial product that was sold to cybercriminals for a price between \$1000 and \$2000 per year. It was designed to exploit vulnerabilities in software, such as Java, Adobe Flash, and Microsoft Silverlight, to infect users with malware. The Blackhole exploit kit quickly became one of the most popular and profitable tools in the cybercriminal's arsenal. The author of the Blackhole exploit kit was eventually arrested in 2013, but this did not stop the proliferation of other exploit kits.

An exploit kit is a software tool that enables cybercriminals to create, distribute, and manage malware attacks. An exploit kit consists of two main components: the malware payload and the exploit code. The payload is usually the malware itself, while the exploit code is the code that enables the malware to infect the victim's computer. Exploit kits can also have various features such as automatic infection, evasion techniques, and data exfiltration.

- Exploit Kits Working

Exploit kits typically use a multi-stage attack process. In the first stage, the attacker uses an initial infection vector, such as a phishing email or malicious website, to entice the victim to visit a compromised site. Once the victim arrives on this site, the exploit kit uses a combination of exploit code to search for vulnerabilities on the victim's computer. These vulnerabilities may be in software such as web browsers, plugins, or plugins, and they are often outdated or not patched. Once a vulnerability is identified, the exploit kit automatically downloads a payload, typically in the form of malware, onto the victim's computer.

Once the malware is installed, it can carry out its intended purpose, which may include stealing sensitive data, encrypting files, or using the victim's computer to launch attacks against other victims. Exploit kits can also have a range of features

such as anti-analysis techniques to avoid detection by security software, encrypted traffic to evade network monitoring, and stealthy exfiltration of data.

- Impact of exploit kits

Exploit kits have had a significant impact on the cybersecurity landscape. They have enabled cybercriminals to launch sophisticated attacks with relative ease, and the financial rewards have been substantial. Some of the damage caused by exploit kits includes data breaches, ransomware attacks, identity theft, and financial fraud. Businesses have been particularly affected, with exploit kits often targeting specific sectors such as finance, healthcare, and retail.

Individuals have also been impacted by exploit kits, with many falling victim to social engineering attacks and unwittingly downloading malware, often resulting in loss of personal data and financial loss. Exploit kits have become increasingly sophisticated in recent years, making them more difficult to detect and defend against.

- Future of Exploit kits

The future of exploit kits is uncertain, with some experts predicting that they will continue to be a significant threat while others suggest that they may decline in popularity. One factor that may influence the future of exploit kits is the availability of vulnerabilities. As software vendors become more proactive in patching vulnerabilities, exploit kits may become less effective. However, as vulnerabilities are discovered daily, and new software is released, this may not be the case.

Another factor that may impact exploit kits' future is the continued use of legacy software, such as Windows 7, which is no longer supported. These older software versions may contain vulnerabilities that will not be patched. They are, therefore, an attractive target for exploit kits.

Finally, the rise of alternative attack methods such as fileless malware or supply chain attacks could herald a decline in the use of exploit kits. The difficulty in detecting these new attack methods makes it challenging to predict whether exploit

kits will continue to be a significant threat.

Exploit kits have been a significant threat to businesses and individuals over the past decade, and their impact is unlikely to diminish soon. The sophistication of exploit kits has increased significantly, making them more challenging to detect and defend against. The continued availability of vulnerabilities, the use of legacy software, and the rise of alternative attack methods such as fileless malware or supply chain attacks will continue to influence exploit kits' future. It is vital for businesses and individuals to remain vigilant and update their systems to avoid falling victim to an exploit kit attack.

5.4 Impact of cybercrime on society

The proliferation of cybercrime poses a substantial risk to individuals who utilize the internet, leading to the widespread theft of personal information from millions of users in recent years. Moreover, the economic impact of cybercrime has been significant, affecting numerous countries on a global scale. Virginia Rometty, the former president and CEO of IBM, has characterized cybercrime as "the largest threat to every profession, every industry, and every firm in the world." To underscore the gravity of this issue, we will now present alarming data from IBM Company [119] that highlights the extent of the damage inflicted by cybercrime upon our society. Please refer to Table 2 for comprehensive information on these distressing statistics.

Table 2: Key findings of IBM report

The average total cost of a data breach	USD 4.35 million
Percentage of organizations that have had more than one breach	83%
Average cost savings associated with fully deployed security AI and automation	USD 3.05 million
Average cost of a ransomware attack, not including the cost of the ransom itself	USD 4.54 million
Average difference in cost where remote work was a factor in causing the breach versus when it wasn't a factor	USD 1 million
Average cost of a breach in the United States, the highest of any country	USD 9.44 million

The Cost of a Data Breach Report is a global report that incorporates data from 17 countries and regions as well as 17 different types of organizations. In this part, we take a look at a number of important metrics on the level of the global average. It was also compared the expenses of various industries across a variety of countries.

The global average total cost of a data breach increased by USD 0.11 million to USD 4.35 million in 2022, the highest it's been in the history of this report (Figure 8). The increase from USD 4.24 million in the 2021 report to USD 4.35 million in the 2022 report represents a 2.6% increase. In the last two years, the average total cost has increased 12.7% from USD 3.86 million in the 2020 report.

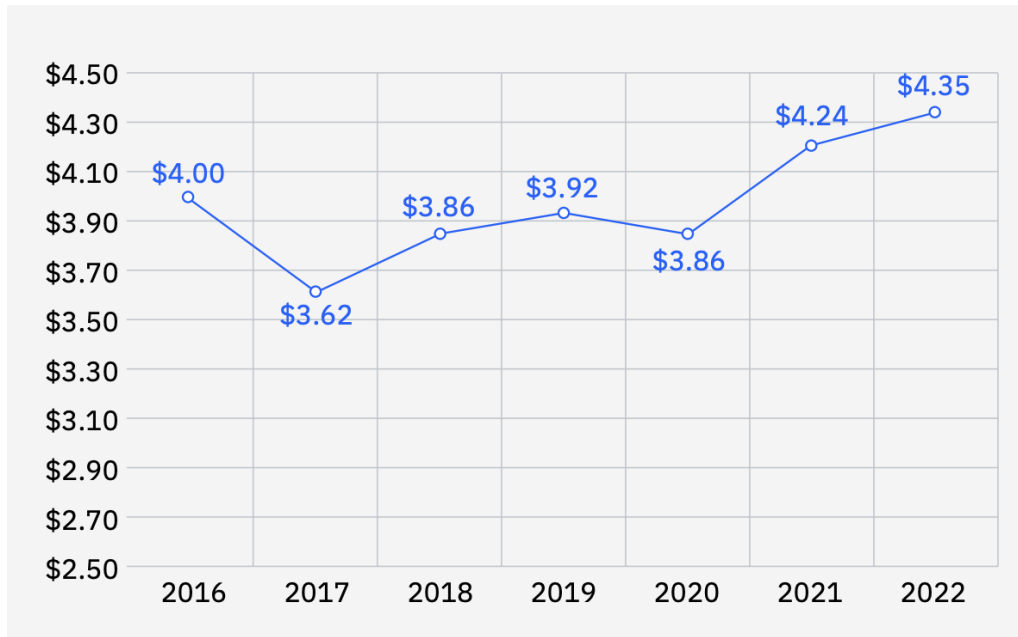


Figure 11: Average total cost of a data breach (measured in USD millions)

The following is a list of the top five countries or regions that had the greatest average cost of a data breach (Figure 9):

1. The United States — USD 9.44 million
2. The Middle East — USD 7.46 million
3. Canada — USD 5.64 million
4. The United Kingdom — USD 5.05 million
5. Germany — USD 4.85 million

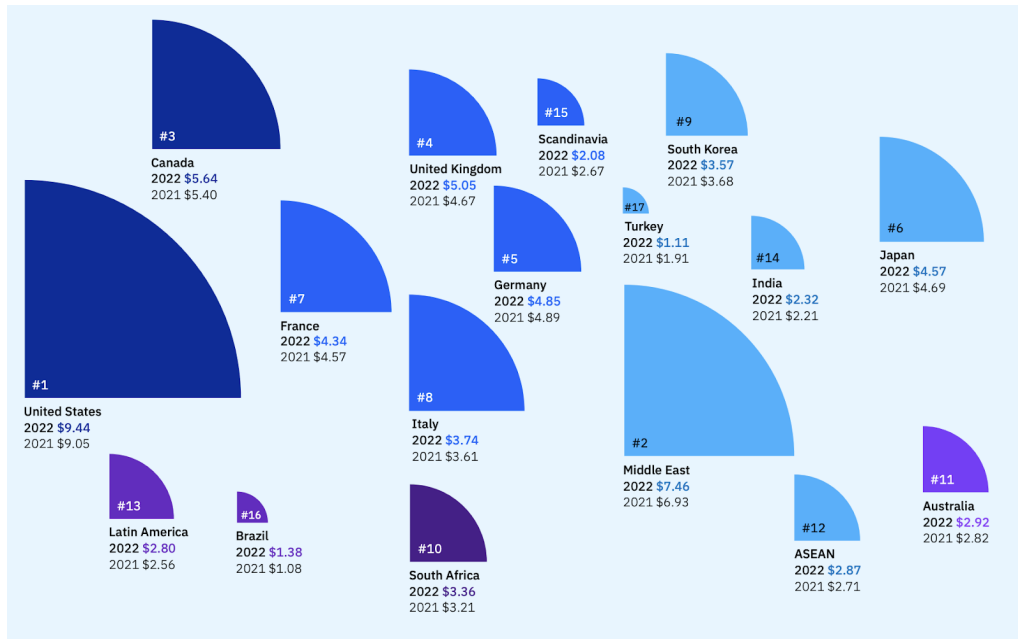


Figure 12: Average total cost of a data breach by country or region (measured in USD millions)

The total average cost of a breach in healthcare went up from USD 9.23 million in the report from 2021 to USD 10.10 million in the report from 2022, representing an increase of USD 0.87 million, or 9.4%. The healthcare industry is one of the most heavily regulated industries, and the government of the United States views it as a crucial component of the country's infrastructure (Figure 10).

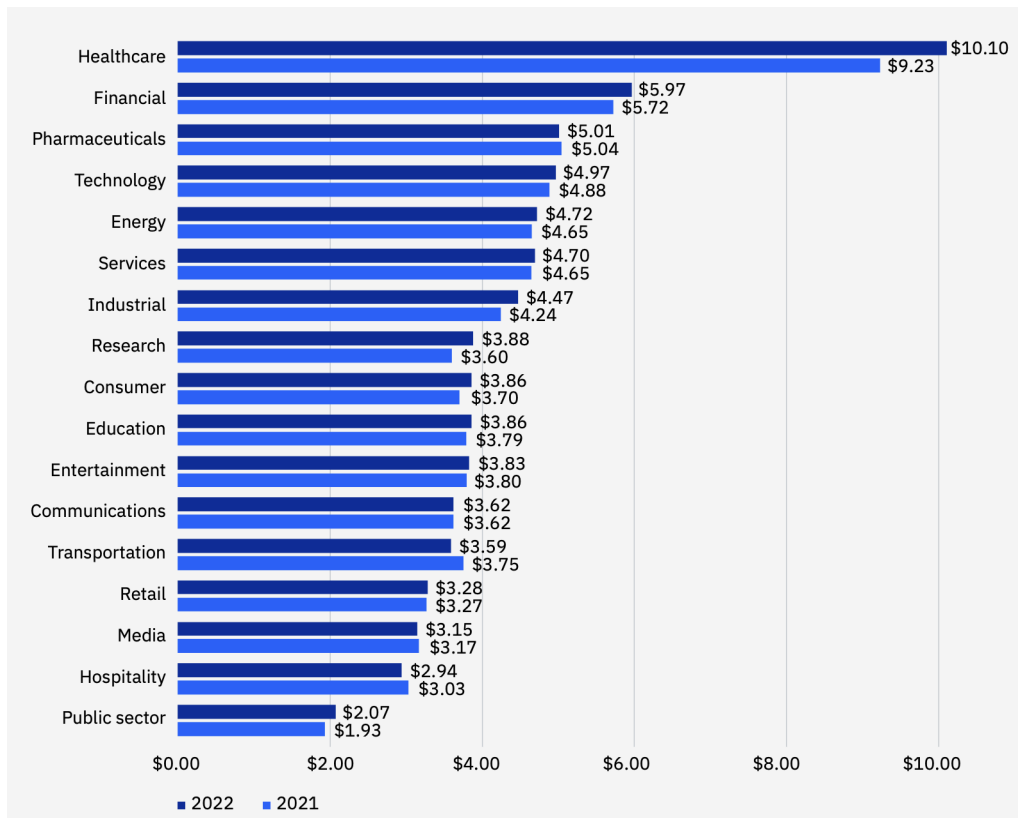


Figure 13: Average total cost of a data breach by industry (measured in USD millions)

6 Conclusion

6.1 Summary and conclusions of the thesis

In addition to investigating different systems and proposing strategies to prevent sensitive information leakage, this study delves deeper into the realm of secure data storage and transmission by examining encryption algorithms. Encryption is a fundamental aspect of information security, and understanding the strengths and weaknesses of various algorithms is crucial for safeguarding sensitive data in modern technological landscapes.

Recognizing the absence of a universal encryption algorithm capable of addressing all security challenges, this research aims to categorize and evaluate the diverse range of algorithms available. By classifying these algorithms based on their problem-solving capabilities, researchers can identify the most suitable options for specific security requirements. This categorization provides valuable insights into the strengths and limitations of each algorithm, enabling informed decisions when implementing encryption measures.

An essential aspect of this study involves analyzing reverse engineering techniques, which play a significant role in unauthorized data extraction and information theft. By comprehensively examining the methodologies employed by malicious actors, researchers gain a deeper understanding of the vulnerabilities that can be exploited. This knowledge empowers developers and security experts to proactively identify potential weaknesses within their systems, enabling them to fortify their applications and protect against potential attacks.

It is important to note that the analysis of reverse engineering techniques in this research is conducted solely from a scientific standpoint. To preserve the integrity of information security, the study refrains from disclosing steps that involve malicious activities, such as adding harmful code or methods for circumventing payment processes. The primary focus remains on providing developers with sufficient information to study vulnerabilities in their applications, encouraging them to enhance the security and integrity of their products.

The research conducted in this study will contribute significantly to the ongoing improvement of computer program security. By gaining a deeper understanding of cyber-crime, particularly its impact on businesses and the resulting financial losses, researchers can identify effective countermeasures and propose strategies to enhance the security of information infrastructure. The insights gained from this investigation will serve as a foundation for designing robust security measures that reduce the likelihood of data compromise and protect against the ever-evolving threat landscape.

The outcomes of this research have been widely disseminated to the scientific community and industry professionals. The publication of the research paper in the esteemed *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)* ensures that the findings reach a broad audience of experts and researchers. Furthermore, the research was presented at the NANA 2021 conference and ICACI 2023 conference, offering an opportunity to exchange knowledge, insights, and practical implications with peers in the field.

By bridging the gap between theoretical research and practical application, this study has the potential to significantly enhance the security infrastructure of various industries. As organizations across sectors strive to protect their sensitive data, the insights gained from this research will serve as a valuable resource for implementing robust security measures and reducing the risks associated with data compromise. Ultimately, this study contributes to creating a more secure and resilient digital ecosystem.

6.2 Future work

While this study has made significant contributions to the field of cybersecurity, encryption algorithms, and reverse engineering, there are several areas that warrant further exploration. The future work outlined below aims to build upon the findings of this research and address some of the remaining challenges in these domains.

- **Advancements in Encryption Algorithms:**

The field of encryption algorithms is continuously evolving, with new algorithms

being developed and existing ones being improved. Future research should focus on exploring emerging encryption techniques, such as post-quantum cryptography, homomorphic encryption, and lattice-based cryptography. Investigating their security properties, performance characteristics, and compatibility with existing systems will be crucial to stay ahead of emerging threats and ensure the development of robust encryption solutions.

- **Analysis of Advanced Reverse Engineering Techniques:**

Malicious actors are constantly evolving their tactics and techniques for reverse engineering software and systems. To effectively combat these threats, future research should delve deeper into advanced reverse engineering techniques employed by attackers. By understanding and analyzing sophisticated methodologies, security practitioners can better identify vulnerabilities, develop countermeasures, and enhance the resilience of their applications.

- **Evaluation of the Human Factor in Cybersecurity:**

While technical solutions play a crucial role in cybersecurity, human factors, such as user behavior and awareness, also significantly impact the overall security posture of organizations. Future research should explore the human element in cybersecurity, including user-centric security education and training, usability of security systems, and the psychology of cybersecurity decision-making. Understanding human factors can help design effective security measures that consider the limitations and capabilities of users.

References

- [1] J. Yin et al. Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning. *World Wide Web*, pages 1–23, 2021.
- [2] J. He et al. A framework for cardiac arrhythmia detection from iot-based ecgs. *World Wide Web*, 23(5):2835–2850, 2020.
- [3] H. Wang et al. Special issue on security and privacy in network computing. *World Wide Web*, 23(2):951–957, 2020.
- [4] H. Wang, J. Cao, and Y. Zhang. A flexible payment scheme and its role-based access control. *IEEE Transactions on knowledge and Data Engineering*, 17(3):425–436, 2005.
- [5] O. G. Abood and S. K. Guirguis. A survey on cryptography algorithms. *International Journal of Scientific and Research Publications (IJSRP)*, 8(7), 2018.
- [6] A. Nadeem and M. Javed. A performance comparison of data encryption algorithms. 2005.
- [7] X. Sun et al. Publishing anonymous survey rating data. *Data Mining and Knowledge Discovery*, 23(3):379–406, 2011.
- [8] E. Kabir et al. Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing. *IEEE Transactions on Cloud Computing*, 8(2):408–417, 2020.
- [9] Y. Ge, M. Orlowska, J. Cao, H. Wang, and Y. Zhang. Mdde: multitasking distributed differential evolution for privacy-preserving database fragmentation. *The VLDB Journal*, pages 1–19, 01 2022.
- [10] J. Li, K. Du, Z. Zhan, H. Wang, and J. Zhang. Distributed differential evolution with adaptive resource allocation. *IEEE Transactions on Cybernetics*, pages 1–14, 2022.

- [11] J. Yin, M. Tang, J. Cao, M. You, H. Wang, and M. Alazab. Knowledge-driven cybersecurity intelligence: Software vulnerability co-exploitation behaviour discovery. *IEEE Transactions on Industrial Informatics*, pages 1–9, 01 2022.
- [12] A. M. Qadir and N. Varol. A review paper on cryptography. 2019.
- [13] X. Sun et al. An efficient hash-based algorithm for minimal k-anonymity. pages 101–107, 2008.
- [14] E. Kabir. A role-involved purpose-based access control model. *Information Systems Frontiers*, 14:809–822, 07 2012.
- [15] H. Wang, L. Sun, and E. Bertino. Building access control policy model for privacy preserving and testing policy conflicting problems. *Journal of Computer and System Sciences*, 80, 12 2014.
- [16] M. Ebrahim, S. Khan, and U. B. Khalid. Symmetric algorithm survey: A comparative analysis. *International Journal of Computer Applications (0975 – 8887)*, 8(6), 2013.
- [17] X. Lv and L. Xu. Aes encryption algorithm keyless entry system. pages 3090–3093, 2012.
- [18] H. Wang, Y. Wang, T. Taleb, and X. Jiang. Editorial: Special issue on security and privacy in network computing. *World Wide Web*, 23, 07 2019.
- [19] H. Wang, Y. Zhang, J. Cao, and V. Varadharajan. Achieving secure and flexible m-services through tickets. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 33(6):697–708, 2003.
- [20] X. Sun, H. Wang, J. Li, and Y. Zhang. Injecting purpose and trust into data anonymisation. *Computers & security*, 30(5):332–345, 2011.
- [21] Y. Zhang et al. On secure wireless communications for service oriented computing. *IEEE Transactions on Services Computing*, 11(2):318–328, 2015.

- [22] H. Wang, Y. Zhang, and J. Cao. Effective collaboration with information sharing in virtual universities. *IEEE Transactions on Knowledge and Data Engineering*, 21(6):840–853, 2008.
- [23] R. Masram, V. Shahare, J. Abraham, and R. Moona. Analysis and comparison of symmetric key cryptographic algorithms based on various file features. *International Journal of Network Security & Its Applications*, 6(4):43–52, 2014.
- [24] D. Pandey, H. Wang, X. Yin, K. Wang, Y. Zhang, and J. Shen. Automatic breast lesion segmentation in phase preserved dce-mris. *Health Information Science and Systems*, 10:1–19, 05 2022.
- [25] R. Singh, S. Subramani, J. Du, Y. Zhang, H. Wang, Y. Miao, and K. Ahmed. Anti-social behavior identification from twitter feeds using traditional machine learning algorithms and deep learning. *ICST Transactions on Scalable Information Systems*, page e17, 05 2023.
- [26] L. Sun, J. Ma, H. Wang, and Y. Zhang. Cloud service description model: An extension of usdl for cloud services. *IEEE Transactions on Services Computing*, PP:1–1, 08 2015.
- [27] E. Kaspersky. Brute force attack: Definition and examples. <https://www.kaspersky.com/resource-center/definitions/brute-force-attack>, 2021. Online. Accessed: 27-Jun-2021.
- [28] H. Wang, J. Cao, and Y. Zhang. Ticket-based service access scheme for mobile users. *Australian Computer Science Communications*, 24(1):285–292, 2002.
- [29] E. Kabir. Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing. *IEEE Transactions on Cloud Computing*, pages 408–417, 08 2015.

- [30] L. Bosnjak, J. Sres, and B. Brumen. Brute-force and dictionary attack on hashed real-world passwords. 2018.
- [31] H. Wang, Z. Zhang, and T. Taleb. Special issue on security and privacy of iot. *World Wide Web*, 21(1):1–6, 2018.
- [32] M. Peng et al. Personalized app recommendation based on app permissions. *World Wide Web*, 21(1):89–104, 2018.
- [33] F. Maqsood, M. Ahmed, M. Mumtaz, and M. Ali. Cryptography: A comparative analysis for modern techniques. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017.
- [34] Opengsm. Aes encryption algorithm and its reliability. <https://www.opengsm.com/blog/algorithm-shifrovaniya-aes/>, 2015. Online. Accessed: 27-Jun-2021.
- [35] M. You, J. Yin, H. Wang, J. Cao, K. Wang, Y. Miao, and E. Bertino. A knowledge graph empowered online learning framework for access control decision-making. *World Wide Web*, 26:1–22, 06 2022.
- [36] H. Wei, Y. Jiao, Y. Mingshan, H. Wang, J. Cao, J. Li, M. Liu, and Chengyuan M. A graph empowered insider threat detection framework based on daily activities. *ISA Transactions*, 2023.
- [37] C. Xiang, N. Fu, and T. Gadekallu. Design of resource matching model of intelligent education system based on machine learning. *ICST Transactions on Scalable Information Systems*, 9:173381, 02 2022.
- [38] H. Zhihan, L. Yuan, and T. Jin. Design of music training assistant system based on artificial intelligence. *ICST Transactions on Scalable Information Systems*, 9:173450, 02 2022.

- [39] J. Shu, X. Jia, K. Yang, and H. Wang. Privacy-preserving task recommendation services for crowdsourcing. *IEEE Transactions on Services Computing*, 14(1):235–247, 2021.
- [40] J. Li, Z. Zhan, H. Wang, and J. Zhang. Data-driven evolutionary algorithm with perturbation-based ensemble surrogates. *IEEE Transactions on Cybernetics*, 51(8):3925–3937, 2021.
- [41] Y. Zhang, Y. Gong, Y. Gao, H. Wang, and J. Zhang. Parameter-free voronoi neighborhood for evolutionary multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 24(2):335–349, 2020.
- [42] Y. Zhou, Z. Lin, L. Tu, J. Huang, and Z. Zhang. Analysis and design of standard knowledge service system based on deep learning. *ICST Transactions on Scalable Information Systems*, page e11, 10 2022.
- [43] M. Fatima, O. Rehman, and I. Rahman. Impact of features reduction on machine learning based intrusion detection systems. *ICST Transactions on Scalable Information Systems*, 9:447, 04 2022.
- [44] V. M. Marabathuni and V. Gopejenko. Mobile security: Analysis and challenges of information. 2021.
- [45] J. Hong, S. Huang, A. Gao, and Y.-C. Lin. Design and implementation of mobile phone information security system based on android platform. 2021.
- [46] Z. Hong, Z. Qiu, S. Zeng, S. Wang, and M. Sandrine. Research on fusion encryption algorithm for internet of things monitoring equipment. pages 425–429, 2017.
- [47] X. Li, L. Yu, and L. Wei. The application of hybrid encryption algorithm in software security. pages 669–672, 2013.

- [48] M. Pasquet, J. Reynaud, and C. Rosenberger. Secure payment with nfc mobile phone in the smarttouch project. *2008 International Symposium on Collaborative Technologies and Systems*, May 2008.
- [49] V. et al Pasupathy. The australian pcehr system: Ensuring privacy and security through an improved access control mechanism. *EAI Endorsed Trans. Scalable Information Systems*, 3(8):e4, 2016.
- [50] R. Sarki, K. Ahmed, H. Wang, Y. Zhang, and K. Wang. Convolutional neural network for multi-class classification of diabetic eye disease. *EAI Endorsed Transactions on Scalable Information Systems*, 9(4):e5, Dec. 2021.
- [51] C. Wang, B. Sun, K. Du, J. Li, Z. Zhan, S. Jeon, H. Wang, and J. Zhang. A novel evolutionary algorithm with column and sub-block local search for sudoku puzzles. *IEEE Transactions on Games*, PP:1–11, 01 2023.
- [52] F. Zhang et al. Decision-based evasion attacks on tree ensemble classifiers. *World Wide Web*, 23(5):2957–2977, 2020.
- [53] K. Cheng, L. Wang, Y. Shen, H. Wang, Y. Wang, X. Jiang, and H. Zhong. Secure k-nn query on encrypted cloud data with multiple keys. *IEEE Transactions on Big Data*, PP:1–1, 05 2017.
- [54] M. Peng, J. Zhu, H. Wang, X. Li, Y. Zhang, X. Zhang, and G. Tian. Mining event-oriented topics in microblog stream with unsupervised multi-view hierarchical embedding. *ACM Transactions on Knowledge Discovery from Data*, 12:1–26, 04 2018.
- [55] H. Jiang et al. Sentence level topic models for associated topics extraction. *World Wide Web*, 22(6):2545–2560, 2019.

- [56] T. Huang, Y. Gong, S. Kwong, H. Wang, and J. Zhang. A niching memetic algorithm for multi-solution traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 24(3):508–522, 2019.
- [57] Y. Wang, Y. Shen, H. Wang, J. Cao, and X. Jiang. Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications. *IEEE Transactions on Big Data*, PP:1–1, 08 2016.
- [58] A. Gallardo, H. Kim, K. Kim, and C. L. Tianying Li. Mobile security strategies and usability problems in ipv and stalking contexts. 2021.
- [59] K. Gai, M. Qiu, and H. Zhao. Privacy-preserving data encryption strategy for big data in mobile cloud computing. *IEEE Transactions on Big Data*, 7(4):678–688, 2021.
- [60] National Institute of Standards and Technology. F.i.p. standard, advanced encryption standard (aes). Technical report, 2001.
- [61] R. Zhu. Data encryption algorithm based on chaos sequence in computer network security. pages 1189–1192, 2023.
- [62] G. Kumar. Performance evaluation of various symmetric encryption algorithms. *2014 International Conference on Parallel, Distributed and Grid Computing*, 2014.
- [63] B. Schneier. Blowfish algorithm. <https://www.schneier.com/academic/blowfish/>. Online. Accessed: 27-Jun-2021.
- [64] H. Mohan, H. Reddy, and A. Raji. Performance analysis of aes and mars encryption algorithms. *International Journal of Computer Science Issues (IJCSI)*, 8(4):363, 2011.
- [65] M. G. Aruna and K. G. Mohan. Secured cloud data migration technique by competent probabilistic public key encryption. *China Communications*, 17(5):168–190, 2020.

- [66] O. D. Apuke. Quantitative research methods: A synopsis approach. *Kuwait Chapter of Arabian Journal of Business and Management Review*, 6(11):40–47, 2017.
- [67] S. Bhunia and M. Tehranipoor. Side-channel attacks. *Hardware Security*, pages 193–218, 2019.
- [68] H. Wang, L. Sun, and E. Bertino. Building access control policy model for privacy preserving and testing policy conflicting problems. *Journal of Computer and System Sciences*, 80(8):1493–1503, 2014.
- [69] A. Ju and Z. Wang. Convolutional block attention module based on visual mechanism for robot image edge detection. *ICST Transactions on Scalable Information Systems*, 9:172214, 11 2021.
- [70] N. Tawhid, S. Siuly, K. Wang, and H. Wang. Automatic and efficient framework for identifying multiple neurological disorders from eeg signals. *IEEE Transactions on Technology and Society*, PP:1–1, 03 2023.
- [71] R. Li. A novel image clustering method based on coupled convolutional and graph convolutional network. *ICST Transactions on Scalable Information Systems*, 9:172132, 11 2021.
- [72] M. Thangavel, P. Varalakshmi, and C. Abinaya. A comparative study of attribute-based encryption schemes for secure cloud data outsourcing. pages 261–266, 2017.
- [73] I. Morgado, A. Paiva, J. Faria, and R. Camacho. Gui reverse engineering with machine learning. pages 27–31, 2012.
- [74] S. Agarwal and A. Aggarwal. Model driven reverse engineering of user interface — a comparative study of static and dynamic model generation tools. pages 268–273, 2014.
- [75] D. Amalfitano, A. Fasolino, and P. Tramontana. Reverse engineering finite state machines from rich internet applications. pages 69–73, 2008.

- [76] Y. Shen et al. Microthings: A generic iot architecture for flexible data aggregation and scalable service cooperation. *IEEE Communications Magazine*, 55(9):86–93, 2017.
- [77] Y. Gong and G. Srivastava. Multi-target trajectory tracking in multi-frame video images of basketball sports based on deep learning. *ICST Transactions on Scalable Information Systems*, page e12, 10 2022.
- [78] Z. Lin and J. Lin. Research on knowledge management of novel power system based on deep learning. *ICST Transactions on Scalable Information Systems*, page e10, 10 2022.
- [79] K. Mobley. Reverse engineering for software performance engineering. pages 302–304, 2007.
- [80] C. M. Allwood. The distinction between qualitative and quantitative research methods is problematic. *Quality & Quantity*, 46(5):1417–1429, 2011.
- [81] P. Fagette. Reverse engineering by design : Using history to teach. *IEEE Pulse*, 4(1):33–38, 2013.
- [82] Y. Ma, Y. Zhao, Z. Zhang, and J. Wang. Distributed data multi-level storage encryption method based on full-flow big data analysis. pages 664–668, 2023.
- [83] Y. Liu, C. Li, Z. Zheng, Q. Guo, and X. Gong. Reverse engineering workload measure based on function classification. pages 660–665, 2023.
- [84] Y. Hui and L. Zesong. Research on real-time analysis and hybrid encryption of big data. pages 52–55, 2019.
- [85] G. Wang, N. Han, Y. Lv, and D. Zhang. Application of object reverse engineering technology in equipment components design. volume 1, pages 320–323, 2012.

- [86] Siyuan C., Xuerong Y., Xiangwei Z., and Shiquan L. A redesign methodology for reverse engineering integrated with haptic modeling. volume 3, pages 105–108, 2010.
- [87] P. More, S. Chandugade, S. Rafiq, and P. Pise. Hybrid encryption techniques for secure sharing of a sensitive data for banking systems over cloud. pages 93–96, 2018.
- [88] T. Gordon, E. Kilgore, N. Wylds, and M. Nowatkowski. Hardware reverse engineering tools and techniques. pages 1–6, 2019.
- [89] A. Dalai, S. Das, and S. Jena. A code obfuscation technique to prevent reverse engineering. pages 828–832, 2017.
- [90] H. I. Lim. Comparative analysis of code obfuscation approaches to protect software products. *International Journal of Computer Theory and Engineering*, 9(1):28–31, 2017.
- [91] J. Thankappan and V. Patil. Detection of web design patterns using reverse engineering. pages 697–701, 2015.
- [92] A.E. Hassan and R.C. Holt. The small world of software reverse engineering. pages 278–283, 2004.
- [93] J. Gao. Retracted: Basketball posture recognition based on hog feature extraction and convolutional neural network [eai endorsed scal inf syst (2022), online first]. *ICST Transactions on Scalable Information Systems*, 9:173787, 04 2022.
- [94] D. Perumal, L.R Sudha, K. Kalaivani, and J. Ganesh. Scene classification of remotely sensed images using optimized rsisc-16 net deep convolutional neural network model. *ICST Transactions on Scalable Information Systems*, 9:173292, 02 2022.

- [95] N. Veeraragavan, L. Arockiam, and S. S. Manikandasaran. Enhanced encryption algorithm (eea) for protecting users' credentials in public cloud. pages 1–6, 2017.
- [96] A. Mahfoud, A. Bakar-Sultan, A. Azim Abd, N. Mohd Ali, and N. Admodisastro. Code obfuscation. where is it heading? *International Journal of Engineering & Technology*, 7(4.1):22, 2018.
- [97] G. Kumar. A survey on program code obfuscation technique. *Engineering and Technology Journal*, 2016. Published.
- [98] M. S. Akhtar and T. Feng. Comparison of classification model for the detection of cyber-attack using ensemble learning models. *ICST Transactions on Scalable Information Systems*, 9:173293, 02 2022.
- [99] Y. Zhang and Y. Yuan. A novel dilated convolutional neural network model for road scene segmentation. *ICST Transactions on Scalable Information Systems*, 9:173164, 01 2022.
- [100] X. Pang, Y. Ge, K. Wang, A. Traina, and H. Wang. Patient assignment optimization in cloud healthcare systems: a distributed genetic algorithm. *Health Information Science and Systems*, 11, 06 2023.
- [101] S. Rehman, C. Allgaier, and V. Gruhn. Security requirements engineering: A framework for cyber-physical systems. pages 315–320, 2018.
- [102] A. Setiawan, A. Syamsudin, and A. Sastrosubroto. Information security governance on national cyber physical systems. pages 1–6, 2016.
- [103] D. Shivpuri. Cyber crime: Are the law outdated for this type of crime. *IJRESM*, 4(7):44–49, Jul. 2021.
- [104] L. Parrondo. Industrial cyber security solutions for the connected enterprise. pages 1–27, 2014.

- [105] P.D. Persadha, A.A. Waskita, and S. Yazid. Comparative study of cyber security policies among malaysia, australia, indonesia: A responsibility perspective. pages 146–150, 2015.
- [106] W. hong, W. Sheng, Z. Qing, and Z. Jian. Research on data encryption of network communication based on big data. pages 129–131, 2020.
- [107] J. Zhang et al. On efficient and robust anonymization for privacy protection on massive streaming categorical information. *IEEE Transactions on Dependable and Secure Computing*, 14(5):507–520, 2015.
- [108] V. Gautam, R. Tiwari, A. K. Jain, and A. Agarwal. Research pattern of internet of things and its impact on cyber security. pages 260–263, 2022.
- [109] Top cybersecurity statistics, trends, and facts | cso online. <https://www.csoonline.com/article/3634869/top-cybersecurity-statistics-trends-and-facts.html>, 2022. [Accessed: 12-Jul-2022].
- [110] What is cybercrime? - definition from techopedia. <https://www.techopedia.com/definition/2387/cybercrime>, 2022. Accessed: 12-Jul-2022.
- [111] A. Choudhary, A. Chaudhary, and S. Devi. Cyber security with emerging technologies challenges. pages 1875–1879, 2022.
- [112] H. Wang and L. Sun. Trust-involved access control in collaborative open social networks. *2010 Fourth International Conference on Network and System Security*, pages 239–246, 2010.
- [113] N. Benias and A. P. Markopoulos. A review on the readiness level and cyber-security challenges in industry 4.0. pages 1–5, 2017.

- [114] M. Frank, M. Leitner, and T. Pahi. Design considerations for cyber security testbeds: A case study on a cyber security testbed for education. pages 38–46, 2017.
- [115] M. Şenol. An approach for creation and implementation of national cyber security strategy. pages 189–194, 2017.
- [116] F. Skopik and S. Filip. Design principles for national cyber security sensor networks: Lessons learned from small-scale demonstrators. pages 1–8, 2019.
- [117] K. Bhat, V. Sundarraj, S. Sinha, and A. Kaul. Ieee cyber security for the smart grid. *IEEE Cyber Security for the Smart Grid*, pages 1–122, 2013.
- [118] N. Ahmad, U. Mokhtar, Fariza P. F., Z. Othman, Yusri Hakim Y., Huda S., and Siti N. Cyber security situational awareness among parents. pages 1–3, 2018.
- [119] Cost of a data breach 2022 | ibm. <https://www.ibm.com/reports/data-breach>, 2022. [Accessed: 15-Aug-2022].