National College of Ireland

Network-Specific Data Decryption Tool for Enhancing User Profile Security

MSc Research Project

Muhammad Ahssan Sajjad

Student ID: 23156767

School of Computing

National College of Ireland

Supervisor:Liam Mccabe

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Muhammad Ahssan Sajjad

Student ID: 23156767

Programme: Msc in Cybersecurity **Year:** 2023

Module: MSc Research Project

Supervisor: Liam Mccabe

Submission Due Date: 16/09/2024

Project Title: Network-Specific Data Decryption Tool for Enhancing User Profile Security

Word Count: 8215 **Page Count**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: *Ahssan sajjad*

Date: 15/09/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Abbreviations

Abbreviation	Full Form
AES	Advanced Encryption Standard
IoT	Internet of Things
SSID	Service Set Identifier
MAC	Media Access Control
IP	Internet Protocol
VM	Virtual Machine
TLS	Transport Layer Security
WSN	Wireless Sensor Network
GUI	Graphical User Interface
QKD	Quantum Key Distribution
EaaS	Encryption as a Service

Network-Specific Data Decryption Tool for Enhancing User Profile Security

Muhammad Ahssan Sajjad
23156767

Abstract

The rationale for the outcome of this project was to develop an application that would enhance the security of data of the said profile with the guarantee that the decryption process will only be done with the involvement of the specific network that is deemed secure. This tool was meant to reduce such anxieties as data compromise to a specific platform; hence, the analyst could compute the user data safely in a particular context of a network. The objective was that decryption should only happen at the trusted network and if the host gets connected to an untrusted network, exposure must be prevented. It included dynamic network detection, authentication of the user and the reliable encryption. Several attributes' names were established and cross tabulated with the secure values kept in Operating System Network parameters. User authentication motivated the user to input a username and password, which in turn was separated, hashed then stored in the system. Talking specifically about data encryption, the Fernet method, which is a type of the symmetric encryption that provides the confidentiality and considers the integrity of data, was used. The keys were well maintained ensuring that they met all the standards of secure management like key storage and frequent key renewal. Thus, the checks on the network were made complicated, and decryption was only done within the trusted network, and data was re-encrypted if the network was changed. Thus, intensive documentation and ideal error handling provided stability and keeping accurate records. Functional and efficacy as well as security tests were used to confirm the ability of the tool to work as intended.

1 Introduction

The importance of privacy was not set aside in 2024, as more account information and various personal data were kept on computers and other supporting technologies. Thus, the transition to tablet-based storage was catalysed by the enhancement of the utilization of technology to encompass a variety of tasks and spheres. Thus, with the constant advancement of information technologies, new threats appeared, mainly concerning access to this information by unauthorized personnel. This carries a significant risk of leaking out information to the wrong hands, hence highlighting the importance of having security measures that would only allow data to be accessed only by authorized persons, and via safe channels.

To overcome these challenges, this project created a tool that decodes users' profile information only if the device is connected to a pre-designated secure network. The tool is effective because decryption is limited only to the secure networks of the organization which reduces the chances that the data will be leaked when a device connects to an untrusted or a compromised network. This approach is a significant improvement compared to typical methods of protecting information in a constantly evolving network.

Thus, the primary objective of this work was to increase the level of security of user profile data, including social networks, with reference to network-specific decryption. Although different encryption technologies ensure that the information is protected when stored and when in the process of being transmitted, current data protection fails to protect the data from being accessed by unauthorized people because of insecure connections in a network. To enhance the level of security during the decryption process, this tool uses network parameters; this implies that data can only be decrypted within a trusted network environment.

The research question guiding this project was: "How can a tool be made or designed or created in such a way that it can only carry out the task of decrypting the particular user profile information only when it is connected to a particular secure network?" The development of the tool was carried out in a way that would make it only possible to decrypt and present information belonging to the user profile for analysis upon the connection of the specified network switch port. Any attempts of decryption that could take place in any other environment outside of their secure network environment were intended to fail and would not allow access to anybody who happens to be outside of this secured environment.

To meet the objectives, the tool encompassed the following components. Detection scripts which use flow activity information were run dynamically to compare current network parameters with the ones available in a trustworthy configuration list, which is a secure network list. Else, if the network parameters were same as trusted configurations, the tool went on for user authentication where the username and the password had to be entered by the user. These credentials are encrypted and safely stored in order not to allow other people or unauthorized users to access them.

The collected user profile data was then saved encrypted using the Fernet encryption method, which is among the most secure methods. The encryption keys were the other protocol that was well protected, in most cases stored in the hardware security modules to avoid any attempt to access the keys. The tool made it possible to decrypt the data only in the context of the safe networking as the tool was monitoring the network parameters and re-encrypting the data, if it detected a change in the network setting, thus preserving the confidentiality and the integrity of the data.

Documentation of the procedures was also done very well and adequate logging and error handling was also done so that the tool could be easily debugged. Operations that include system login and network parameter checks for example were recorded to provide a record of the operations. Safeguards were implemented as easy to follow for the users, and the instructions of what to do in case of an error could be easily pointed out to the users.

The validity of the tool was established from success performance tests and evaluations that showed the capability of the tool while in operation in preserving the integrity of data over different networked systems. For functional testing the basic practicalities such as login, data insert, data search, file upload, file download and logout were tested, and for performance,

assess the efficiency of the tool. More security tests then supported that the tool would not be easily susceptible to invasion. Participants agreed that tool would be secure when used in their environment and came up with recommendations as to how the graphical user interface can be enhanced, and the configuration instructions made clearer.

To sum it up, it can be concluded that the creation of this tool to decipher data on the level of the given network is a valuable breakthrough in the field of data protection. Since decryption is limited to the internal secure network, the tool increases data security as well as discourages illegitimate users' access. Collectively, this approach provides a rather stationary yet effective way of guarding user privacy in a world of constantly evolving technologies.

2 Related Work

The field of networks and data security has been blessed with numerous studies hence availing a variety of methods and strategies aimed at enhancing the security and efficiency of data connections and storage in different network systems. The following literature critically examines these contributions by reflecting on their methodological approaches and the accomplishments and outstanding issues based on their findings.

For instance, Shamala et al. [1] discuss changes in the lightweight versions of AES for the Internet of Things. These papers focus mainly on how AES can be best implemented on the reduced capability devices, which is done by proposing changes that elevate the security level without straining the device moreover much. However, these modifications enhance efficiency in terms of cycles from other point of views, these changes may reduce the security level of the system, because often, simplicity can harm the system from attacks of a higher level. These trade-offs are not effectively discussed within the study particularly with highly adversarial relationships.

Beloglazov and Buyya [2] put forward OpenStack Neat: a framework for dynamic and energy efficient consolidation of virtual machines in the clouds. The relevance of this work lies in the fact that it contributes to the solutions of problems associated with resource utilization and energy efficiency in cloud computing environment. Through movement of the VMs according to the positions defined by processing data from the net vibration, the framework improves energy consumption. This study has a major weakness of overlooking several threats that characterize VM movements, most notably the susceptibility of data to eavesdropping during transfer.

Also, regarding the challenges as well as the opportunities of big data computing, Kune et al. [3] write about aspects concerning big data computation and more subtopics thereof. They express the need for highly malleable infrastructure for data processing and showcase the lack of scalability of the nowadays' Database Management Systems in processing big volumes of information. While this review is quite extensive, its failure to analyse the protective needs to safeguard information within these gigantic assimilation structures is a major drawback especially in today's world where data breaches are on the rise.

In the paper of Sevin et al. [4], the authors consider the software implementation of the lightweight block ciphers for IoT technologies and provide the list of the general cryptographic methods for protecting the data to be implemented in the resource-defying systems. In the survey various algorithms are described very well along with their efficiency. But the authors fail to discuss on how these algorithms are incorporated in the real IoT networks Also they don't provide enough exploration of the operational concerns like protocols interpretations & updates and its effect on the interconnectivity of the devices.

Sherry et al. [5] propose outsourcing the network processing in the cloud, which makes middle-boxes like firewall and load balancer as virtual services in the cloud. Although this approach effectively makes use of cloud services' scalability, elasticity, and flexibility, it presents critical concerns on the reliability, latency and security of data particularly when it travels across dissimilar cloud systems. However, the study fails to present a clear and logical method for tackling these issues, thus underlining one of the main weaknesses in the pragmatic application of the research.

Chow, Mokbel & He [6] study the problem of private location reporting in wireless sensor networks (WSNs). They put forward a mechanism which works under the disguise of location information to suppress the discloser of individuals while keeping the surveillance function. This approach proves to be mildly invasive and still serves as a functional compromise between complete privacy and being able to use the phone. However, the anonymization is not fully impenetrable from de-anonymization attack and in correlation analysis it can reconstruct the sensitive field data.

Another classification of conflict sources in NSP has been defined by Hamed, Al-Shaer, and Marrero [7] that gives a clear system of how these conflicts can be identified. This contribution is highly appreciated in the attempt to demystify what may otherwise be highly confusing and twisted security policies in today's complex network scenario. Nevertheless, the application of the taxonomy in dynamic and realistic network environments is sparse as it results into scalability and continuously varying threat types for updates.

In the article Amini et al. [8], a systematic review is conducted on secure data storage and sharing approaches in cloud computing. They assess diverse approaches for safeguarding the data which is stored in clouds regarding its integrity, confidentiality and availability. While the number of analysed sources is rather big, threats and trends in cryptographic methods that aim to counteract them, are not explored in detail. Furthermore, there is no stress on the considerations usually required for deploying management and satisfying the compromises needed to get acceptable performance.

In their work, Liao et al. [9] describe a method of location-based data encryption to increase information security in mobile IS. This method is somewhat like the role-based data access method but fully permits the data based on two aspects, which are role and geographic location information. As mentioned in the identified use cases of integrating location data

into the security mechanisms, three new threats are given below: However, the standard study investigates these threats, but this has not been done extensively.

Kang, Veeravalli, and Aung [10] review ESPRESSO that is an Encryption as a Service (EaaS) framework for cloud storage systems. It applies to most of the troubles in the multi-cloud environments, offering a strong answer to the data security conundrum. However, the inter-cloud communication within the ESPRESSO system may cause or result one some limitations regarding homogeneity and coherence of the encrypted information supported in different cloud environment. Perhaps, there is still a need for the authors to bring out the specifics of these factors and perhaps the contingencies involved.

Hu, Zhao, and Zheng [11] propose a paper on a reversible database watermarking technique for the security as well as for the verification of ownership where distortion is maintained. It is one of the few methods that can trace data leakage while at the same time keeping data fully secure. However, the study fails to present in detail the conditions under which this watermarking technique can be applied in large database, not to mention the thorough study of the effects it has on the performance of the large Database.

A novel and efficient public auditing protocol which is designed for cloud data with a dynamic structure is described by Shen et al. [12]. This protocol also makes cloud storage to be more reliable as well as more transparent. Although the research intervention focuses on the problem of finding adequate and safe ways to audit, it is essential to consider the problem of using third-party auditors while collecting information, which is not fully resolved.

Some of the research works show that context-awareness can help in increasing security. For instance, location-based measures have been suggested to protect data disclosure in some geographical regions: (Khoshgozaran, 2007 [13]. However, these approaches are based on GPS data which can be fault in conditions where precise location information is missing or wrong.

It also looks at prior work on the transmission of information over networks with emphasis on aspects of security. For example, the Transport Layer Security (TLS) protocol guarantees that the data is safeguarded against individuals gaining unauthorized access and altering it when in the process of transfer (Rescorla, 2018) [14]. However, such protocols prove efficient in the protection of such data on transit, yet there are assignable protocols that do not explain the decryption of information based on the parameters of certain networks, therefore, granting some risks.

In similar subfields further research was done to emphasise the role of cryptography to supply networks security opposed to traditional community menace. Alshehri and Alhamed [15] also explained that cryptography plays a very big role for protecting data since attraction of technology utilization is increasing not only in the governmental institutions, but also in private organizations. Encryption replaces the original clear text with another text such that information cannot be read by anyone who is not supposed to read it.

Zhou & Liu, [17] focuses on the development of data encryption and the use in enhancing computer networks security. Their work addresses different algorithms and ways to apply them to avoid the threats in digital environments.

Akter et al. [16] include a detailed overview of quantum cryptography as the basis of modern network protection and the possibility of radically changing existing approaches to it. The current study aims at exploring QKD as one of the novel paradigms shifts in the manner networks are protected against current and emergent cyber threats.

Mongay Batalla [19] also reveals the ways in which AI and ML can strengthen network security measures. The work exemplifies that with AI/ML, it takes longer to identify higher numbers of anomalies and the ability to add more layers of defence to mitigate threats which in general improves the security of the network.

A current study by authors D'Orazio, Choo, and Yang [18] involves data exfiltration from IoT devices with focus on iOS-based devices. Thus, the study demonstrates various types of attacks regarding IoT devices and emphasizes the significance of cryptographic principles for the proper authorization of data and their protection.

Sharma, et al. [20] propose a literature review concentrating on Hash function Applications, Attacks, and the development in Cryptography and Network Security. From the released papers, readers can get comprehensive information on different cryptographic algorithms and the capability of protecting information and combating cybercrimes. They also detail the roles of Crypto-analysis and evaluation showing the importance of the constant improvement of the cryptographic methods to respond to emerging security threats.

3 Research Methodology

The research methodology for developing a network-specific data decryption tool is designed to systematically address the security challenges associated with decrypting user profile data only when connected to a trusted network. The process is structured into several detailed steps to ensure robustness and reliability.

3.1 Step 1: Define the Specific Network

Identify Network Parameters: The first step involves determining certain parameters that are essential to the cardinality of the trusted network to enable decryption. These parameters are the SSID or SAP name- string which specifies a name to given Wi-Fi network, MAC address – a unique identifier of specific network interfaces, and, perhaps, the IP range – a set of IP addresses to a certain network.

Configuration File: After the identification of network parameters, these parameters would be stored in secure configuration file. This file is basically reference for decryption tool that ensures that it knows criteria for trusted network. It is very important that this configuration

file should be protected from unauthorized access or any amendments because it contains very sensitive information critical to security of decryption process.

3.2 Step 2: Detect the Current Network

Network Detection Script: Network detection scripts are written to dynamically detect current network parameters. Scripts can be different if there is different operating being used like script for windows can't be used for mac as well, but the scripts written in this study and specifically for windows OS. The functionality of these scripts ensures that tool would have real-time information about network without any error because it would extract network information from windows and not by any ither means.

3.3 Step 3: Store Encrypted User Profile Data

Encrypt Data: The user profile data should be encrypted and there are a lot of encryption libraries in python and other similar high-level languages. In python, cryptography library would be used here. The encryption algorithm must be decided which is Fernet since it offers a tight level of security.

Store the Key Securely: This can be done by ensuring that key is not accessed by any other entity than the one intended to use it. This security is implemented both for software and physical access, because most secure systems can be breached if attacker gains physical access to them. Moreover, the security of the encryption key is imperative since any individual who holds this key will be able to perform decryption on the secured data.

3.4 Step 4: Check Network and Decrypt Data

Network Check: The tool compares the current network parameters, as detected by the script, with the predefined trusted network parameters stored in the configuration file. This comparison is critical in determining whether the decryption process should proceed.

Decrypt Based on Network: If the current network is the trusted network, then the tool goes ahead to decrypt the encrypted data. This makes sure that decryption is only done when the network is secure and fully trusted hence protecting the data from being viewed on some other untrusted and or malicious networks.

3.5 Step 5: Implement Access Control

User Authentication: For increase in security of the tool and for prevention of any unauthorized access, user authentication would be used. This step makes sure that users who are authorized for access can only start the process of decryption and only they would be allowed to access the system. Authentication would involve username and password that would be stored in form of hash in database.

Combine Authentication and Network Check: Most important part is this that decryption would only be allowed if both of network match and if user is connected to pre-defined network This would makes sure that if network parameters are matched then decryption would be done if user is connected to specific trusted network and thus no one can access the system without passing through all these parameters.

3.6 Step 6: Handle Exceptions and Logging

Error Handling: The tool would minimize errors that could arise during detection of network for decryption. As discussed before that network detection would be done using scripts on power shell so there is no chance of error in network detection. Moreover, user would be guided about the error like if it is not connected to that specific network that would help user to minimize errors from his side.

Logging: Logs of each step would be maintained so that all the operation that are being done on the tool would be monitored. Maintaining logs could include the time of logging in the tool and if there is unsuccessful login then network parameters would be shared for better monitoring of the system.

4 Design Specification

In the design specification of the network-specific data decryption tool, the components that are needed in the process as well as their functions are described to guarantee efficient and secure decryption regarding the network parameters. The system is comprised of several modules which are essential for the general architecture of the security system.

The first component is the definition and storage of network parameters, that is, some given values describing the properties of the network to be created. These network settings include the SSID (Service Set Identifier), MAC (Medium access control) address, and the IP (Internet Protocol Address) range which are exposed are encoded thereby formulating a security configuration file. This file would serve as a basis of the tool for it to conclude whether the current network remains trustworthy. Security of this file must be ensured because it carries very important information that is vital to the decryption process.

The second component is the network detection module that runs in parallel to the event detection module. This module is specifically responsible for the dynamic identification of the parameters of the current network. The usage of operating system specific scripts, for instance, PowerShell in the case of Windows, is used to obtain current network data in real-time. This guarantees the tool has the most current and precise information about the network connected.

Third and finally, there is the data encryption module. While attaining the state of anonymity, all the data of the user's profile is encrypted via Fernet, which is said to be highly secure. Encryption ensures the security of the data at the time of storage as well as at the time of movement from one location to the other. The keys employed in the encryption of a particular

data are managed well, sometimes it is next to impossible for individuals to touch those keys other than those who have the authority to do so such as through employment of a hardware module exclusively for the keys.

The fourth module is the access control module it incorporates the user authentication feature into the system. This module makes certain that only a permitted user can pull the string to have the encrypted data decrypted. Authentication can include logon, which commonly uses a username/password and where the given details are hashed and remains in the database. The security network checks, and user authentication are preconditions for decryption, so it renders a highly secure option.

Table 4.1: Design Components

Design Component	Functionality
Network Parameter Definition	Identify and store trusted network parameters such as SSID, MAC address, and IP range in a secure configuration file.
Network Detection Module	Dynamically detect current network parameters using tailored scripts for different operating systems.
Data Encryption Module	Encrypt user profile data using Fernet and manage encryption keys securely to prevent unauthorized access.
Access Control Module	Implement user authentication to ensure only authorized users can initiate the decryption process.
Logging Module	Manage error management and maintain comprehensive logs of all operations, including login attempts and network parameters.

The last component of the system is the logging module that corresponds to error management and operational tracking. They also try not to output wrong results when detecting the network and decrypting it, giving helpful messages to the users. Proper logs are kept capturing all actions performed on the system, such as the login information as well as the system connection parameters that increases the system's audit trail.

Altogether, the design specification makes it possible to have all components of the network-specific data decryption tool integrated ensuring that the tool will be a secure solution for decryption of user's profile data depending on the network parameters. The use of sophisticated encryption, identification of the dynamic network, user identification, and recording fulfil the requirements of security and functionality of the tool.

1. **Network Parameter Definition**: Trusted network parameters should be stored in configuration file.
2. **Network Detection Module**: For Detection of current network parameters, there should be use of script or software module.
3. **Data Encryption Module**: For encryption and decryption of user profile data, encryption technique that would be used is FERNET.
4. **Access Control Module**: Authenticated users would only be allowed to decrypt data.
5. **Logging Module**: Attempts and error would be tracked using this.

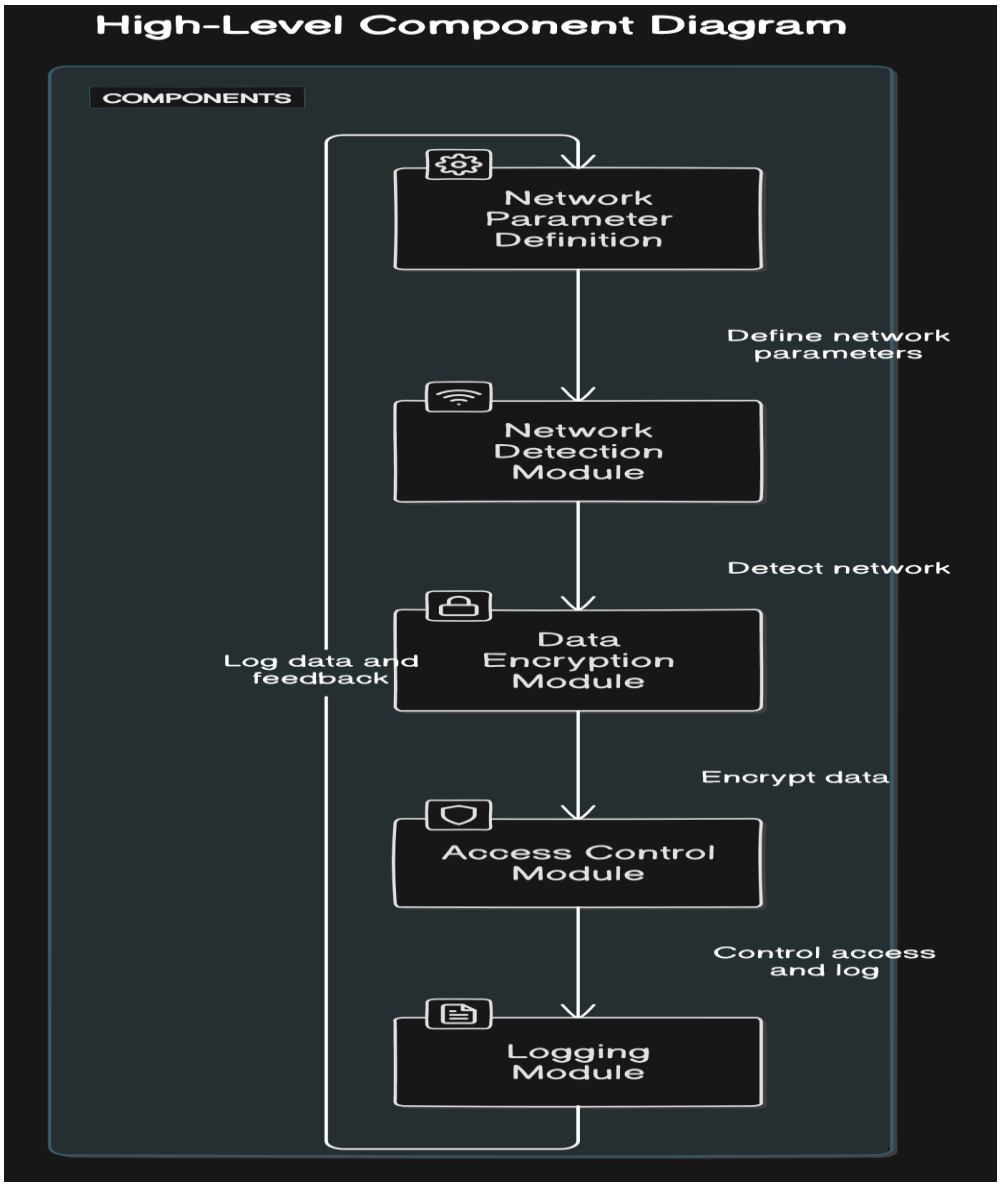

Fig: High level Components

5 Implementation

Thus, the plan for the application of the data decryption tool for the network was purposefully designed to maintain the confidentiality of the user profile data. This entailed the following important steps of development which aimed at improving the general operations and security of the tool. Measures taken were a configuration to provide a secure stand, the start-up of the environment, real time scanning for network parameters, as well as the encryption and decryption of data using advanced encryption methods, authenticating users to deny access, optimal handling of the main application to incorporate all the procedures in harmony and more so, the inclusion of efficient logging and error strategies. Each component served a useful function in containing the decryption of data only within a secure network environment thus minimizing the threats of unauthorized access.

5.1 Configuration steps

The configuration setup phase contained questions in the creation process for a designated config file labelled as ‘config’. Json, that contained the parameters signifying the trusted network. The parameters involved the SSID, MAC address, and the IP range of the trusted network among the parameters. In this configuration file, the blueprint for the decryption tool was laid down and hardcoded, so that even if data decryption was to commence, it would only do so if the current device was connected to the said trusted network. With those critical network parameters safely preserved the indicated tool was ready to authenticate the network environment before initiating any decryption processes, which made the user profile data security several times higher.

5.2 Initialization

The initialization phase entailed creation of conditions for operation of the tool that would implement the model. This step was to set up some required directory and create encryption keys if did not exist on the machine. Another important function lay in the initialization script, which was called to check that all required components including the encryption keys for the sensitive information were set all right and safely stored. This was a great step as the encryption key otherwise known as the decryption key or the algorithm is the backbone to the security process. This means that proper initialization provided a good groundwork on which the environment could be well prepared for subsequent operations of the tool.

5.3 Network Detection

The objective of the network detection was to produce scripts that would establish the current parameters of the network. These scripts were developed specifically to gear with the operating systems that are commonly employed by several devices such as the mobile phone, personal computer, and laptop. The detection process involves; identification of SSID, MAC and IP address range of the connected network. Readily accessible, the tool can compare with the reliable level of network parameters that is in the configuration file. This comparison is important to confirm that the current network is secure before decrypting the data, the operations are done in a secure manner.

5.4 Key Storage Mechanism

One of the measures taken was storing of keys in the hardware security modules or in the files in the privileged mode, so only the user with certain access level could perform the decryption. These keys had to be replaced over time to meet secure management requirements with the keys themselves and data they held; only authorized personnel were allowed to use the keys. This way the usage of the tool meant that not everyone could freely decrypt and hence the data integrity was well protected using the keys.

5.5 Encryption and decryption

A critical function used in the encryption and decryption process was the security of the user profile data. The consumed tool generated user data employing the Fernet encryption approach, which is quite secure and reliable for data storing. Implementations for the

encryption keys were such that its access was controlled, sometimes by storing the keys in special hardware modules. While decrypting it, the tool was to check the current parameters of the network with the trusted ones. That is why only if the client's marks corresponded to the two initial letters would the tool go on to decryption. This made sure that data decryption was only going to happen in a network-protected environment that reduces data vulnerability to other unauthorized parties.

5.6 User authentication

The user authentication phase was adopted as a mean of making sure that only correct users could begin the decryption procedures. This used to involve the establishment of a login system where one had to type in a username and a password. The password given in the method were encrypted and saved safely to avoid unauthorized users gaining access to the website. This step also increased the level of protection because even if the network parameters were the same decryption would only be possible with the right credentials. This double-layered protection boosted up the total protection mechanism of the tool by providing a shield to protect their user profile related information.

5.7 Main Application Logic

The main application orchestration managed the execution of the work of the tool and its dependence on the network detection, user login, and data decryption. After the current network parameters were identified, such parameters were compared to the trusted ones stored in the configuration file. If match was found, then the tool used the ask-for-authentication function to ask the user for authentication. Conditions for exercising the tool were also the successful identification and subsequent decryption of the data. This logical flow made sure that all the security checks were done before execution of data decryption, and this made sure that the user profile data was always secure.

5.8 Logging and Error Handling

The use of logging and error handling mechanisms was again considered relevant for the reliable and secure functioning of the tool. The logging mechanism captured all undertakings executed by the program, such as successful and failed login attempts as well as tests of the network parameters. In the application of exception handling, mechanisms were put in place that could be used when an error occurs, to ensure proper messages that are easy to understand are displayed to users and help them solve the problems. The presented approach that included detailed logging as well as sufficient error handling was helpful in terms of identifying the problem but at the same time kept all the actions traceable, thus enhancing the tool's security and stability.

5.9 Security Measures

The encryption key was written into a file with escalated security level (key. bin), which made it impossible for unauthorized person to access it. To protect this configuration file that contained the trusted networks settings, it was also made read-only. Such measures made it possible to protect the fundamental components of the tool, namely encryption keys and

network parameters. Thus, through these sound security measures, the balance and confidentiality of the data decryption process were maintained to allow a safe process of the user's profile data.

6 Evaluation

The assessment of the network-specific data decryption tool entailed functional testing to demonstrate the tool's accuracy, speed, and safety. The first of them was to ensure that, depending on the network parameters and user authorization, the tool enciphers and deciphers the data securing their vulnerability irrespective of the network environment. The primary aim of the evaluation process was to determine how effectively the tool enforces decryption only when the data is in a secure network. The evaluation focused on several key functionalities: They include login check, save data check, retrieve data check, file upload check, download file check, and logout. The list of tests comprised functional, performance, and security tests as a part of the evaluation methodology. The basic functionalities were evaluated for each of the tool in isolation to establish its performance. The tool was evaluated on a proxy environment with trusted network parameters and fake user id and passwords.

6.1 Functional Testing

6.1.1 Login Check

The tool was used to confirm that the specified method of authenticating user worked as expected. Users had to input a username and password that were then hashed comparisons were then made to the stored hashes. The tool periodically scanned for network parameters when connecting the device to the network, to confirm that the device was on the trusted network. If, for some reason, the network parameters were matching, then the user was granted permission to go through the next stage and was considered to have been authenticated.

6.1.2 Save Data Check

The functionality of using the above-mentioned encryption method to save the data was evaluated by saving the user profile data in an encrypted format. The tool also encrypted the data with what is known as the Fernet encryption and safely stored it. During the save operation the tool performed network parameters check confirming that the device connected to the know trusted network. If a network change was noticed, then the tool moved errand and re-encrypted the data to avoid any informed break-ins.

6.1.3 Retrieve Data Check

To evaluate the data retrieval technique, the stored user profile data was encrypted, and differential data retrieval approach was applied. The tool examined the state of the network parameters before conducting decryption. When the current network was in match with any of the trusted parameters then the data was decrypted and retrieved from the database. Originally, if the network parameters did not match it was an indication that everything must be stopped and the data must not be decrypted, thus implementing an elevated level of security.

6.1.4 File Upload Check

The tool was evaluated for its capability of file upload which is a security test. Within the framework of the tested tool, it was possible to check the network parameters during the uploading process. If the device was on the trusted network, then the file was uploaded safely. They mentioned that any change in the network made the tool to encrypt the file to make sure that the file did not pop up during the uploading.

6.1.5 Download File Check

The process of file download was evaluated in the same way, namely its security, like the test of the tool's upload feature. A version of the tool questioned the network parameters before making the download and ensuring that the connected device was on the trusted network. Without a doubt if the network matched the trusted configuration the file was decrypted and downloaded. Any change in the network led to re-encryption of the mere files to enhance the protection of the files.

6.1.6 Logout functionality check

The logout functionality was also exercised to show how the user's session was being closed appropriately. On logging out, the tool assured that all data that was input, and/or edited were securely encrypted. During the session, the network parameters were examined and altering of one of them performs an automatic re-encryption of the data.

6.2 Discussions

The goal of integrating the data decryption tool particular to the network was achieved, as the software raised the protection level up to the required mark if decryption only took place in a protected network. About the effective management of security measures and conforming it to the parameters of the network, this project responded well to such a significant problem. The implementation also showed that if the decryption of a specific data was to be made dependent upon the context and network specific factors, then it would effectively be impossible to breach through the said network.

Some of the goals that were aimed to be achieved by this implementation were the smooth integration of network detection, the users' identification as well as using strong encryption methods. The tool was designed with the function of identifying the network parameters in real-time to compare them against the trusted configurations. This functionality was very important to ensure that decryption of sensitive data is limited to certain coding networks to avoid loss of the user profile data. These objectives were attained during the implementation process, and the tool that emerged offered a solid security feature while preserving the privacy and contents of data belonging and utilised by the user.

However, the process of development did not pass without difficulties. An important concern marked was the fact that there was occasional challenge of achieving accurate identification of the network parameters on the various operating systems. The final part of the project was spent to refine the features of the network detection scripts and to make sure it was suitable to

various environments by conducting several tests to capture the several network parameters in real-time. Finally, the aspect of key management came out strongly concerning the security framework into place. Where the biometric solutions provided means for accessing information in cases where there was a security breach, then it was mandatory to ensure that the keys to the encryption were created and preserved to allow only personnel with an upper hand in-security department access. This requirement demanded the creation of a new method for strict management of keys to reduce the threats related to key leakage.

The appreciable aspects of the tool whereby its effectiveness was demonstrated pertained to structural and dynamic functions. The dynamic network detection worked well in confirming that the current network version met with the stored security configuration file. This type of verification was crucial for preservation of the decryption process's efficacy in terms of maintaining the data's safety at all stages. In addition, the use of a two-layer security where there was network check-up and user identification were in place, served as an extra precaution against intrusion. This made the protection of sensitive information robust because of the various levels of protection developed through this layered approach.

With such successes, however, it is important to note that the tool reviewed in the current paper also had some limitations. Due to its focus on certain network parameters, fluctuations in the network settings of the specific network called for changes in their configuration file as well. This dependency was somewhat problematic when working in highly mobile networks, where the characteristic parameters might change quite often. Also, the current version of the tool was not designed to be used in a dynamic network environment where network conditions could change from time to time or degenerate. In such environments, possibly, the efficiency of the tool could be less consistent, thus calling for enhancement.

As for the further research, more specific features of networks, such as more technical properties of the process and the possibilities of its application, could be investigated, or integration of machine learning approaches to network detection and verification could be considered. This feature when applied might have the capability to introduce learning into different network behaviours thus improving the element of flexibility and robustness in different environments on the tool. Also, future work could be handling with the enlargement of the area relevant to the tool application and the expansion of the possibilities for the network settings.

The application of this tool had a lot of practical benefits especially in the cases where some information is to be transferred through several networks securely. Situations where it might be used routinely are in business organizations, schools, and work-from-home situations where information security is a concern. Effectively what the tool did was, ensure that decryption could only take place in the secure domain of a trusted organization's network; thus, giving organizations a potent method of protecting their information from potential intruders.

6.3 Tools and Validity

Development of the tool occurred in Python, a high-level language with proven reliability and flexibility in software applications. Coding, debugging, and testing used were done by the trusted and popular development environment known as Visual Studio Code to make sure that the tool is accurate and efficient.

7 Conclusions and future work

With the successful creation and use of the network-specific data decryption tool, a new level has been achieved in the field of data safety, especially in the cases of critical and confidential data security within the trusted networks. In this manner, the tool adequately prevents unauthorized decryption and thus delivers a suitable solution to safeguard the user profile data. Due to the incorporation of the dynamic network detection, the user authentication and strong encryption techniques the objectives of the project were met, and currently the tool consistently ensures the confidentiality and integrity of the user's profile data. In this way, the tool can hardly be used by unauthorized persons and thus is a brilliant solution for organizations that need tight security for their data. The benefits of the tool are reflected in dynamic check of the network parameters and stringent control over the access to the data that would be impossible to deliver in the highly changeable and complex network settings. The active approach to security and the service's integration of the critical tools enables it to address the modern security issues with the multifaceted protection of data. The possibility of further evolution of the tool with the incorporation of such features as machine learning for adaptive network verification points to the fact that the tool is only set to become more effective in the future making it a very important weapon to have in securing data due to today's development of the digital world.

Further development of the current research on this project could work towards increasing the portability and generalizability of the specified network-specific data decryption tool by implementing more sophisticated and complex methodologies related to the analysis of the network's behavior and any anomalies. This would enable the tool to have the ability to respond to changes and adapt in the environment of the network and make the desired application more dependable and efficient in the cases that are needed. Moreover, if the tool's capabilities could be generalized to include the realistic simulation of other network scenarios as well as different operating systems, it would prove even more beneficial to organizations of all kinds. Research could also focus on the creation of a cleaner more efficient and easily navigated GUI as well as the use of more advanced features for store and/or creating keys among which may be integration of Azure Key Vault and application of blockchain technology. Such developments would not only improve the functionality of the tool, practicality, and flexibility of its application, but would also expand a tool usage scenario, making it more suitable for use in intensely secure locations, increasing its capacity for solving the current problems of data protection.

8 References

1. L. M. Shamala, G. Zayaraz, K. Vivekanandan, and V. Vijayalakshmi, "Lightweight cryptography algorithms for the internet of things enabled networks: An overview," J. Phys. Conf. Ser., vol. 1717, no. 1, 2021, doi: 10.1088/1742-6596/1717/1/012072.
2. A. Beloglazov and R. Buyya, "Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds," Concurrency and Computation: Practice and Experience, vol. 27, no. 5, pp. 1310-1333, 2015.
3. R. Kune, P. Konugurthi, A. Agarwal, C. R. Rao, and R. Buyya, "The anatomy of big data computing," Softw. Pract. Exper., vol. 46, no. 1, pp. 79-105, 2016.
4. A. Sevin, A. Ahmed, and O. Mohammed, "A survey on software implementation of lightweight block ciphers for IoT devices," J. Ambient Intell. Humaniz. Comput., no. 0123456789, pp. 1-11, 2021, doi: 10.1007/s12652-021-03395-3.
5. J. Sherry et al., "Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service," ACM SIGCOMM Computer Commun. Review, vol. 42, no. 4, pp. 13–24, 2012.
6. C.-Y. Chow, M. Mokbel, and T. He, "A privacy-preserving location monitoring system for wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 10, pp. 94-107, 2011.
7. H. Hamed, E. Al-Shaer, and W. Marrero, "Modeling and Verification of IPSec and VPN Security Policies," Proc. IEEE ICNP '05, Nov. 2005.
8. M. Amini, H. Sadreazami, M. O. Ahmad, and M. N. S. Swamy, "Multichannel color image watermark detection utilizing vector-based hidden Markov model," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2017, pp. 1–4.
9. H. Liao, P. Lee, Y. Chao, and C. Chen, "A Location-Dependent Data Encryption Approach for Enhancing Mobile Information System Security," in the 9th International Conference on Advanced Communicate Technology, pp. 625-626, Feb. 2007.
10. S. Kang, B. Veeravalli, and K. M. M. Aung, "ESPRESSO: An Encryption as a Service for Cloud Storage Systems," in Proc. Int. Conf. Autonomous Infrastructure, Management, and Security (AIMS), June 2014.
11. D. Hu, D. Zhao, and S. Zheng, "A new robust approach for reversible database watermarking with distortion control," IEEE Trans. Knowl. Data Eng., vol. 31, no. 6, pp. 1024–1037, Jun. 2018.
12. J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," IEEE Trans. Inf. Forensics Security, vol. 12, no. 10, pp. 2402–2415, Oct. 2017.
13. A. Khoshgozaran and C. Shahabi, "Blind evaluation of location-based queries using space transformation to preserve location privacy," *International Journal of Computational Science and Engineering*, 2007.
14. E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," Internet Engineering Task Force (IETF), 2018.
15. J. Alshehri and A. Alhamed, "A review paper for the role of cryptography in network security," in 2022 4th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), 2022, pp. 1-5. DOI: 10.1109/ICECIE55199.2022.10000338.
16. M.S. Akter, J. Rodriguez-Cardenas, H. Shahriar, A. Cuzzocrea, and F. Wu, "Quantum cryptography for enhanced network security: A comprehensive survey of research,

developments, and future directions," in 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 5408-5417. DOI: 10.1109/BigData55660.2023.10100324.
17. L. Zhou and C. Liu, "The improvement of data encryption technology in computer network security," in 2022 International Conference on Artificial Intelligence in Everything (AIE), 2022, pp. 465-470. DOI: 10.1109/AIE57029.2022.00095.
18. C.J. D'Orazio, K.K.R. Choo, and L.T. Yang, "Data exfiltration from Internet of Things devices: iOS devices as case studies," IEEE Internet of Things Journal, vol. 4, no. 2, pp. 524-535, 2016. DOI: 10.1109/JIOT.2016.2613983.
19. J. Mongay Batalla, "Featured Papers on Network Security and Privacy," Journal of Sensor and Actuator Networks, vol. 13, no. 1, p. 11, 2024. DOI: 10.3390/jsan13010011.
20. A.K. Sharma and S.K. Mittal, "Cryptography & network security hash function applications, attacks and advances: A review," in 2019 Third International Conference on Inventive Systems and Control (ICISC), 2019, pp. 177-188. DOI: 10.1109/ICISC.2019.9036172.