VISION:

To be a centre of excellence in Education and Research in the field of Computer Science and Engineering by empowering the students to be highly competent, technologically proficient, self-motivated, innovative professionals, entrepreneurs and responsible global citizens possessing human values to meet the global challenges.

MISSION:

M1: To impact the students with strong fundamental concepts, analytical capability, and problem-solving skills, thereby enhancing the employability skills required for the industries.
M2: To provide a conducive teaching learning, and research environment through faculty training, self-learning and sound academic practices to carry out research with reputed research institutes and industries.
M3: To train the students to become the most sought-after professionals by providing them the opportunities to promote organizational and leadership skills in students through various extracurricular and co-curricular events.
M4: To include the qualities of leadership and entrepreneurship with good human values and professional ethics to become good citizens and serve society.

ABSTRACT:

In today's society, the security of our personal information is a major concern. With the increasing use of technology in our daily lives, we are constantly sharing and using data that may not be entirely private. Hackers and scammers are always on the lookout for vulnerabilities in the network, making it difficult to ensure that our data is secure. Despite the many security measures in place, they are often not up to date and cannot keep up with the speed at which the network is growing. This puts many applications that require security at risk. To address this issue, we propose to develop more efficient techniques that can prevent data leaks and provide better data security. Our project aims to deliver solutions to the current challenges faced by users in terms of data security. We will work towards bridging the gap between the existing security measures and the current network requirements. By implementing advanced algorithms and techniques, we hope to provide better protection for personal data and ensure that it remains secure.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In today's world, we can accomplish various tasks with just a single click, thanks to the advancements in technology. People use their smartphones for various day-to-day tasks and share different media with the world. However, with these new revolutions, the need for data security has also increased. Cryptography and Steganography are two of the main techniques that offer this security[2]. Cryptography involves encrypting the data to make it unreadable, while Steganography is used to conceal the data into a cover file, hiding its existence. Various existing systems provide different Steganography-based applications, each implemented with different algorithms for different uses. These systems use different types of cover media to conceal data, and the data to be hidden varies from technique to technique. Some of the existing applications that help in securing data through Steganography are the Stego App[2], MobiStego, MoBiSiS, and Smart Steg[3].

Steganography is a fascinating process that involves hiding secret information within an unsuspecting graphic. The word "steganos" comes from the Greek language, meaning "hidden" or "covered", while "graph" means "to write". Although Steganography techniques are advantageous to use, they have a few limitations. Our project introduces an Android application called "StegMate" that utilizes various types of steganography, such as text, video, audio, and network, to offer a more comprehensive approach to data security. The application ensures that any confidential message is only accessible to its intended recipients.

1.1 Problem Statement

The objective is to encode various types of secret media, such as images, text, and audio, into a cover file that can be an image, audio, or video. The produced stego-media, which is a media file with a secret message embedded within, can be shared with others. The recipients should be able to decode any of the stego-media that is shared with them. The system aims to reduce the load and increase efficiency during encoding/decoding, as well as reduce the time required to generate the stego-media. Additionally, it should be adaptable and integrated with new and emerging technologies.

The project's main objective is to develop an optimal technique for generating embedded media that can adapt to all types of extensions.

1.2 Existing System

In today's world, data security is of utmost importance. Many contenders, such as WhatsApp, fulfil this need by providing end-to-end encryption for all messages sent. Additionally, WhatsApp offers a feature called "once-view" where media can be viewed only once before disappearing. The SmartSteg Application uses the BMP image format, which is lossless, and the LSB algorithm for encryption[3]. Another application uses the RSA Algorithm and LSB Insertion to achieve Image Steganography[4]. The Stego App allows for embedding both text and images within an image[2]. Some applications also reduce the size of the stego-image to be MMS compatible, making it easier to share[1].

1.3 Disadvantages of Existing System

It is worth noting that while there is a systematic approach to embedding data into images, many applications have limited functionality that is restricted to their app. For instance, the widely used LSB insertion method only supports embedding into images and does not support multiple extension types. Additionally, some applications are outdated and do not support current image specifications, limiting their use to images with lower size and encoding. As a result, the usage of these apps has significantly decreased, and the current image-embedding apps are not multi-platformed.

1.4 Proposed System

Our proposed technique aims to provide users with a seamless way to embed various types of media, such as text, images, and audio, into a cover file. This cover file can support any extension for images or audio, making it a versatile solution for users. By reducing system resource consumption, we aim to increase the efficiency of the application, allowing users to perform tasks faster and more effectively. To achieve this, we plan to incorporate a mechanism for file upload and live media capture, enabling users to easily add new content to their cover files. Additionally, we will support file sharing, allowing users to share their cover files with others. To ensure the security of the application, we will implement robust security measures that prevent unauthorized access. This will ensure that only the user can access the application and their cover file, providing peace of mind and security.

CHAPTER 2

LITERATURE SURVEY

Data security is a very important aspect in today's world. Whilst all the systems are getting interconnected, securing the shared data is also important. There exist various challenges in the transmission of data. The term security has a very vast explanation.

Intellipaat[5] defines it as the protections put in place to secure databases, websites, and computers against unwanted access. This procedure also includes safeguards against data loss or corruption. Small firms and large enterprises should implement protection measures to secure data. Failure to have an effective information security plan in place can have severe repercussions for companies. According to the *Economist,* we are currently living in the "zettabyte age." For the amount of traffic being generated, the most suitable step is to manage and safeguard the transfer of sensitive or personal information at every known location. Both proactive enterprises and global regulators are progressing significantly.

IBM[6] states that Data security is the practice of protecting digital information from unauthorized access, corruption or theft throughout its entire lifecycle. It's a concept that encompasses every aspect of information security from the physical security of hardware and storage devices to administrative and access controls, as well as the logical security of software applications. It also includes organizational policies and procedures. Data security is the practice of protecting digital information from unauthorized access, corruption or theft throughout its entire lifecycle. It's a concept that encompasses every aspect of information security from the physical security of hardware and storage devices to administrative and access controls, as well as the logical security of software applications. It also includes organizational policies and procedures.

Medium[7] - In the current digital era, data privacy and security have emerged as two pressing challenges. Data privacy and security are getting more complicated and contentious as more and more personal data is being gathered and analyzed. Personal data collection and storage have become simpler thanks to the digital age, but data theft has also become simpler for cyber criminals. Personal data is being gathered and processed at a rate that has never been seen before as our lives become more and more digital. Organizations are gathering a ton of personal information about people from social media platforms and e-commerce sites to healthcare providers and financial institutions. Although this information can be extremely useful for research, marketing, and other purposes, it also poses serious hazards to the security and privacy of the people who use it.

The New York Times[8] – Privacy can never be guaranteed with absolute certainty. The risks should always be minimized, and balanced against the benefits of the innovations that may arise from increased data availability. A better program would target aid to business sectors and geographies that most need help, using real-time data from the businesses themselves. This data already exists, but only behind company walls. It should be anonymized as carefully as possible and assembled for public use so that local policymakers and entrepreneurs can direct the relief to those who need it most.

The Journalist's Resource[9] – On your smartphone, you're not much more than a data machine, generating reams of valuable information that tech companies can mine for insights, sell to advertisers and use to optimize their products. But it's easy to forget these risks to personal privacy and security while tapping out messages to friends or scrolling endlessly through the web. The distraction machines at our fingertips ask for access and we give it up quickly, hastily agreeing to unread privacy policies and terms of service in exchange for a fresh jolt of content.

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATION

Android [4]is undoubtedly the most widely used operating system in the smartphone industry. In this study, the proposed system was implemented using the Android SDK (Software Development Kit) and Android Studio IDE (Integrated Development Environment) as the primary development tools. The fundamental principle behind the development of this application is to make it available on all devices that run the Android operating system. The Android SDK and Android Studio provide the necessary functions to achieve this goal. The graphical user interface design was done using Java FX, which allows for rich interface and graphical designs. This ensures that the application is user-friendly and visually appealing, making it easier for users to navigate and use the application effectively. The encryption and decryption algorithms are based on RSA, AES, and DES algorithms whereas the messages were embedded using the least significant bit (LSB) insertion algorithm. The source code for the application was developed using Android SDK supported by the Android NDK (Native Development Kit) and was run and compiled on the Android Studio IDE. The requirement that the system needs is categorized into functional and non-functional requirements. These requirements are listed below:

3.1 FUNCTIONAL REQUIREMENTS

These are the statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. Physical equipment, such as servers, personal computers, cabling, network and clustering switches, backup devices, storage devices, and disaster recovery devices, is required for implementation. In some cases, the functional requirements may also explicitly state what the system should not do.

The major functional requirements in the project are:

• The system must allow a user to take a picture with the camera or upload an image from the phone or device gallery.

• The system should successfully allow a user to generate the keys for encryption.

• The system should allow a user to perform the process of encryption using the public key generated.

• The system should allow a successful transfer of keys between the sender and the receiver.

• The application is expected to allow the embedding of the encrypted message in the image.

• The application should allow retrieval of information from the image at the receiver's end.

• The system should necessarily allow a user to decrypt encrypted messages using the private key generated by the application itself.

3.2 NON-FUNCTIONAL REQUIREMENTS

1. Security: The app should employ robust encryption algorithms to ensure that the hidden data remains confidential and cannot be easily deciphered by unauthorized individuals. It should also implement techniques to prevent detection of the hidden data, such as using advanced steganographic methods and disguising the hidden data within innocuous files.

2. Performance: The app should be optimized to minimize its impact on system resources, such as CPU and memory usage. It should be designed to efficiently handle the hiding and extraction of data, ensuring that the process is fast and does not cause significant delays or slowdowns.

3. User Interface: The app should have an intuitive and user-friendly interface, with clear instructions and controls that allow users to easily perform steganography operations. It should provide a seamless user experience, guiding users through the process and providing feedback on the status of their operations.

4. Compatibility: The app should be compatible with different operating systems (such as Windows, macOS, and Linux) and devices (such as desktops, laptops, and mobile devices). It should support various file formats for hiding and extracting data, ensuring that users can work with a wide range of files.

5. Reliability: The app should be stable and reliable, ensuring that the hidden data can be successfully extracted without any loss or corruption. It should handle errors gracefully and provide appropriate error messages to users in case of any issues.

6. Scalability: The app should be designed to handle a large volume of data, allowing users to hide and extract data from files of varying sizes. It should also support different types of media files, such as images, audio, and video, enabling users to hide data in various formats.

7. Documentation: The app should provide comprehensive documentation, including user guides and technical manuals, to assist users in understanding and utilizing its features. The documentation should cover installation instructions, usage guidelines, troubleshooting tips, and any other relevant information.

8. Privacy: The app should respect user privacy and not collect or store any personal or sensitive information without the explicit consent of the user. It should adhere to privacy regulations and best practices to ensure the protection of user data.

9. Accessibility: The app should be designed to be accessible to users with disabilities, following accessibility guidelines such as providing alternative text for images, keyboard navigation support, and compatibility with assistive technologies. It should strive to provide an inclusive user experience for all users.

10. Maintenance and Support: The app should have a dedicated support system in place to address user queries, provide updates, and fix any issues that may arise. It should have a mechanism for users to report bugs or suggest improvements and the development team should actively maintain and update the app to ensure its continued functionality and security.

3.3 MINIMUM HARDWARE AND SOFTWARE REQUIREMENTS FOR DEVELOPMENT

This outlines the minimum software and hardware requirements for deploying matter most. Requirements may vary based on utilization and observing the performance of pilot projects is recommended before scale-out. Usage of CPU, RAM, and storage space can vary significantly based on user behaviour. These hardware recommendations are based on traditional deployments and may grow or shrink depending on how active your users are. Moreover, memory requirements can be driven by peak file-sharing activity. The recommendation is based on the default 50 MB maximum file size, which can be adjusted from the System Console. Changing this number may change memory requirements. While this may be more specific to individual graphic programming using IDE, it's still a fantastic idea to update graphics cards as well. Below is an overview of combinations of specifications or server configurations that are commonly used.

Operating System: Android 8.0 or higher

Processor: ARMv8, ARM64, or x86_64 processor

RAM: At least 2 GB

Storage: At least 100 MB of available storage

The software requirements:

1. Operating System: Android Studio is compatible with Windows, macOS, and Linux operating systems. Ensure that your development machine meets the requirements of the chosen operating system.

2. Java Development Kit (JDK): Android Studio requires JDK to be installed on your development machine. Make sure you have the latest version of JDK installed.

3. Android Studio: Download and install the latest version of Android Studio from the official website. Android Studio provides the necessary tools and environment for Android app development.

4. Android SDK: Android Studio comes bundled with the Android Software Development Kit (SDK). The SDK contains libraries, tools, and resources required for developing Android apps. Make sure to install the necessary SDK components for the targeted Android versions.

5. Gradle: Android Studio uses Gradle as the build system for managing dependencies and building the app. Ensure that you have the compatible version of Gradle installed.

6. Integrated Development Environment (IDE): Android Studio provides a powerful IDE for developing Android apps. Familiarize yourself with the features and functionalities of the IDE to enhance your development experience.

7. Emulator or Physical Device: To test and run your steganography app, you will need either an Android emulator or a physical Android device. Android Studio provides an emulator for testing, but using a physical device is recommended for better performance and real-world testing.

8. Version Control System: It is recommended to use a version control system like Git to manage your source code and collaborate with other developers if needed.

9. Additional Libraries and Dependencies: Depending on the specific requirements of your steganography app, you may need to include additional libraries or dependencies. Research and identify the appropriate libraries for steganography operations and integrate them into your project.

10. Documentation and Resources: Android Studio provides extensive documentation and resources to help you with Android app development. Make use of official documentation, online tutorials, and community forums to enhance your understanding and troubleshoot any issues you may encounter during development.

CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

4.1 USE CASE DIAGRAM

A use case is a description of how a user interacts with a software application to achieve a particular goal. In an Android application, use cases can be categorized into different sections, such as user registration, login, profile management, data security, and so on. Each use case has its specific requirements and functionalities that need to be implemented to ensure the smooth functioning of the application. To develop an efficient Android application, it is essential to identify and define the use cases and their requirements. Proper planning and implementation of each use case can lead to a user-friendly and high-performing Android application.

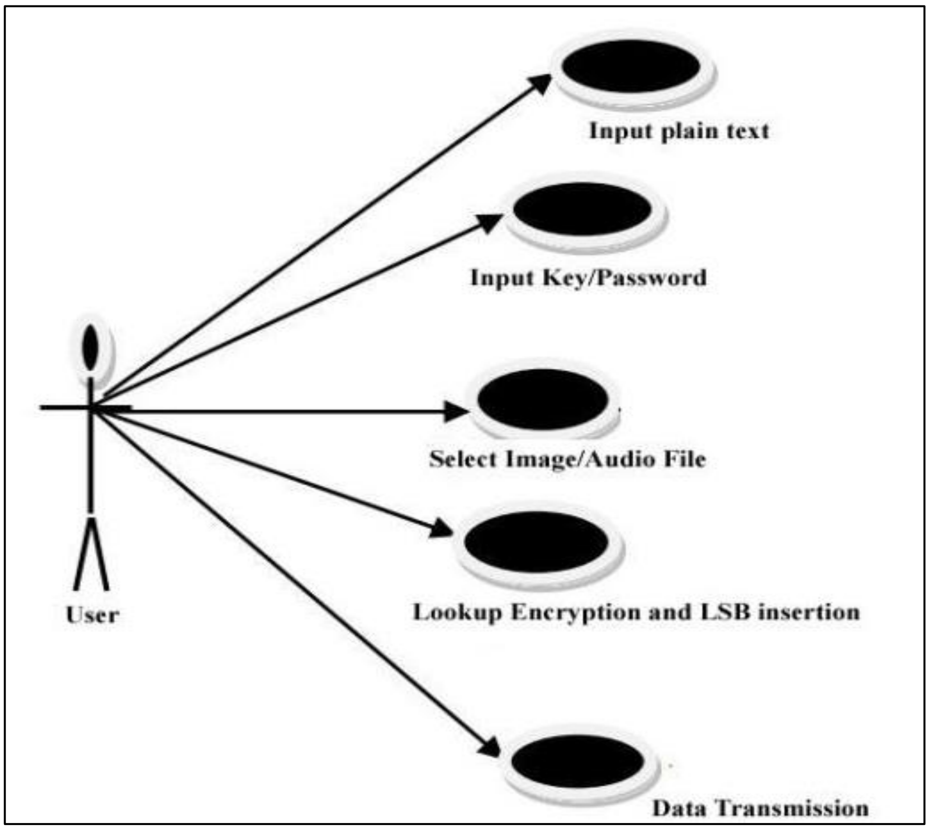

Fig 1. Use-case Diagram

A use case in Android refers to a specific interaction or scenario between a user and an Android application. It describes how a user interacts with the application to accomplish a specific task or goal. Some common use cases in Android applications include:

1. User registration: This use case involves the process of creating a new account for the user. It typically requires the user to provide their name, email address, and password.

2. Login: This use case involves the process of logging in to the application using the user's credentials. The user must enter their email address and password to gain access to their account.

3. Profile management: This use case involves the management of the user's profile information, such as updating their name, email address, password, and other settings.

4. Data security: This use case involves implementing security features such as encryption, authentication, and authorization to protect the user's data from unauthorized access.

5. Push notifications: This use case involves sending push notifications to the user to notify them of important events or updates.

6. Social media integration: This use case involves integrating the application with popular social media platforms like Facebook, Twitter, and Instagram to enable users to share content and interact with their friends.

Proper identification and implementation of each use case are essential for developing a user-friendly and high-performing Android application.

CHAPTER 5

METHODOLOGY

5.1 ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for developing Android applications. It was developed by Google and is built on JetBrains' IntelliJ IDEA software. The software is specifically designed for Android development and can be downloaded on Windows, macOS, and Linux-based operating systems. It has replaced the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Google announced Android Studio on May 16, 2013, at the Google I/O conference. Initially, it was released as an early access preview stage starting from version 0.1 in May 2013. Later on, it entered the beta stage starting from version 0.8, which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. By the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development. On May 7, 2019, Google announced Kotlin as its preferred language for Android app development, replacing Java. Java is still supported, as is C++.

Android Studio has several features, such as a rich layout editor that allows users to drag and drop UI components and preview layouts on multiple screen configurations. It also supports building Android Wear apps and has built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine. The Android Virtual Device (Emulator) allows users to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) such as Java, C++, and more with extensions, such as Go. Android Studio 3.0 or later supports Kotlin, and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer content policy.

Android is one of the most widely used mobile operating systems in the world. It offers a user-friendly interface, customizable features, and an open-source platform, making it a popular choice for mobile app development. Android app development has several advantages over other platforms. For starters, Android has a vast user base, which means that there is a high demand for Android apps. Additionally, Android apps can be developed using a wide range of programming languages, including Java, Kotlin, and C++, offering developers the flexibility to choose the language they are most comfortable with. Furthermore, Android provides a comprehensive set of tools and libraries that make the app development process more efficient. Overall, Android app development is an excellent choice for building robust, scalable, and user-friendly mobile applications.

Our project aims to identify existing applications and test them to build more efficient, bug-free software that can be incorporated into various industrial solutions.

5.1.1 ANDROID DEBUG BRIDGE(ADB)

Android Debug Bridge (adb) is a command-line tool that allows you to communicate with a device. The tool provides access to a Unix shell, which you can use to run various commands on the device. With adb, you can perform a range of device actions, such as installing and debugging apps.

The adb program is a client-server program that consists of three components:

• A **client**, which sends commands. The client runs on your development machine, and you can invoke it from a command-line terminal by issuing an adb command.

• A **daemon** (adbd), which runs commands on a device. The daemon runs as a background process on each device.

• A **server**, which manages communication between the client and the daemon. The server runs as a background process on your development machine.

5.1.2 GRADLE

Gradle is a build automation tool that is open source and emphasizes flexibility and performance. To put it simply, it is a tool that automates the process of building an application. Gradle is also the official build tool for Android. We can observe the process of

"Gradle Build Running" in our Integrated Development Environments (IDEs) whenever we attempt to run the developed code in an emulator or an actual device via USB.

Plugins for Gradle

Gradle is a build automation tool for Android Developers, but it cannot automate the entire build process on its own. That's why its definition is centred around "Flexibility". To use third-party libraries in our project, we must add them as dependencies in our app-level Gradle file. We can only use their classes after syncing the Gradle with the project. Gradle's flexibility is achieved through its "Plug-in" feature, which allows you to browse and download all available Gradle plugins.

5.1.3 JDK

The **Java Development Kit (JDK)** is a technology distribution created by Oracle Corporation. It adheres to the Java Language Specification (JLS) and the Java Virtual Machine Specification (JVMS) and provides the Java Application Programming Interface (API) Standard Edition (SE). It is based on the OpenJDK, which is a community-driven project that Oracle oversees. The JDK provides various software tools for working with Java applications. Some of these tools include a virtual machine, a compiler, performance monitoring utilities, a debugger, and other useful software for Java programmers.

Oracle has released the current version of the software under the Oracle No-Fee Terms and Conditions (NFTC) license. The company has made binaries available for Windows, macOS, and Linux-based operating systems for the x86-64 architecture, and for macOS and Linux for the aarch64 architecture. Previous versions supported the Oracle Solaris operating system and SPARC architecture. Oracle's main implementation of the JVMS is known as the HotSpot (virtual machine).

The JDK (Java Development Kit) contains various tools used for developing Java applications. These tools include:

- Applet viewer: a tool for running and debugging Java applets without using a web browser
- apt: the annotation-processing tool
- extcheck: a utility that detects conflicts between JAR files
- IDLJ: the IDL-to-Java compiler, which generates Java bindings from a Java IDL file
- JAB switch: the Java Access Bridge, which exposes assistive technologies on Microsoft Windows systems

- java: the loader for Java applications, which interprets class files generated by the Javac compiler. It replaces the old deployment launcher, jre, which is no longer included in Sun JDK.
- javac: the Java compiler, which converts source code into Java bytecode
- javadoc: the documentation generator, which automatically generates documentation from source code comments
- jar: the archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.

5.2 PROGRAMMING LANGUAGES

5.2.1 JAVA

Java is a programming language that is designed to have as few dependencies as possible. It is object-oriented, class-based, and high-level. The language is intended to be general-purpose so that programmers can write code once and run it anywhere (WORA). This means that compiled Java code can run on all platforms that support Java without recompiling. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.

Java applications are typically compiled to bytecode, which can run on any Java virtual machine (JVM), regardless of the underlying computer architecture. Additionally, the Java runtime provides dynamic capabilities like reflection and runtime code modification that are typically not available in traditional compiled languages.

Java has been a very popular programming language since its release, and it was the third most popular programming language in 2022, according to GitHub. However, there has been a gradual decline in the use of Java in recent years, with other languages using JVM gaining popularity. As of October 2023, Java is ranked fourth on the TIOBE index.

James Gosling developed Java at Sun Microsystems in May 1995 as a core component of the Java platform. The original and reference implementation of Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, most of Sun's Java technologies have been relicensed under the GPL-2.0-only license in compliance with the specifications of the Java Community Process. Oracle offers its own HotSpot Java Virtual Machine, but the official reference implementation is the OpenJDK JVM, which is free open-source software used by most developers. It is also the default JVM for almost all Linux distributions.

As of September 2023, Java 21 is the latest version, which is also a long-term support (LTS) version. Java 8, 11, and 17 are previous LTS versions that are still officially supported.

5.2.2 XML

XML or **Extensible Markup Language** is a file format and markup language for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a way that is both human-readable and machine-readable. XML is an open standard that was defined by the World Wide Web Consortium's XML 1.0 Specification in 1998. The language is designed to be simple, general, and usable across the Internet. It is a text-based data format that supports different human languages through Unicode. XML is not limited to documents, as it is widely used for representing arbitrary data structures used in web services. Many schema systems exist to define XML-based languages, and programmers have developed many APIs to aid the processing of XML data.

XML is commonly used to exchange data over the Internet. Hundreds of document formats using XML syntax have been developed, including RSS, Atom, Office Open XML, OpenDocument, SVG, COLLADA, and XHTML. XML also provides the base language for communication protocols like SOAP and XMPP. It is the message exchange format for the Asynchronous JavaScript and XML (AJAX) programming technique.

Many industry data standards like Health Level 7, OpenTravel Alliance, FpML, MISMO, and National Information Exchange Model are based on XML and the rich features of the XML schema specification. In publishing, Darwin Information Typing Architecture is an XML industry data standard. XML is used extensively to underpin various publishing formats.

BIBLIOGRAPHY

[1] Zavrak, Sultan, Seyhmus Yilmaz, and Huseyin Bodur. "An Implementation of Android-based Steganography Application." International Journal of Computer Science and Information Security 13.12 (2015): 73.

[2] Ullah, Azmat, and Mohsin Ijaz. "Stego App: Android-based Image Steganography Application using LSB Algorithm." Int. Res. J. Eng. Technol 5.9 (2018): 862-865.

[3] Bucerzan, Dominic, Crina Ratiu, and Misu-Jan Manolescu. "SmartSteg: A new android-based steganography application." Int. J. Comput. Commun. Control 8.5 (2013): 681-688.

[4] Apau, Richard, and Clement Adomako. "Design of image steganography based on RSA algorithm and LSB insertion for Android smartphones." International Journal of Computer Applications 164.1 (2017): 0975-8887.

[5] The Importance of Data Security in 2023 - Intellipaat

[6] What is Data Security? Data Security Definition and Overview | IBM

[7] Data Privacy and Security: The Importance of Protecting Personal Information in the Digital Age | by Ejike Uchenna Splendor | Medium

[8] Balancing Privacy With Data Sharing for the Public Good - The New York Times (nytimes.com)

[9] Data security: Research on privacy in the digital age (journalistsresource.org)