**Institute for Data Engineering**
Module Advanced Internet Computing – Lab Exercises

**Project Topic B – Cloud-based Secure Data Storage and Sharing System**

# Introduction

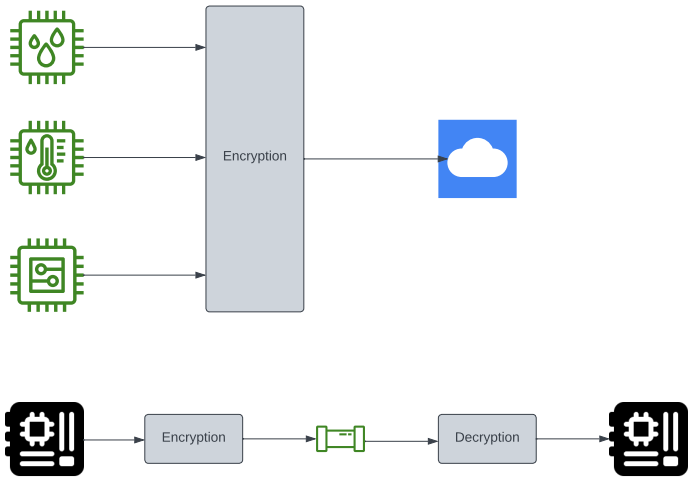


Figure 1: High-level overview of the system

Cryptography is the practice of securing communication from unauthorized access by converting the original message into a form that is unintelligible to anyone except the intended recipient. It is an essential part of modern communication systems, including Internet security, e-commerce, and digital payments. The fundamental principles of cryptography involve *encryption*, *authentication*, *non-repudiation*, and *integrity*.

- **Encryption** is one of the fundamental principles of cryptography. It involves the process of converting plaintext (i.e., unencrypted data) into ciphertext (i.e., encrypted data) using an algorithm and a secret key. The ciphertext can only be decrypted back into plaintext by someone who has access to the secret key. Encryption is widely used to protect sensitive information such as financial data, personal information, and confidential documents.
- **Authentication** is the process of verifying the identity of a user or system. This involves providing evidence of the user's identity to the recipient to ensure that the sender of a message is who they claim to be. This is typically achieved through the use of digital certificates, which are electronic documents that provide proof of identity.
- **Non-repudiation** ensures that the sender of a message cannot deny having sent the message. It involves the use of digital signatures, which are unique codes that are attached to messages to provide evidence of the sender's identity and the message's authenticity. Non-

repudiation is essential in legal and financial transactions, where it is necessary to prove that a message was sent and received.

- **Integrity** is the principle that a message or data has not been altered during transmission. This ensures that the message or data received is the same as the one that was sent, without any unauthorized changes.

Your task in this project is to set up an Internet of Things (IoT) [2] testbed made up from a number of Raspberry Pis and to connect the devices to the cloud for encrypted data storage and peer-to-peer data transmission featuring end-to-end encryption.

End-to-end encryption ensures that only the sender and the intended recipient of a message can access its contents. This means that even if a message is intercepted by a third party, they will not be able to read it. End-to-end encryption is an important way to keep data secure and private. End-to-end encryption can also help to protect against government surveillance and data breaches. By encrypting data throughout the entire communication process, end-to-end encryption makes it more difficult for governments or hackers to gain access to sensitive information. Multiple organisations have prioritised providing end-to-end encryption in their products and services[1][2][3].

Peer-to-peer sharing is a way of exchanging data between devices without the need for a central server or intermediary. In a peer-to-peer network, each device can act as both a client and a server, allowing data to be shared directly between devices. This can be a more efficient and decentralized way to share data, and it can also help to improve privacy and security by reducing the need for intermediaries.

While many different IoT technologies are available and could be used as example technologies [1], we will use small Raspberry Pi clusters as example technologies. The Raspberry Pis are equipped with GrovePi[4], LEDs, an LCD display and sensors as listed below. The hardware equipment (i.e., Raspberry Pi, GrovePi, and further equipment) will be supplied (and needs to be handed back at the end of the semester). Remember that the provided hardware behaves not on an industry level and it is possible that some sensors return inaccurate or wrong data. In case the returned value is not valuable in any way, try switching to another sensor. Additionally, there could be problems with the GrovePi that require some troubleshooting like a firmware update or some special handling of concurrency when sending requests to the board.

By working on this project, you will gain hands-on experience with these important concepts and learn how they can be applied in real-world scenarios to improve data security and privacy.

## Equipment:

As written above, the data to be shared is IoT sensor data. Accordingly, you will first have to source the data from the sensors. For this, each group will receive the following equipment:

- A Raspberry Pi 3B+ set with all necessary accessories (Click for details)
- A compatible GrovePi+ Starter Kit (Click for details)
- A microSD card with adapter for the Raspberry Pi OS (Click for details)
- A Grove heart rate sensor (Click for details)
- A Grove air quality sensor (Click for details)
- A Grove gesture recognition sensor (Click for details)

[1] `https://support.apple.com/en-us/HT202303`
[2] `https://cloud.google.com/docs/security/encryption/default-encryption`
[3] `https://faq.whatsapp.com/820124435853543`
[4] `http://www.dexterindustries.com/grovepi/`

## Requirements and Design Decisions:

The project outcome will have three major components: the data storage system and the data sharing system.

1. **The data storage system:** The storage system will involve acquiring streaming data from the sensors, encrypting them in batches and storing them in the Google Cloud.
   a) Connect the Grove sensors to the Raspberry Pi boards following the instructions provided with the sensors. Please note that the Raspberry Pi will be provided without any operating system,i.e., you need to start by installing the Raspberry Pi OS[5]. This operating system is predetermined, i.e., you need to use the Raspberry Pi OS.
   b) Install the necessary software on the Raspberry Pi boards to enable communication with the sensors. You can find the required software and installation instructions on the Web sites linked above.
   c) Configure the Raspberry Pi boards to connect to the Internet and the Google Cloud storage. The data connection should be over Wi-Fi.
   d) Collect the data from the sensors and process them according to a predefined batch size. The user should be able to define the batch size according to their requirements. The batch size should depend either on the number of items or a specific time interval, or both (in case there is a delay in transmission).
   e) The definition of this project demands that the sender should be the only one who is able to decrypt the data. Hence, a symmetric encryption scheme can be used.
   f) Choose a data storage service provided by the Google Cloud Platform, such as Cloud Storage, Cloud SQL etc.
   g) Implement a system to upload the encrypted batches from the Raspberry Pi to the storage service you have chosen.
   h) The system should also be able to retrieve and decrypt data from the same storage.
2. **The data sharing system:** The sharing system should enable end-to-end encrypted peer-to-peer data transmission between two parties.
   a) Decide on an asymmetric encryption scheme to enable end-to-end encrypted communication.
   b) Since you will only be provided with a single Raspberry Pi set, you should make the communication system platform agnostic. Hence, the Raspberry Pi should be able to communicate with your PC, provided your application is running on both.
   c) The system should have a system to distribute the keys before the start of a communication. Decide on a key exchange algorithm for the same.
   d) The sender should encrypt the data and the receiver should be able to decrypt the data using the keys decided upon.
   e) The system should have a user interface where each party can see the data being sent and received.
   f) The sharing system should also use the same batch-base processing as in the storage system. The receiver should be able to verify the integrity of the batch (e.g., using hashes).
   g) Decide on the actual medium of communication. All solutions will be accepted as long as they successfully transfer data between two endpoints. A centralized storage location (e.g., Google Cloud storage) can also be used as an intermediary.

[5] `https://www.raspberrypi.com/software/`

# Outcomes

The expected outcomes of this project are two-fold: (1) the actual project solution, (2) two presentations of the results.

## Project Solution

The project has to be hosted on a Git repository in TUHH's GitLab instance[6] – you will get instructions about the repository setup at the lab kickoff meeting. Every member of your team has to use its own, separate Git account. *We will check who has contributed to the source code*, so please make sure you use your own account when submitting code to the repository. Furthermore, it is required to provide an easy-to-follow README that details how to deploy, start and test the solution. The best practice is to provide a README that describes "Plug-and-Play" instructions on how to use your solution.

For the submission (see below), you also have to create one or more Docker images, which contain(s) your complete implemented solution, i.e., including all dependencies.

## Presentations

There are two presentations. The first presentation is during the mid-term meetings, and covers your achievements till then and the second presentation is during the final meetings and contains all your results. The actual dates are announced in Stud.IP.

Every member of your team is required to present in either the first *or* the second presentation. Each presentation needs to consist of a slides part and a demo of your implementation. Think carefully about how you are going to demonstrate your implementation, as this will be part of the grading. You have 10 minutes (strict) of time for your presentation at the mid-term meetings, and 15 minutes (also strict) of time for your presentation at the final meeting. We recommend to use about 5 minutes at each meeting for the slides part of your presentation, and the remaining 5 / 10 minutes, respectively, for the live demonstration of your implementation results. After each presentation, there will be a Q&A session of max. 5 minutes. The past has shown that providing a nice use case story usually helps to present the project outcomes. While providing such a use case story is not an absolute must, it will surely help the audience to understand your work better.

## Grading

A maximum of 60 points are awarded in total for the project. Of this, 70% are awarded for the implementation (taking into account both quality and creativity of the solution as well as code quality and documentation), and 30% are awarded for the presentations (taking into account content, quality of slides, presentation skills, and discussion).

A strict policy is applied regarding plagiarism. Plagiarism in the source code will lead to 0 points for the particular student who has implemented this part of the code. If more than one group member plagiarizes, this may lead to further penalties, i.e., 0 points for the implementation of the whole group.

## Deadline

The hard deadline for the project is **July 10th, 2023**, 23:59. Please submit the presentations via Stud.IP. Late submissions will not be accepted.

## Test Cloud Infrastructure

This year, the Google Cloud Platform is supporting the IoT lecture with an Education Grant, which you can use for Virtual Machines (VMs) and other cloud resources. The computational

[6] `https://collaborating.tuhh.de/`

resources can be used for free, however, you need to own a Google account to use them. To access the resources, please visit the following URL (please note that this is a masking URL, i.e., you need to click it, simply copy/pasting it will *not* help): http://google.force.com/. The full URL is also provided in Stud.IP.

Please note that you need to request the resources until August 1th, 2023; the coupons are then valid until April 1th, 2024. Afterwards, they will become void and you will not be able to access the budget.

The e-mail address you provide in this form does *not* have to be your Google account, but it *must* be an address within the TU Hamburg domain, i.e., `@tuhh.de`. By providing such an address, you can claim a coupon code, which you can then use to redeem a $50.00 voucher for the Google Cloud Platform. Detailed instructions are provided at this URL and subsequent e-mails you will receive from Google.

You can redeem one coupon code per e-mail address at a time. The voucher of $50.00 is in most cases sufficient to conduct the implementation tasks of the lab exercises. However, if this is not the case, please send a mail to avik.banerjee@tuhh.de. Please also take into account that there are the free tier resources of the Google Cloud Platform, which are usually sufficient to test and run your solutions.

Note that exceeding this budget will lead to immediate shutdown of the project by Google, and we have no means of influencing this, or providing extensions. We therefore recommend you to monitor your resource usage, in order to avoid unnecessary spending, for instance, by VMs left running.

You are allowed to use the Google Cloud Resources only for the purpose of this course. Both Google and TU Hamburg are monitoring the use of the resources, and any misuse (hosting of illegal data, cryptocurrency mining, etc.) will be punished. In case of any questions regarding the organization of the Google Cloud Platform, please send an e-mail to avik.banerjee@tuhh.de.

Please take care that you do not publish any credentials or (ssh) keys on the Internet. For your implementation, please use dedicated credential files and add these credential files and the keys to the .gitignore file or put them on a system path outside of your Git repository. If you accidentally leak any credentials or ssh keys, inform us immediately. Google is pretty good at identifying leaked credentials, and will then shut down your account at least temporarily. If this happens just before the deadline, there is a good chance that your account will not be released again in due time.

In case of noncompliance, the person(s) responsible will fail the lab!

## References


[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communication Surveys & Tutorials*, 17(4):2347–2376, 2015.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer Networks*, 54:2787–2805, 2010.