**1. Basic Communication Simulation:**

* **What we did:** We started by simulating a basic communication system between two satellites. One satellite (the sender) sends a message, and the other satellite (the receiver) gets the message.
* **Key concepts:**
  + **Packet:** We sent a message packed into a "data packet," which is like a letter that contains the actual message along with some additional information (like who sent it).
  + **Sender and Receiver:** The sender sends the message, and the receiver extracts and reads it.

**2. Adding Encryption (AES):**

* **What we did:** To make the communication secure, we added **AES encryption** (a method of scrambling the message) to protect the message from being read by anyone other than the receiver.
* **Key concepts:**
  + **Encryption:** This is the process of turning a readable message (plaintext) into an unreadable format (ciphertext) using a secret key.
  + **AES Encryption:** It's like having a secret lock and key to encrypt and decrypt the message. Only the receiver who has the key can unlock (decrypt) it and read the message.
  + **AES Key:** The key used for encryption/decryption. It's a random key generated by the sender and used to secure the message.
* **Process:**
  + The sender encrypts the message using a randomly generated AES key.
  + The receiver decrypts it using the same key to get the original message back.

**3. RSA Encryption for Key Exchange:**

* **What we did:** Next, we introduced **RSA encryption**, a method used to securely exchange the AES key between the sender and receiver.
* **Key concepts:**
  + **RSA Encryption:** It's like a public lock and private key system. Anyone can lock (encrypt) something with a public key, but only the person with the private key can unlock (decrypt) it.
  + **Public and Private Keys:** The sender uses the receiver's public key to encrypt the AES key. The receiver then uses their private key to decrypt it.
  + This step ensures that the AES key (which is used to encrypt the message) is securely sent between the two satellites.

**4. Combining AES and RSA (Hybrid Encryption):**

* **What we did:** We combined **RSA** (for securely exchanging the AES key) and **AES** (for encrypting the actual message). This combination is called **hybrid encryption**.
* **Key concepts:**
  + **Hybrid Encryption:** It uses the strength of both encryption methods. RSA securely shares the AES key, and AES securely encrypts the actual message.
  + The sender encrypts the message with AES and encrypts the AES key with RSA. The receiver decrypts the AES key with RSA and then uses it to decrypt the message.

**5. Building CCSDS-style Packets:**

* **What we did:** We added **CCSDS-style packets** to the communication system. CCSDS (Consultative Committee for Space Data Systems) is a standard for communication between space systems, and it defines how data is formatted in packets.
* **Key concepts:**
  + **Packet Structure:** We created a data packet that includes not only the encrypted message and encrypted AES key but also additional information like a timestamp, sequence number, and packet type. This is necessary for organized and reliable communication.

**6. Visualizing the Message Flow:**

* **What we did:** We used **matplotlib**, a Python library, to create a diagram showing how the message flows through the system. This helps in visualizing the different steps, such as encryption, key exchange, and decryption.
* **Key concepts:**
  + **Message Flow:** We mapped out the steps involved in sending and receiving a secure message. This flow includes creating the message, encrypting it, sending it, decrypting it, and reading it.
  + **Arrows and Steps:** Each step in the process is displayed as a box with an arrow pointing to the next step, making it easier to understand the sequence of events.

**7. Simulating Timing and Delays:**

* **What we did:** We added a **timing simulation** to show how long each step in the communication process takes. This gives us an idea of the delays in the system.
* **Key concepts:**
  + **Timestamp:** We simulated the time at which each step happens and visualized this as a timeline.
  + **Delay Simulation:** We introduced a small delay (e.g., 0.5 seconds) between each step to simulate real-world timing in satellite communication, as delays can occur in space systems.

**In Summary:**

We built a simulation of a secure communication system between satellites. Here’s the complete process:

1. **Message Creation:** The sender creates a message.
2. **Encryption:** The message is encrypted using AES (with a secret key).
3. **RSA Encryption:** The AES key is encrypted with RSA and sent to the receiver.
4. **Secure Communication:** The receiver decrypts the AES key using their private RSA key and then decrypts the message using the AES key.
5. **CCSDS Packet:** The message and encrypted key are packed into a CCSDS-style packet, following a specific format used in satellite communication.
6. **Visualizing the Process:** We visualize the flow of messages, encryption, and decryption steps.
7. **Simulating Delays:** We simulate the time taken for each step and show how delays might look in the system.

This system simulates how secure communication would work in a satellite network using encryption, key exchange, and organized data packets.

**Topic Overview: Data Security Using Cryptographic Encryption for Satellite Networks**

The goal of this project is to explore **data security** in satellite communication systems. In real-world satellite communication, data is transmitted over long distances and in challenging conditions, making it vulnerable to various types of attacks. To ensure the privacy, integrity, and authenticity of the messages being sent between satellites, **cryptographic encryption** is used. This encryption ensures that even if an unauthorized party intercepts the data, they will not be able to read or tamper with it.

The encryption methods used in the project are:

1. **AES (Advanced Encryption Standard):** Used to encrypt the actual message being sent. It is a symmetric encryption algorithm, meaning the same key is used for both encryption and decryption.
2. **RSA (Rivest-Shamir-Adleman):** Used for **secure key exchange**. RSA is an asymmetric encryption algorithm, where two keys are used: one public (for encryption) and one private (for decryption).

**Why We Did This:**

**1. Why Satellite Communication Needs Security:**

Satellite communication is used in critical systems, such as GPS, weather monitoring, defense, and communication for remote regions. Due to the open nature of space and the vast distances involved, data transmitted via satellites is vulnerable to various threats, including:

* **Eavesdropping** (unauthorized interception of data)
* **Data tampering** (altering the transmitted message)
* **Replay attacks** (replaying previously captured messages)
* **Impersonation** (faking the identity of a satellite or ground station)

Therefore, it is essential to secure the data using encryption techniques to protect against these threats.

**2. Why We Used AES and RSA:**

In satellite communication, securing both the message and the key is important. Here’s why we chose AES and RSA:

* **AES (Symmetric Encryption):**
  + **Efficiency:** AES is faster than RSA for encrypting large amounts of data, making it ideal for encrypting the actual message.
  + **Security:** AES is widely considered secure and is used in many real-world systems (like online banking, VPNs, etc.).
  + **Symmetric Key Issue:** The problem with AES is that both the sender and the receiver need the same secret key to encrypt and decrypt the message, which brings up the challenge of securely exchanging the key.
* **RSA (Asymmetric Encryption):**
  + **Key Exchange:** RSA solves the key exchange problem by allowing the secure transmission of the AES key. The sender uses the receiver’s public RSA key to encrypt the AES key. The receiver can then decrypt the AES key using their private RSA key.
  + **Security:** RSA provides security by using two different keys (public and private), so even if someone intercepts the public key, they cannot decrypt the message without the private key.

By combining these two techniques, **hybrid encryption** ensures both security and efficiency:

* **RSA** handles the secure exchange of keys.
* **AES** handles the fast and secure encryption of the message itself.

**3. Why We Used CCSDS Format:**

The **CCSDS (Consultative Committee for Space Data Systems)** packet format is a standardized format used for space communication. It defines how data should be structured for sending and receiving over satellite systems. In this project, we used the CCSDS-style packet to:

* Follow the actual standard used in satellite communication.
* Include essential metadata like timestamp, packet type, sequence numbers, etc.
* Organize the encrypted message and encrypted AES key into a structured packet that can be sent, stored, and processed easily.

**4. Why We Simulated Timing and Delays:**

In real-world satellite communication, the transmission of data is not instantaneous. There are **delays** due to factors like:

* **Distance between satellites** (signals take time to travel).
* **Processing time** for encryption and decryption.
* **Transmission delay** based on network conditions.

By simulating delays in the system, we can:

* Understand the **time it takes** for each step in the communication process (encryption, transmission, decryption).
* Visualize the timing of each action to get a sense of how long satellite communication takes in reality.
* Identify possible **bottlenecks** or areas where improvements can be made for better performance.

**5. Why We Visualized the Message Flow:**

Visualization is key to understanding complex systems, especially in a multi-step process like satellite communication. By using diagrams, we can:

* **Show the sequence of events** that happen when the sender and receiver communicate.
* Help **clarify the roles** of different components (like encryption, decryption, and key exchange).
* **Simplify** the concept of secure communication by showing it step-by-step.

**6. Why We Combined All These Elements:**

Satellite communication systems are complex, and they need to be both **secure** and **efficient**. By combining:

* **AES encryption** for securing the message,
* **RSA encryption** for securely exchanging the AES key,
* **CCSDS packet formatting** for structuring the data,
* **Timing simulations** for real-world communication delays,

we created a realistic and secure simulation of how data might be transmitted between satellites in a **real-world space communication system**. This simulation can serve as a model for understanding the challenges and solutions involved in secure satellite communication, and it can be extended to more complex systems for **mission-critical** satellite communications, where security is paramount.

**Conclusion:**

This project aimed to demonstrate how we can use encryption techniques (AES and RSA) to protect data in satellite communication, and how to ensure that the communication is not only secure but also properly formatted and timed. This approach is essential for real-world applications, like satellite-based internet, space exploration, and defense communication, where data integrity and security are critical.