INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

WORKSHEET-5

Security in the Internet of Things

Name:	**Number:**
Bikal Bista	a43323
Dibya Raj Khatri	a57096

1. Data Security in IoT

(a) Data Confidentiality in the Internet of Things

Data confidentiality ensures that only authorized individuals or systems have access to sensitive information. In IoT, this means protecting the data exchanged between devices from being intercepted or read by unauthorized parties through encryption and access control mechanisms.

(b) Data Integrity in the Internet of Things

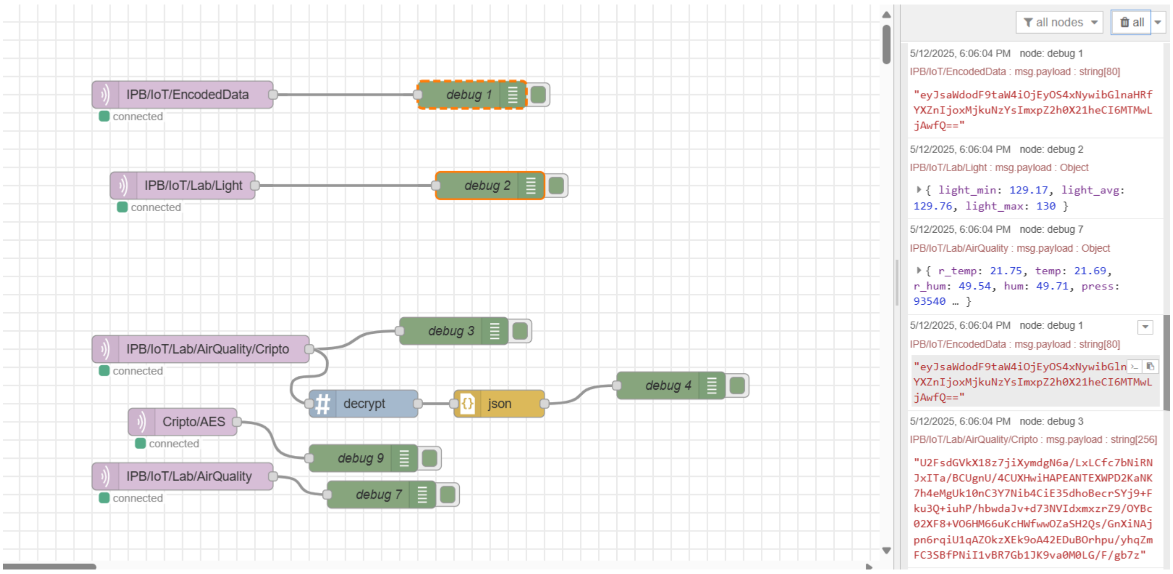
Data integrity ensures that the data being transmitted or received has not been tampered with. In IoT systems, this is crucial to make sure commands or sensor data remain accurate and trustworthy during transmission.

(c) Data Availability in the Internet of Things

Data availability means ensuring that the IoT services and data are accessible when needed. This includes having reliable devices, networks, and power supplies so that systems such as health monitors or industrial controls can function without interruptions.

2.Encoded Data

When data is transmitted over the internet, it often needs to be encoded into a standardized format to ensure compatibility across different systems and protocols. One common technique is Base64 encoding, which converts binary or structured data, such as JSON, into a plain text format that can be safely sent over networks. In this project, after subscribing to the MQTT topic `IPB/IoT/EncodedData`, I received the encoded payload `"eyJsaWdodF9taW4iOjEyNS44MywibGlnaHRfYXZnIjoxMjYuMjksImxpZ2h0X21heCI6MTI2LjY3fQ=="`. Using the online tool CyberChef, I decoded this Base64 string into a readable JSON object containing three values: `light_min: 125.83`, `light_avg: 126.29`, and `light_max: 126.67`. These values exactly matched the object received from the MQTT topic `IPB/IoT/Lab/Light`, confirming that the original data was encoded using Base64 for safe transmission. This process highlights the importance of character encoding techniques like Base64 in ensuring reliable and interpretable data exchange in IoT environments.



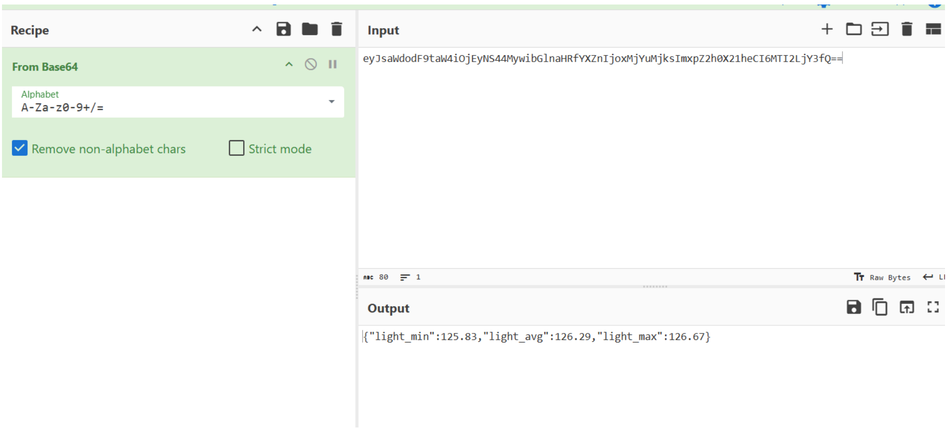
```
5/12/2025, 6:06:04 PM   node: debug 1
IPB/IoT/EncodedData : msg.payload : string[80]
"eyJsaWdodF9taW4iOjEyOS4xNywibGlnaHRfYXZnIjoxMjkuNzYsImxpZ2h0X21heCI6MTMwLjAwfQ=="

5/12/2025, 6:06:04 PM   node: debug 2
IPB/IoT/Lab/Light : msg.payload : Object
▸{ light_min: 129.17, light_avg: 129.76, light_max: 130 }

5/12/2025, 6:06:04 PM   node: debug 7
IPB/IoT/Lab/AirQuality : msg.payload : Object
▸{ r_temp: 21.75, temp: 21.69, r_hum: 49.54, hum: 49.71, press: 93540 … }

5/12/2025, 6:06:04 PM   node: debug 1
IPB/IoT/EncodedData : msg.payload : string[80]
"eyJsaWdodF9taW4iOjEyOS4xNywibGlnaHRfYXZnIjoxMjkuNzYsImxpZ2h0X21heCI6MTMwLjAwfQ=="

5/12/2025, 6:06:04 PM   node: debug 3
IPB/IoT/Lab/AirQuality/Cripto : msg.payload : string[256]
"U2FsdGVkX18z7jiXymdgN6a/LxLCfc7bNiRNJxITa/BCUgnU/4CUXHwiHAPEANTEXWPD2KaNK7h4eMgUk10nC3Y7Nib4CiE35dhoBecrSYj9+Fku3Q+iuhP/hbwdaJv+d73NVIdxmxzrZ9/OYBc02XF8+VO6HM66uKcHWfwwOZaSH2Qs/GnXiNAjpn6rqiU1qAZOkzXEk9oA42EDuBOrhpu/yhqZmFC3SBfPNiI1vBR7Gb1JK9va0M0LG/F/gb7z"
```



Alphabet A-Za-z0-9+/=

☑ Remove non-alphabet chars ☐ Strict mode

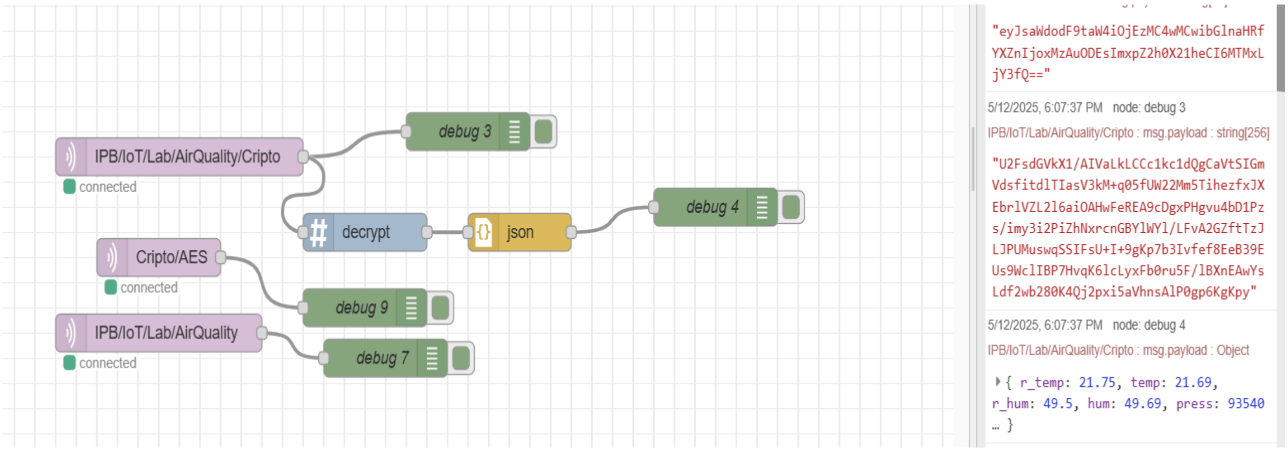
```
eyJsaWdodF9taW4iOjEyNS44MywibGlnaHRfYXZnIjoxMjYuMjksImxpZ2h0X21heCI6MTI2LjY3fQ==
```

abc 80 ≡ 1

```
{"light_min":125.83,"light_avg":126.29,"light_max":126.67}
```

3.Encrypted Data

To analyze encrypted data in this project, I subscribed to the MQTT topic `IPB/IoT/Lab/AirQuality/Cripto` using Node-RED and received the following encrypted payload: `"U2FsdGVkX18z7jiXymdgN6a/LxLCfc7bNiRNJxITa/BCUgnU/4CUXHwiHAPEANTEXWPD2KaNK7h4eMgUk10nC3Y7Nib4CiE35dhoBecrSYj9+Fku3Q+iuhP/hbwdaJv+d73NVIdxmxzrZ9/OYBc02XF8+VO6HM66uKcHWfwwOZaSH2Qs/GnXiNAjpn6rqiU1qAZOkzXEk9oA42EDuBOrhpu/yhqZmFC3SBfPNiI1vBR7Gb1JK9va0M0LG/F/gb7z"`. To decrypt it, I also subscribed to the topic `Cripto/AES` and obtained the secret key needed for AES decryption. Using the `node-red-contrib-crypto-js` library with the AES algorithm and the provided key, I successfully decrypted the message, revealing readable air quality data that matched the values from the unencrypted topic `IPB/IoT/Lab/AirQuality`. This confirms that the encrypted data was securely transmitted and accurately decrypted, demonstrating the effectiveness of AES encryption for protecting IoT data in transit.



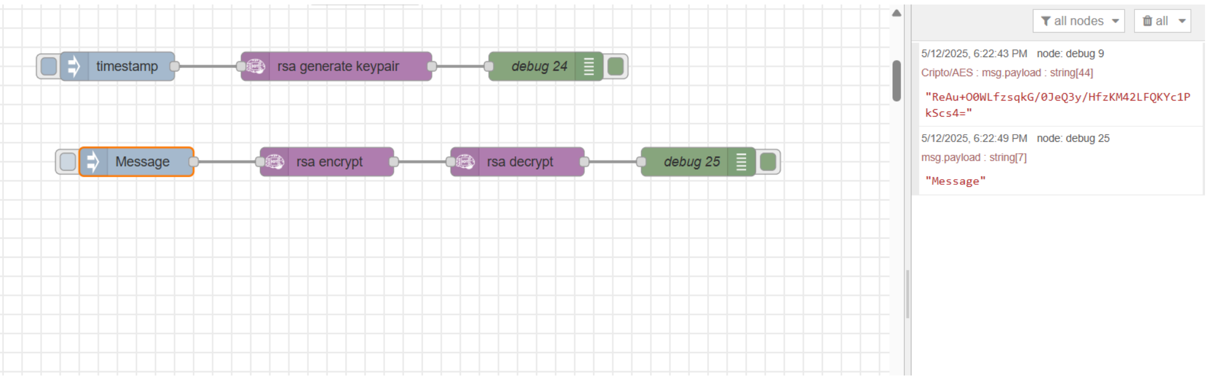
```
"eyJsaWdodF9taW4iOjEzMC4wMCwibGlnaHRf
YXZnIjoxMzAuODEsImxpZ2h0X21heCI6MTMxL
jY3fQ=="

5/12/2025, 6:07:37 PM   node: debug 3
IPB/IoT/Lab/AirQuality/Cripto : msg.payload : string[256]
"U2FsdGVkX1/AIVaLkLCCc1kc1dQgCaVtSIGm
VdsfitdlTIasV3kM+q05fUW22Mm5TihezfxJX
EbrlVZL2l6aiOAHwFeREA9cDgxPHgvu4bD1Pz
s/imy3i2PiZhNxrcnGBYlWYl/LFvA2GZftTzJ
LJPUMuswqSSIFsU+I+9gKp7b3Ivfef8EeB39E
Us9WclIBP7HvqK6lcLyxFb0ru5F/lBXnEAwYs
Ldf2wb280K4Qj2pxi5aVhnsAlP0gp6KgKpy"

5/12/2025, 6:07:37 PM   node: debug 4
IPB/IoT/Lab/AirQuality/Cripto : msg.payload : Object
▸{ r_temp: 21.75, temp: 21.69,
r_hum: 49.5, hum: 49.69, press: 93540
… }
```

4.node-red library “sense-rsa”

Using the Node-RED library “sense-rsa,” I generated a pair of cryptographic keys: a public key and a private key. I then created a custom message and successfully encrypted it using the public key, and decrypted it using the private key. In this process, the public key is used to encrypt data, making it readable only by someone with the matching private key, which ensures confidentiality. This asymmetric encryption approach is highly relevant in the Internet of Things (IoT) because it allows secure communication between devices, where data can be transmitted openly but only decrypted by trusted recipients, protecting sensitive information such as sensor readings, user data, or commands from unauthorized access or tampering.



```
5/12/2025, 6:22:43 PM   node: debug 9
Cripto/AES : msg.payload : string[44]
"ReAu+O0WLfzsqkG/0JeQ3y/HfzKM42LFQKYc1P
kScs4="
5/12/2025, 6:22:49 PM   node: debug 25
msg.payload : string[7]
"Message"
```

5. the main differences between coding and encrypting data

The main differences between coding and encrypting data lie in their purpose, security level, and who can understand the transformed data. Coding (or encoding) is the process of converting data into a different format using a publicly known scheme—like Base64 or UTF-8—so that it can be properly stored or transmitted; it's not meant to hide the data, and anyone can decode it easily. In contrast, encrypting data transforms it into an unreadable format using a specific algorithm and a secret key, with the main goal of protecting the data from unauthorized access; only someone with the correct decryption key can read the original information. While encoding ensures compatibility and data integrity, encryption ensures confidentiality and security, which is critical in sensitive applications like online banking, communications, and IoT systems.

7.A

(a) What is the hash function?

A hash function is a one-way mathematical algorithm that converts input data of any size into a fixed-size string, known as a hash. This process is irreversible, meaning you cannot obtain the original data from the hash value. Hash functions are commonly used in computer systems for purposes like verifying data integrity, storing passwords securely, and detecting changes in data. Even a small change in the input will produce a significantly different hash, making it effective for ensuring that data has not been altered.

(b) In addition to authentication, how does Node-RED handle credential management (e.g. passwords)? How important is it to guarantee the security of this information?

Node-RED handles credential management by storing sensitive data, such as passwords or API keys, in a separate file called `flows_cred.json`. These credentials are encrypted using a secret key defined in the `settings.js` file. This means the credentials are not stored in plain text, adding a layer of

protection. Securing this information is critically important because unauthorized access to these credentials can lead to system compromise, unauthorized control of connected devices, or data leaks. Proper credential handling ensures that authentication data remains confidential and reduces security risks.

(c) Considering the answer to the previous question, how does this approach increase system security and protect against certain types of attacks?
By encrypting credentials separately from the main application flow and using a defined secret key, Node-RED improves system security by ensuring that sensitive data cannot be easily read or reused by attackers, even if they gain access to the file system. This protects the system from attacks like credential theft, brute force password attacks, and man-in-the-middle (MITM) attacks. The use of encryption and secure credential storage helps enforce confidentiality and integrity, which are essential principles of cybersecurity, especially in IoT environments where devices often communicate sensitive data.