The WiFi based secured wireless communication using RSA

ABSTRACT:

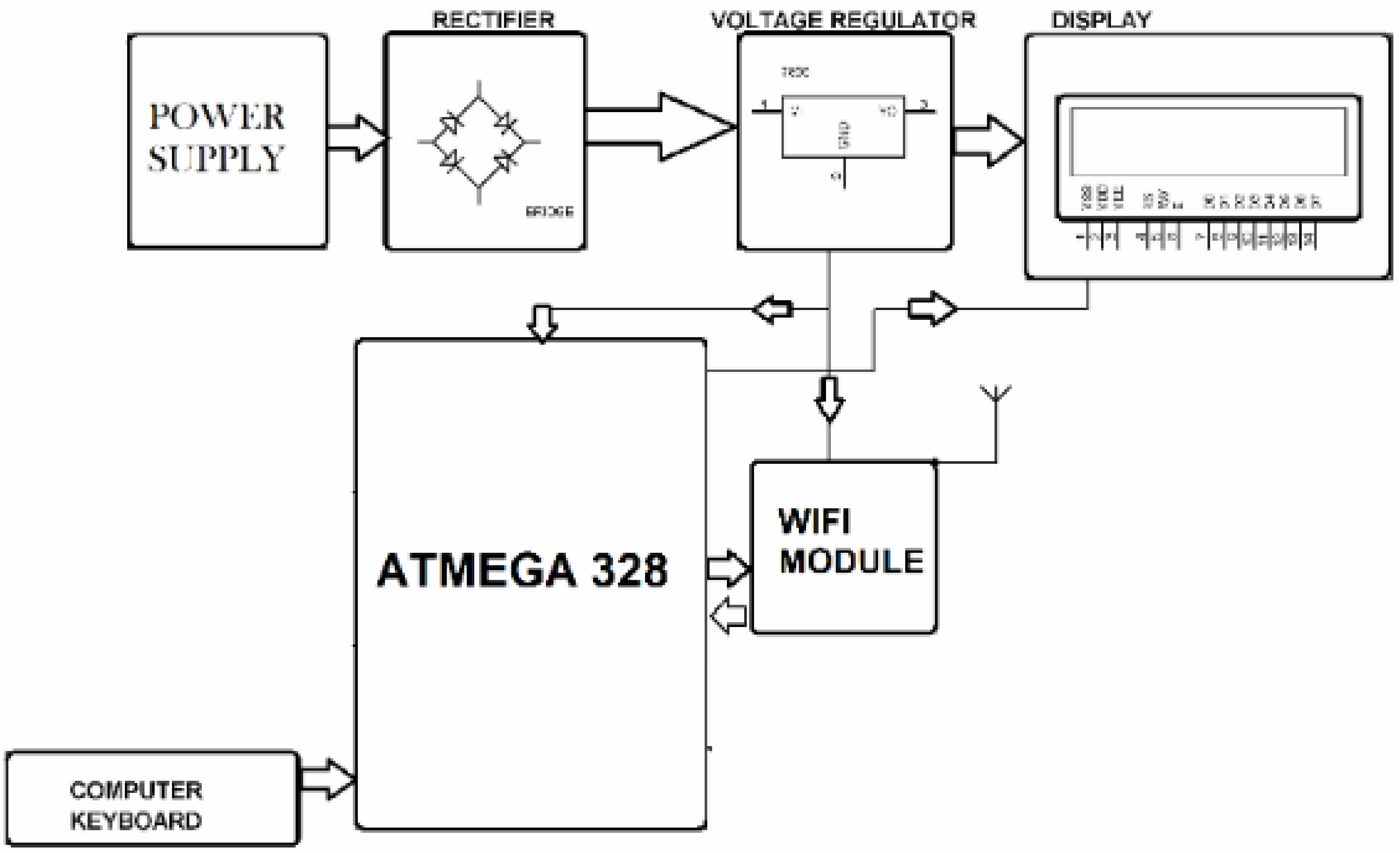
The WiFi based secured wireless communication using RSA encryption allows us to communicate wirelessly with security feature. The data transfer during communication between two system is encrypted using RSA encryption which is highly secure.

The data can be decrypted with correct key only, otherwise it returns some garbage value. This is two way communication system where we can transmit as well as receive at both ends. We have used Atmega microcontroller interfaced with xbee along with LCD display to send message and key, also have USB keyboards connected toeach system and powered by 12V supply.

After starting system, we will able to enter message on system. The maximum limit of message is 32 character. After that system asks for key, the key limit is 16character it can be number or alphabet. Entering key will send the encrypted message to other system. Then the other system ask key to view the message. If the users enter correct key the message gets decrypted otherwise it will show garbage value thus securing the wireless communication.

Block Diagram :

TRANSMITTER:



METHODOLOGY:

The methodology for "Wi-Fi Based Secure Wireless Communication Using RSA" is designed to achieve secure data transmission by integrating the RSA encryption algorithm with Wi-Fi communication. The block diagram shown in Figure provides a comprehensive overview of the system's architecture, which includes key components such as the ATMEGA328 microcontroller, ESP8266 Wi-Fi module, computer keyboard, LCD display, and the power supply module.

The process begins with the computer keyboard acting as the input device to enter the plain text message. This data is passed to the ATMEGA328 microcontroller, which serves as the central control unit. The microcontroller implements the RSA encryption algorithm to encrypt the input data. RSA encryption generates a secure ciphertext using a public key before sending it to the recipient. This ensures that the data cannot be accessed by unauthorized parties during transmission.

The encrypted data is then transmitted via the ESP8266 Wi-Fi module. Communication between the microcontroller and the ESP8266 module takes place through the UART interface using TX (Transmit) and RX (Receive) pins. The ESP8266 module wirelessly transmits the encrypted data to a receiving device, which decrypts it using the private RSA key to recover the original message. To ensure proper operation of the Wi-Fi module, a 3.3V regulated power supply is provided using a voltage regulator (AMS1117-3.3) to step down the 5V power supply. The LCD display is used to provide real-time feedback on the system status, such as "data transmission," "encryption completed," or "reception successful." This aids in monitoring the system’s functionality durin oeration.

Hardware Specifications :

□ 1. Microcontroller

• The Microcontroller (AT mega) communicates with the ESP8266 module to send and receive data over Wi-Fi. This is done through the UART (Universal Asynchronous Receiver/Transmitter) interface using the TX and RX pins.

• TX (AT mega) → RX (ESP8266): Data from the microcontroller to the ESP8266.

• RX (AT mega) → TX (ESP8266): Data from the ESP8266 back to the microcontroller**.**

2. Power Supply (3.3V)

• The ESP8266 requires a 3.3V power supply for operation, and it cannot be directly powered by 5V (which is commonly provided by microcontrollers or USB ports).

• A voltage regulator (such as AMS1117-3.3) is used to step down the 5V supply to the required 3.3V. This ensures that the ESP8266 operates without the risk of damaging the module.

• GND (Ground) of the microcontroller, ESP8266, and the voltage regulator are connected to ensure a common ground reference**.**

3. Reset Pin (RST): The Reset Pin is used to reset the ESP8266 module. This can be controlled by the
microcontroller or can be triggered manually, depending on the project requirements.

• When the module needs to restart or recover from an error, this pin is pulled low for a short duration, causing the module to reboot**.**

4. UART Interface (TX/RX)

• The TX and RX pins of the ESP8266 are used for communication with the Microcontroller.

• TX (ESP8266) → RX (Microcontroller): The ESP8266 sends data to the microcontroller.

• RX (ESP8266) → TX (Microcontroller): The microcontroller sends data to the ESP8266.

• Data can be transferred between the devices, including commands, encrypted messages, and decryption keys**.**

5. Wi-Fi Communication (TCP/IP Stack)

• Wi-Fi Network Interface: The ESP8266 has a built-in TCP/IP stack that handles all the Wi-Fi communication protocols, allowing it to send and receive data over a Wi-Fi network.

• Connect to Wi-Fi: The ESP8266 can connect to a local Wi-Fi network as a Station (STA) or it can act as an Access Point (AP) to allow devices to connect directly to it.

• Data Transmission: Once connected, the ESP8266 can transmit data over the network using protocols like TCP, UDP, or HTTP, based on the application requirements.

6. GPIO Pins

- The ESP8266 has GPIO (General Purpose Input/Output) pins that can be used for additional tasks, such as controlling LEDs, sensors, or reading user input. These GPIO pins are not critical for the basic Wi-Fi communication but can be used in your project for extended functionality, like reading data or triggering actions based on certain conditions.

7. Status Indicator (LEDs)

- Some ESP8266 modules include status LED indicators to show the current state of the device, such as:
- Wi-Fi Status: Whether the ESP8266 is connected to a Wi-Fi network or not.
- Data Transmission: Indicates whether data is being sent or received.

8. Antenna (for Wi-Fi Communication)

- The Antenna is critical for providing wireless connectivity. The ESP8266 module uses a built-in antenna (or an external one, depending on the specific module) to communicate with Wi-Fi routers or other devices.

Software Specifications :

1.Arduino Compiler: Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

2.ALGORITHM/FLOWCHART

RSA (RIVEST SHAMIR ADLEMAN) ALGORITHM:

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric means that it works on two different keys i.e. Public Key and Private Key. As the name describes the Public Key is given to everyone and the Private key is kept private.

REFERENCES:

[1] G. Leelavathi, M. Kumar, S. Rao, and P. Reddy, "End-to-End Intelligent Security Model for WSN," Wireless Personal Communications, vol. 136, no. 3, pp. 1675–1703, 2024.

[2] F. Al-Shamri and S. Safwan, "Secure Image Transmission Using RSA and OFDMA," EURASIP Journal on Image and Video Processing, vol. 2024, no. 1, Article 6, 2024.

[3] H. Sharma and M. Singh, "RSA Algorithm in Wireless Communication: A Comparative Study with Elliptic Curve Cryptography," International Journal of Computer Applications, vol. 182, no. 20, pp. 25–30, 2023.

[4] P. Soni and J. Gupta, "Securing IoT Communication Using RSA and ECC," Journal of Network Security, vol. 15, no. 4, pp. 45–53, 2023.

[5] R. Choudhury and M. Sen, "Implementing RSA Encryption in Embedded Systems for Secure Wireless Communication," Embedded Systems Letters, vol. 14, no. 2, pp. 50–57, 2022.

[6] S. S. Kaur and G. P. Singh, "Hybrid Encryption Using RSA and AES for Secure Wireless Communication," International Journal of Network Security, vol. 23, no. 1, pp. 1–10, 2021.

[7] Kumar and S. Patel, "New RSA Encryption Mechanism Using One-Time Keys," Electronics, vol. 9, no. 4, Article 612, 2020.

[8] R. R., "Wi-Fi Based Secure Wireless Communication Using RSA," International Journal of Advanced Engineering Research, vol. 14, no. 3, pp. 150–156, 2019.

[9] Y. Zhang and X. Wang, "Improved RSA Encryption Algorithm for Wireless Networks," IEEE Transactions on Wireless Communications, vol. 16, no. 11, pp. 7445–7453, 2017.

[10] J. Lee and K. Kim, "Research and Implementation of RSA Algorithm for Encryption and Decryption," IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 530– 543, 2012