Secured IoT LoRa Sensor Nodes using the DS28S60 and Amazon Web Services (AWS)

MAXREFDES9001

Introduction

The MAXREFDES9001 is a complete internet-of-things (IoT) security reference design featuring a LoRa radio based, low-power, temperature sensor node secured with a DS28S60 secure coprocessor, a LoRa gateway, and a cloud application implemented in AWS infrastructure. This reference design highlights a robust and easy to manage end-to-end security scheme with authentication and confidentiality capabilities independent of the transmission link in use—the LoRaWAN protocol in this case. The MAXREFDES9001 is designed to easily integrate into embedded systems enabling confidentiality, authentication, and integrity of information.

The sensor node is motioned by the tiny, low-power, Cortex-M4-based microcontroller MAX32660 which periodically measures the ambient temperature with the help of the DS7505, authenticates and encrypts the temperature value using AES-GCM with the DS28S60 secure coprocessor, and sends it to the AWS infrastructure over a LoRaWAN network, through a Raspberry Pi-powered gateway. To prevent rogue nodes from publishing data, joining the sensor nodes to the network requires a prior local verification using a convenient NFC-based strong authentication with the help of the MAX66242 Secure Authenticator and a dedicated Android application running on an NFC-enabled Android device.

Once the authentication is successful, proving that the sensor node is genuine, the Android device communicates with the cloud application through the Internet to provision the sensor node; that is, to generate a certificate for the sensor node and perform an AES-GCM key exchange between that sensor node and the AWS infrastructure. The Android device uses the MAX66242 as an NFC bridge to communicate with the sensor node device's microcontroller application and store the certificate into the DS28S60 coprocessor, and to have the key exchange done between the DS28S60 and the cloud application using the Elliptic Curve Diffie-Hellman (ECDH) protocol. Once the key exchange is completed, the sensor node is ready to send its data to the cloud application using the negotiated AES-GCM key. Further sensor node authentication by the cloud application is possible using ECDSA since the sensor node now has a valid certificate with a matching key pair. Incidentally, the provisioning process also joins the end device to the LoRaWAN network implemented using the AWS IoT core, but this is not the main purpose of the reference design that shows a way to secure data without relying on the security of the various underlying communication links.

Features

- DS28S60 ChipDNA technology protects private and secret keys against invasive attacks.
- DS28S60 provides end-to-end security using hardware-based ECDSA authentication, ECDH key exchange, and AES-GCM authenticated encryption.
- Complete low-power sensor node board design
- Sample LoRaWAN gateway implementation based on Raspberry Pi
- Sample cloud application implemented in AWS infrastructure highlighting end-to-end security with the sensor board's DS28S60 including ECDH key exchange, and AES-GCM secure communication.
- Source code
- Peripheral module-compatible sensor expansion port
- Raspberry Pi enables portable LoRaWAN gateway deployment.

Hardware Specification

The reference design includes the following major components: DS28S60, MAX32660, MAX66242, DS7505, and SX1262. The DS28S260 is a cryptographic coprocessor that encrypts temperature measurements using an onboard AES-GCM engine, which is accessible through a SPI interface. The DS7505 is a temperature sensor which uses the I^2C interface to provide the environment's temperature. The MAX66242 enables the NFC communication between the Android mobile device and the node. The MAX32660 is a low-power microcontroller that enables the communication between the different modules. Finally, the SX1262 is used to modulate and transmit the encrypted temperature measurement through LoRa protocol.

Rev 1; 6/24

© 2024 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.

Designed–Built–Tested

This document describes the hardware shown in Figure 1, as well as its supporting software. It provides a detailed, systematic technical guide to set up and understand the MAXREFDES9001 reference design. The system has been built and tested, details of which follow later in this document.

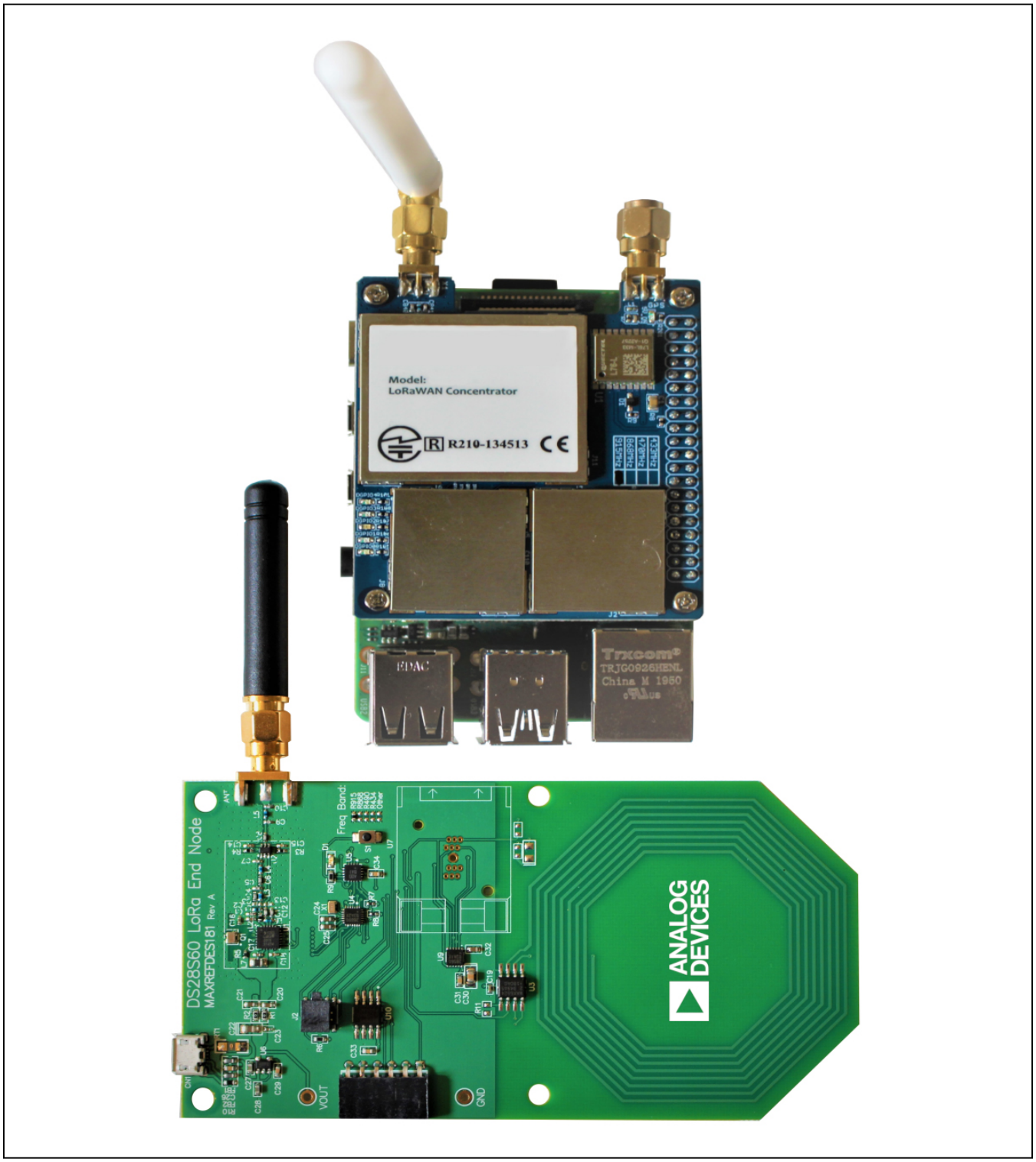

Figure 1. MAXREFDES9001 Hardware

Quick Start

Required Equipment

- Any PC or notebook computer with an internet browser and a free USB port
- MAX32625PICO board
- Two USB A to USB Micro-B cable
- 10-pin Arm Cortex Debug cable
- MAXREFDES9001 LoRa sensor node
- Android mobile device
- Raspberry Pi with Dragino PG1301 LoRaWAN Concentratorr

Procedure

The reference design, while not available to purchase, was fully assembled and tested by Analog Devices. Use the following steps to verify operation:

1) Flash the MAXREFDES9001 board with the sensor node firmware (refer to the *Flashing the Firmware* document).
2) Install the MAXREFDES9001 Android Application in the Android device (refer to the *Android Application Deployment* document).
3) Set up the AWS infrastructure and LoRaWAN Gateway. (Refer to the *AWS and LoRaWAN Gateway Quick-Start Guide*.)

Detailed Description of Hardware

The high-level block diagram of the MAXREFDES9001 hardware is shown in Figure 2. This system is composed of three main components: sensor node, LoRaWAN gateway, and Android device.

a) Sensor Node

The sensor node incorporates the following major components:

- DS28S60 Cryptographic Coprocessor
  Provides secure authentication and data encryption
- DS7505 Temperature Sensor
  Measures temperature data for transmission

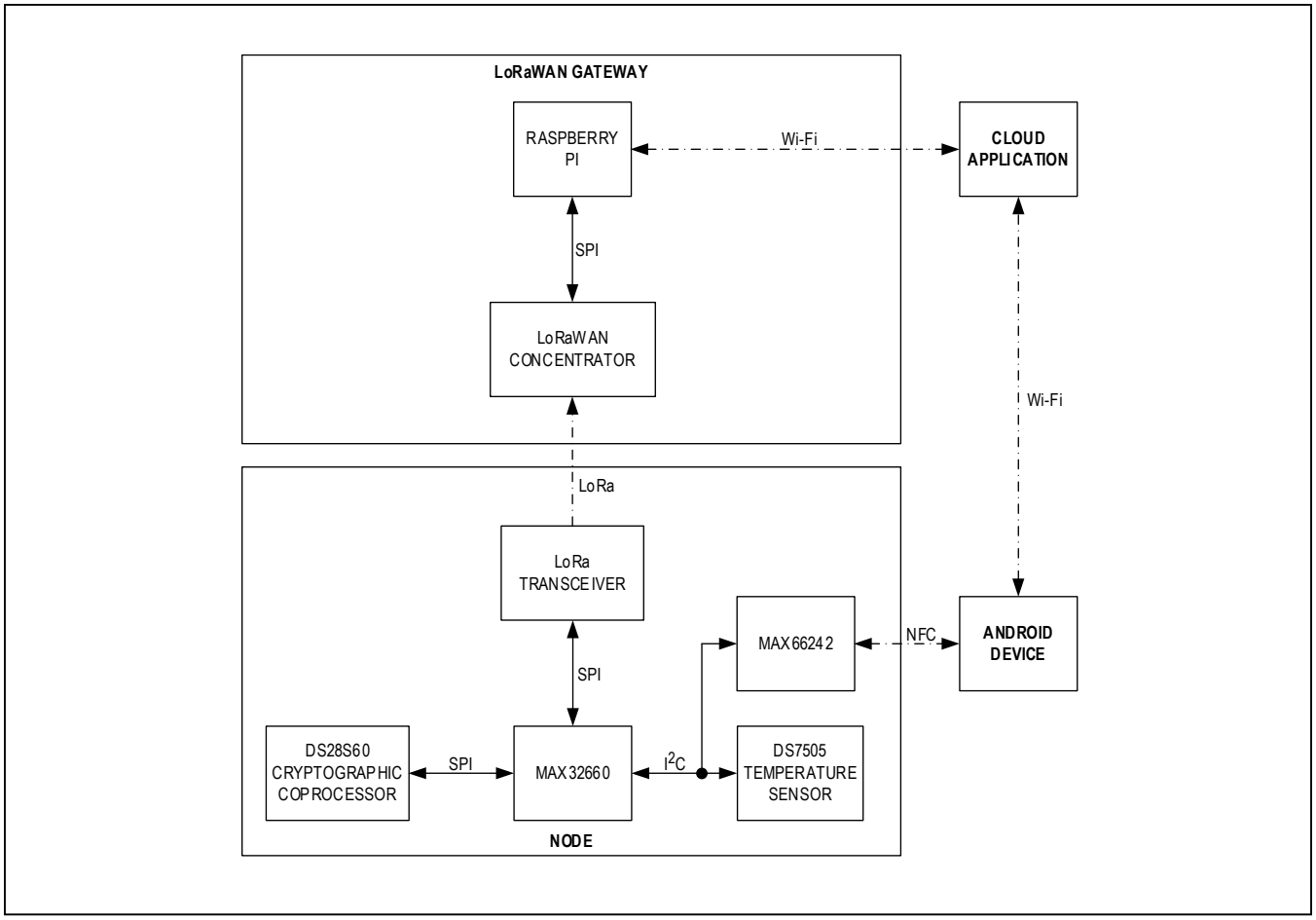

Figure 2. High-Level Block Diagram of the MAXREFDES9001

- MAX32660 Low-Power Microcontroller: Manages the sensor node operations and data processing
- MAX66242 NFC Secure Authenticator: Facilitates secure NFC communication and storage
- SX1262 LoRa Transceiver: Enables long-range wireless communication using the LoRa protocol

b) LoRaWAN Gateway

The LoRaWAN gateway consists of:

- Raspberry Pi: Serves as the main control unit for processing and forwarding data
- Dragino PG1301 LoRaWAN Concentrator: Handles the reception and transmission of LoRa packets from multiple sensor nodes

c) Android Device

The Android device runs the MAXREFDES9001 app, which provides an interface for provisioning, authenticating, and managing the sensor node. It communicates with the sensor node through NFC and interacts with the cloud application through WebSocket APIs.

Detailed Description of Software

The MAXREFDES9001 software is divided into three components: sensor device firmware, cloud application, and Android application.

The reference design sequence is as follows:

a) Provisioning of Sensor Node

1) Generation of Device Key Pair
   - The Android application, using the MAX66242 as a communication bridge, triggers the generation of a key pair for the sensor node.
   - It then reads the DS28S60 public key, ROM ID, and MANID.
2) Certificate Request
   - The Android application sends a request to the cloud application to generate a certificate for the end device, including the device's unique ID and public key.
3) Certificate Storage
   - The cloud application generates the certificate and returns it to the Android application.
   - The certificate is then stored in the DS28S60 memory.
4) AWS Public Key Storage
   - The cloud application supplies its public key, which is stored in the DS28S60 memory.
5) AES-GCM Key Generation
   - Using a Diffie-Hellman key exchange, identical AES-GCM keys are generated both in the cloud application and on the DS28S60.
6) LoRaWan Session Keys Provisioning
   - The cloud application supplies the necessary session keys to join the LoRaWan network. The keys are stored in the end device.

b) Authentication of Sensor Node

1) Certificate and Public Key Retrieval
   - The Android application requests the stored certificate and public key from the DS28S60.
2) Challenge Request
   - The Android application sends the DS28S60's certificate and public key to the cloud application to request a challenge.
3) Generate ECDSA Signature
   - Upon receiving the challenge from the cloud application, the Android application requests the DS28S60 to generate an ECDSA signature using the provided challenge.
4) Signature Verification
   - The Android application sends the ECDSA signature to the cloud application for verification.

c) Data Transmission

1) Temperature Measurement
   - The sensor node measures the temperature using the DS7505 sensor.
2) Encryption
   - The measured temperature is encrypted using the DS28S60's AES-GCM engine with the previously exchanged AES key.
3) Secure Transmission
   - The secure measurement is transmitted to the cloud application through the LoRaWan gateway.
4) Authorization
   - The cloud application retrieves the certificate and public keys to authorize the sensor node.
5) Packet Reception
   - The cloud application receives the secure measurement packet.
6) Key Retrieval
   - The cloud application retrieves the AES-GCM decryption and verification key associated with the sender end device based on the ID found in the measurement packet.
7) Data Decryption and Storage
   - The measurement data is decrypted and stored in Amazon DynamoDB, which is a part of the cloud application.
8) Data Viewing
   - The stored data in the database can be viewed through a web browser.

Sensor Node Firmware

The sensor node firmware consist of four major functions:

a) Provisioning

This process involves the DS28S60 generating a public key and returning its ROMID and MANID. The ECC certificate, LoRaWAN session keys, and AES-GCM peer public key provided by the server are stored in the DS28S60 memory. This is achieved by writing to and reading from the MAX66242 memory through I^2C with the information requested by the application.

b) Authentication

In this phase, the ECC certificate stored in the DS28S60 memory is provided. A random challenge is received from the server after the ECC certificate's authenticity is verified. The DS28S60 performs a Read Page Authentication using the received random challenge and returns the ECDSA signature.

c) Data Encryption

This step involves reading the current temperature using the DS7505 temperature sensor. The DS28S60 generates an AES-GCM encryption key by performing a Diffie-Hellman key exchange with the AES-GCM keys generated by the server. After performing AES-GCM encryption using the DS28S60, a ciphertext and an authentication tag are generated.

d) Data Transmission

This involves sending LoRa packets periodically to the LoRaWAN gateway. The onboard LED flashes to show that the LoRa packets are being transmitted. The LoRa end device joins the LoRaWAN network through Activation by Personalization (ABP) as the LoRaWAN keys are obtained during the provisioning sequence.

The LoRa packet is transmitted by the sensor node in JSON format.

- Encrypted data packet:
  Payload: DS28S60 ROM ID, AES-GCM Keys, AES-GCM Ciphertext (Temperature), AES-GCM Authentication Tag

Cloud Application Software

The AWS infrastructure hosts the cloud application software, which manages various aspects of the IoT ecosystem involving the sensor nodes. The key components and their roles are as follows:

a) AWS Lambda Functions

Serverless computing is managed by AWS Lambda functions, which perform tasks such as processing incoming data from the sensor nodes, generating and verifying certificates, decrypting data packets, and interacting with other AWS services.

b) Amazon DynamoDB

Data storage is managed by Amazon DynamoDB, a NoSQL database service. It stores the sensor nodes information, certificates, public keys, and encrypted measurement data.

c) AWS IoT Core

This service facilitates the connectivity and management of LoRaWAN devices. It handles the provisioning, authentication, and communication with the LoRaWAN gateway and routes the sensor nodes packet to the AWS Lambda functions for processing.

d) Amazon API Gateway

The Amazon API gateway is used to create, publish, maintain, monitor, and secure APIs. It serves as an interface for the Android application to interact with the cloud application through a WebSocket API.

e) AWS Amplify

AWS Amplify hosts the web application that accesses Amazon DynamoDB to display sensor node information in a user-friendly interface. As shown in Figure 3, the web application provides a detailed view of the data collected from the sensor nodes, including:

- Time Stamp: Exact date and time when the data was recorded
- Node Name: Identifier for each sensor node
- Sensor Node Rom ID: Unique ROM ID of each sensor node
- Encryption Key: AES-GCM encryption key used for securing the data
- Encrypted Data: Data collected by the sensor node, displayed in its encrypted form
- Sensor Value: Decrypted temperature measurement from the sensor node
- Authorized: Indicates whether the data was authenticated and authorized successfully

The web application allows users to search through the records, navigate between pages, and perform actions such as retrieving or deleting data using buttons like "Get Data" and "Delete Data".

Android Application Software

The MAXREFDES9001 Android app acts as a communication interface between the MAXREFDES9001 sensor node board and the cloud application used for this reference design. The primary goal of this application is to provide an intuitive interface that demonstrates the features of the DS28S60 cryptographic coprocessor by provisioning and authenticating the sensor node. The Android application commands the DS28S60 to generate the necessary information to register the sensor node with the cloud application.

Main Functions

The purpose of this demo is to showcase the features of the DS28S60 cryptographic coprocessor for which two main functions were developed.

a) Node Provisioning

During this process, the DS28S60 is asked to generate a public key from the Android application through the NFC interface that gets stored into the MAX66242's memory, which is also embedded into the node board. The Android App requests the key and transmits it back to the cloud application through a WebSocket client. The cloud application validates this data and generates a certificate that is returned to the Android application. This data is sent back to the MAX66242 though NFC, which will be then collected by the DS28S60 and will in turn store it.

b) Node Authentication

This function verifies the authenticity of the node board allowing it to transmit sensor data to the cloud application once the device is authenticated, the cloud application data is stored in the DS28S60. By running this command, the Android application request the DS28S60 to return the certificate generated during the provisioning stage and its public key through the MAX66242 NFC Tag. This data is sent to the cloud application which returns a random challenge that is used by the DS28S60 to generate a signature which gets sent to the server again to confirm its authenticity.

For more details about the Android application functionality, refer to the Android Application Details Document.

Figure 4 shows what the GUI looks like when running on the Android device. See Table 1 for more details on each functionality. See the *Design Resources* section to download the software and source code.

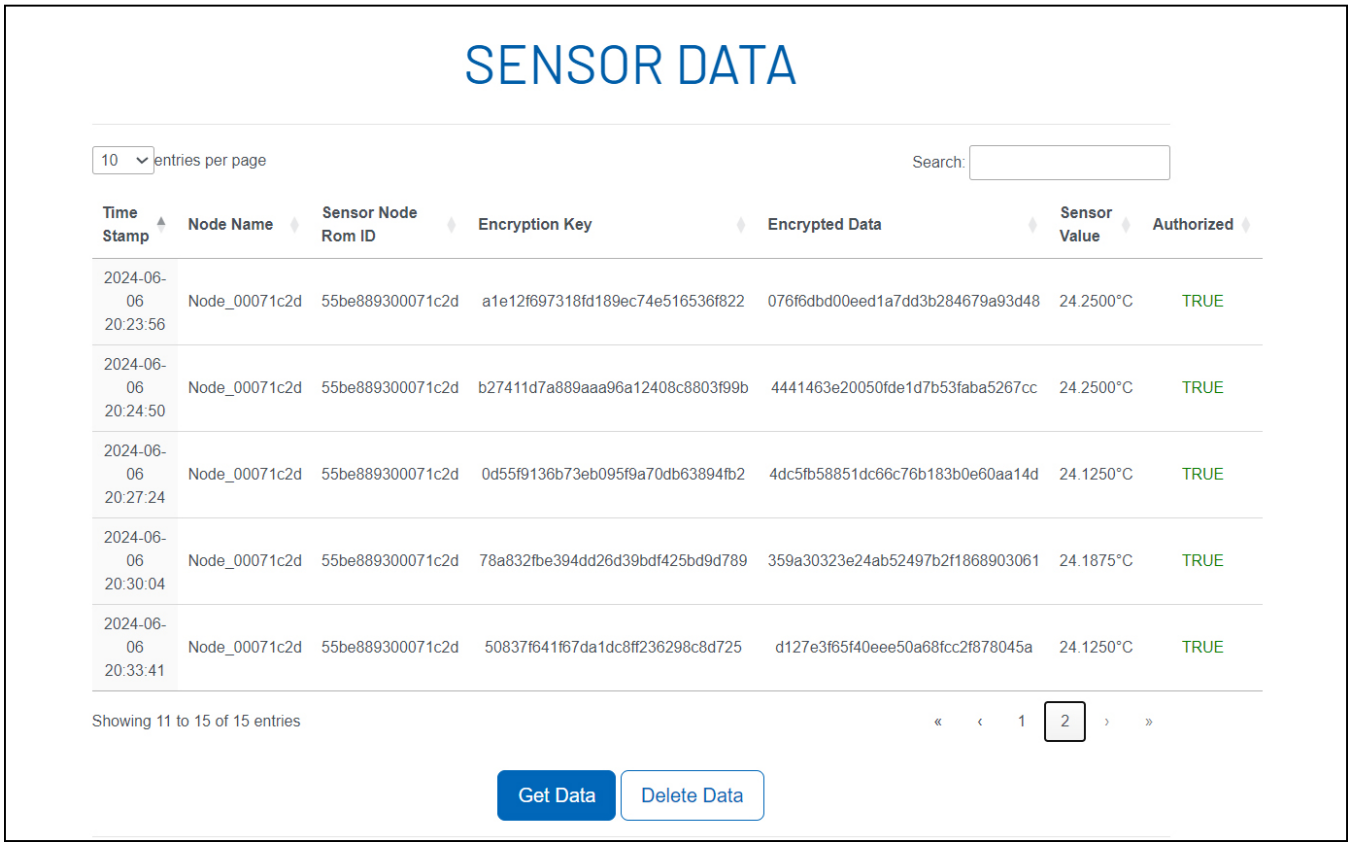

Figure 3. Data Received from Sensor Nodes

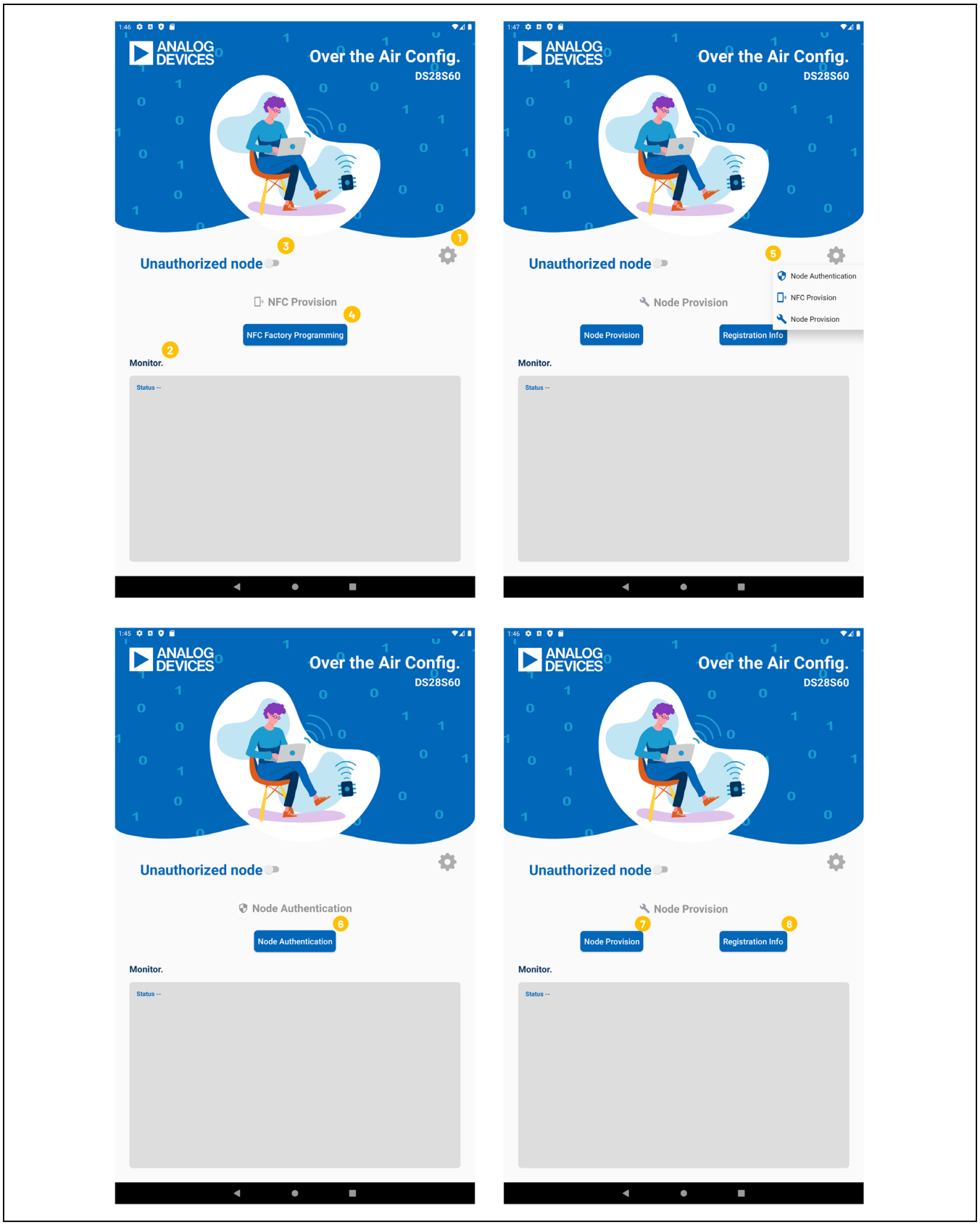

Figure 4. Main Display of the MAXREFDES9001 GUI

Table 1. GUI Controls

DESCRIPTION	FUNCTION NUMBER	DETAILS
Command Menu	1	Displays the different command options available
Display Monitor	2	Displays information during the different command processes
Unauthorized Node Mode	3	Enable or disable sending bogus data simulating the behavior of a counterfeit device
NFC Provision	4	Runs the NFC Provision Sequence
Command Options	5	List out all the different command options the user has access to
Node Authentication Command	6	Runs the Node Authentication Sequence
Node Provision	7	Runs the Node Provision Sequence
Registration Info	8	Runs the Registration Info Sequence

Design Resources

Download the complete set of **Design Resources** including schematics, bill of materials, PCB layout, and test files.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	11/20	Initial release	—
1	6/24	Updated Title, Introduction, Features, Figure 1, Quick Start, Detailed Description of Hardware, Detailed Description of Software, Figure 3, Figure 4, and Table 1	All

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.