**Security Features Documentation**

**ESP32 SIM7600 IoT Device Firmware**

**Date:** April 04, 2025  
**Generated by:** Grok 3 (xAI)

This document outlines the security features implemented in the ESP32-based IoT device firmware using the SIM7600 modem. These features ensure secure communication, data protection, and firmware integrity for the device.

**1. AES-256 Encryption**

* **Description:** Symmetric encryption using the Advanced Encryption Standard with a 256-bit key.
* **Implementation:**
  + Encrypts messages (encryptMessage) and Non-Volatile Storage (NVS) data (encryptNVSData).
  + Uses Cipher Block Chaining (CBC) mode with PKCS7 padding.
* **Purpose:** Protects data confidentiality during transmission and storage.

**2. ECDH (Elliptic Curve Diffie-Hellman)**

* **Description:** Key exchange protocol using the SECP256R1 elliptic curve.
* **Implementation:**
  + Generates a shared secret between the device and server (testECDH, requestCredentialsFromServer).
  + Utilizes the mbedtls library for cryptographic operations.
* **Purpose:** Securely negotiates encryption keys for provisioning.

**3. ECDSA (Elliptic Curve Digital Signature Algorithm)**

* **Description:** Digital signature algorithm for verifying data authenticity.
* **Implementation:**
  + Verifies OTA firmware updates using the server’s public key (verifyOTASignature).
  + Uses SECP256R1 curve with mbedtls library.
* **Purpose:** Ensures firmware updates are from a trusted source and untampered.

**4. SHA-256 Hashing**

* **Description:** Cryptographic hash function producing a 256-bit digest.
* **Implementation:**
  + Computes firmware hashes for OTA integrity checks (finishOTA).
  + Derives encryption keys from ECDH shared secrets (handleMessage).
* **Purpose:** Validates data integrity and key derivation.

**5. Base64 Encoding/Decoding**

* **Description:** Encoding scheme for binary-to-text conversion.
* **Implementation:**
  + Encodes/decodes encrypted data and signatures (base64\_encode, base64\_decode).
* **Purpose:** Ensures safe transmission of binary data over MQTT.

**6. Secure MQTT Communication**

* **Description:** Encrypted communication over MQTT using SSL/TLS.
* **Implementation:**
  + Uploads a root CA certificate (uploadCertificate) and configures SSL (setupSSL).
  + Connects to an MQTT broker over port 8883 with TLS.
* **Purpose:** Secures device-server communication against eavesdropping and tampering.

**7. Random Key Generation**

* **Description:** Generation of cryptographic keys and nonces.
* **Implementation:**
  + Uses esp\_fill\_random to generate AES keys and device keys (generateEncryptionKeys).
  + Creates random nonces for provisioning (requestCredentialsFromServer).
* **Purpose:** Enhances security by ensuring unique, unpredictable cryptographic material.

**8. Secure Storage (NVS)**

* **Description:** Encrypted storage of sensitive data in Non-Volatile Storage.
* **Implementation:**
  + Encrypts credentials (username, password) with AES-256 (saveCredentials, loadCredentials).
  + Uses the Preferences library for NVS access.
* **Purpose:** Protects stored credentials from unauthorized access.

**9. Provisioning Security**

* **Description:** Secure device provisioning with server authentication.
* **Implementation:**
  + Exchanges nonce and public key with the server (requestCredentialsFromServer).
  + Decrypts and validates credentials using ECDH-derived keys (handleMessage).
* **Purpose:** Ensures only trusted servers can provision the device.

**10. OTA Firmware Update Security**

* **Description:** Secure over-the-air firmware update mechanism.
* **Implementation:**
  + Verifies firmware size, SHA-256 hash, and ECDSA signature (finishOTA).
  + Tracks chunks to ensure complete updates (processOTAFirmware, checkMissingChunks).
  + Supports rollback to previous firmware (revertToPreviousFirmware).
* **Purpose:** Prevents installation of malicious or corrupted firmware.

**11. Factory Reset**

* **Description:** Mechanism to reset the device to factory settings.
* **Implementation:**
  + Clears NVS and boots from the factory partition (performFactoryReset).
* **Purpose:** Allows recovery from compromised states or misconfigurations.

**12. Watchdog Timer (WDT)**

* **Description:** Hardware timer to detect and recover from system hangs.
* **Implementation:**
  + Configured with a 30-second timeout (esp\_task\_wdt\_init).
  + Reset periodically in the main loop (esp\_task\_wdt\_reset).
* **Purpose:** Enhances reliability and mitigates denial-of-service attacks.

**13. Unique Device Identification**

* **Description:** Unique identifier for device authentication.
* **Implementation:**
  + Retrieves the SIM7600 modem’s IMEI as a UUID (getIMEI).
* **Purpose:** Ensures device-specific provisioning and tracking.

**14. Input Validation and Error Handling**

* **Description:** Checks to prevent exploitation of malformed inputs.
* **Implementation:**
  + Validates data lengths, formats, and cryptographic results (e.g., base64\_decode, verifyOTASignature).
* **Purpose:** Reduces vulnerabilities from invalid or malicious inputs.

**15. Timeouts and Retry Limits**

* **Description:** Limits on provisioning and retry attempts.
* **Implementation:**
  + Enforces a 20-minute provisioning timeout (STATE\_WAIT\_PROVISION).
  + Limits retries to 10 attempts with increasing delays (MAX\_RETRIES, retryState).
* **Purpose:** Mitigates brute-force and denial-of-service attacks.

**Conclusion**

The firmware integrates a comprehensive set of security features leveraging modern cryptographic techniques (AES-256, ECDH, ECDSA, SHA-256), secure communication protocols (MQTT over TLS), and robust device management (OTA, factory reset, WDT). These measures collectively protect against eavesdropping, tampering, unauthorized access, and ensure the integrity and authenticity of the device’s operations.