AdafruitRSA Library Documentation

Release 1.0

Brent Rubell

May 24, 2021

Contents

RSA implementation based on Sybren A. Stüvel's python-rsa pure-python RSA implementation.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Adafruit CircuitPython Logger Module

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

CHAPTER 2

Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI. To install for current user:

```
pip3 install adafruit-circuitpython-rsa
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-rsa
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-rsa
```

CHAPTER 3

Usage Example

Examples for this library are avaliable in the examples/ folder.

CHAPTER 4

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 5

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/rsa_simpletest.py

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

# Adafruit_CircuitPython_RSA Encryption/Decryption
import adafruit_rsa

# Create a keypair
print("Generating keypair...")
(public_key, private_key) = adafruit_rsa.newkeys(512)

# Message to send
message = "hello blinka"

# Encode the string as bytes (Adafruit_RSA only operates on bytes!)
message = message.encode("utf-8")

# Encrypt the message using the public key
print("Encrypting message...")
encrypted_message = adafruit_rsa.encrypt(message, public_key)

# Decrypt the encrypted message using a private key
print("Decrypting message...")
decrypted_message = adafruit_rsa.decrypt(encrypted_message, private_key)

# Print out the decrypted message
print("Decrypted Message: ", decrypted_message.decode("utf-8"))
```

6.2 Adafruit CircuitPython RSA API

RSA module

Module for calculating large primes, and RSA encryption, decryption, signing and verification. Includes generating public and private keys.

WARNING: this implementation does not use compression of the cleartext input to prevent repetitions, or other common security improvements. Use with care.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index

A