Data Entry Terminal

Overview

The Data Entry Terminal is a Rust-based CLI application designed to securely collect user data and generate cryptographic elements for secure transactions and identity verification. The application performs the following key functions:

1. Collects user information.
2. Generates an RSA key pair.
3. Produces a BIP39 mnemonic phrase.
4. Converts the mnemonic and user information into QR codes.

Prerequisites

- Rust programming language environment.
- Cargo package manager.

Installation

To get started, clone the repository and navigate to the project directory:

```
sh

git clone [Repository URL]
cd data-entry-terminal
```

Install the required Rust dependencies by running:

```
sh

cargo build
```

Usage

Run the program using Cargo:

```
sh

cargo run
```

Follow the on-screen prompts to enter user information.

Dependencies

The project relies on the following crates:

- `rsa`: For RSA key pair generation.
- `zeroize`: To securely zero out memory where sensitive data was stored.
- `tiny-bip39`: For generating BIP39 mnemonic phrases.
- `serde` and `serde_json`: To serialize data into JSON format.
- `qrcode`: For generating QR codes containing user data.
- `rand`: To generate cryptographic random numbers.

Ensure that the `Cargo.toml` file contains the correct versions of these dependencies:

```
toml

[dependencies]
rsa = { version = "0.9.3", features = ["pem"] }
zeroize = "1.6.0"
tiny-bip39 = "1.0.0"
serde = { version = "1.0", features = ["derive"] }
serde_json = "1.0"
qrcode = "0.12.0"
rand = "0.8.3"
```

Documentation & Schematics

RSA Cryptography (`rsa` crate)

- Official documentation: docs.rs/rsa
- Schematic Overview:
  - Key generation using RSA.
  - Serialization of the public key into PEM format.

BIP39 Mnemonics (`tiny-bip39` crate)

- Official documentation: docs.rs/tiny-bip39
- Schematic Overview:
  - Generation of a 24-word mnemonic phrase for secure backup.

JSON Serialization (`serde` and `serde_json` crates)

- Official documentation:
  - Serde: docs.serde.rs/serde
  - Serde JSON: docs.serde.rs/serde_json
- Schematic Overview:
  - Structuring of user data.
  - Serialization into JSON format for QR code generation.

QR Code Generation (`qrcode` crate)

- Official documentation: docs.rs/qrcode
- Schematic Overview:

    - Encoding user information and mnemonic phrase into QR codes.

Random Number Generation (`rand` crate)

- Official documentation: docs.rs/rand
- Schematic Overview:
    - Generation of secure random numbers for cryptographic operations.

Security Considerations

- Handle private keys and sensitive data securely.
- Transmit sensitive information over secure channels.
- Keep dependencies updated for security patches.
- Perform security audits before production use.

License

Specify your license here.

This documentation serves as a quick reference for developers to understand and integrate with the application. It includes installation instructions, usage guidelines, and a detailed overview of dependencies along with their documentation and schematic descriptions.