Prover-Verifier Protocol Documentation

Introduction

This documentation explains the Prover-Verifier Protocol implemented in the provided Python code using the Tkinter library for the graphical user interface. The protocol is designed to demonstrate the knowledge of a secret value (`x`) without revealing it directly. The protocol involves two parties: the Prover (Sender) and the Verifier (Receiver).

Classes and Methods

`ProverVerifierApp` Class

`__init_(self, root)`

Constructor method for initializing the GUI application. Creates the main window and sets the title. Initializes variables to store protocol parameters and values generated during the protocol.

`set_params(self)`

Method triggered when the "Set Parameters" button is clicked. Parses and validates input parameters (PRIMENO and generator). Generates the secret value (`secretVal`) and the Prover's value (`X`) based on the provided parameters. Updates the GUI to display the generated values.

`generate_values(self, generator, PRIMENO)`

Method to generate a secret value (`secretVal`) and the Prover's value (`X`) based on the given generator and PRIMENO. Returns `secretVal` and `X`.

`generate_Y(self)`

Method triggered when the "Generate Y" button is clicked. Parses and validates the input value (`y`) provided by the Prover. Generates the Verifier's value (`Y`) based on the Prover's value (`generator`) and `y`. Updates the GUI to display the generated `Y`.

`prove(self)`

Method triggered when the "Prove Knowledge" button is clicked. Parses and validates the input value (`c`) provided by the Verifier. Calculates the final value (`z`) and two intermediate values (`val1` and `val2`) based on the protocol. Compares `val1` and `val2` to determine if the Prover has proven knowledge of the secret value. Updates the GUI to display the calculated values and the result of the proof.

`prove_knowledge(self, PRIMENO, X, y, Y, c)`

Method to calculate the final value (`z`) and two intermediate values (`val1` and `val2`) based on the given parameters. Returns `z`, `val1`, and `val2`.

Main Block

`if __name__ == "__main__":`

Creates a Tkinter root window. Instantiates the `ProverVerifierApp` class. Starts the Tkinter event loop using `root.mainloop()`.

GUI Components

- **Entry Widgets:**
  - `PRIMENO`: Entry for the prime number used in the protocol.
  - `Generator`: Entry for the generator used in the protocol.
  - `y`: Entry for the Prover's input value.
  - `c`: Entry for the Verifier's input value.
- **Labels:**
  - Display labels for various protocol parameters and generated values.
  - Labels for different stages of the protocol, such as Sender (Prover), Receiver (Verifier), etc.
- **Buttons:**
  - "Set Parameters": Sets the protocol parameters.
  - "Generate Y": Generates the Verifier's value `Y`.
  - "Prove Knowledge": Initiates the proof of knowledge.
- **Output Labels:**
  - Display calculated values (`z`, `val1`, `val2`), the result of the proof, and any error messages.

Protocol Overview

1. **Set Parameters:**
   - Prover sets the prime number (`PRIMENO`) and generator.
2. **Generate Values:**
   - Prover generates a secret value (`secretVal`) and a value (`X`) based on the provided parameters.
3. **Prover (Sender) Stage:**
   - Prover shares the secret value (`secretVal`) and the value (`X`).
4. **Verifier (Receiver) Stage:**
   - Verifier receives `X` and requests the Prover to provide a value (`y`).
   - Prover generates and shares the value `Y` based on the input `y`.
5. **Prove Knowledge:**
   - Verifier provides a random value (`c`).
   - Prover calculates `z`, `val1`, and `val2` based on the protocol.
   - Verifier checks if `val1` equals `val2` to determine if the Prover has proven knowledge of the secret value.