ICS 505 Cryptography

Home Assignment 2

Dr. Muhammad Hataba, muhammad.hataba@giu-uni.de
TA. John Ehab, john.ehab@giu-uni.de

Zero-Knowledge Proofs (ZKPs) are advanced cryptographic protocols that allow one party to prove to another that they know a secret without revealing the secret itself. ZKP is particularly useful in authentication schemes, where it can preserve the privacy of user credentials while ensuring secure verification.

In this practical assignment, you will implement a basic one-factor authentication scheme using ZKP, where the username and password together serve as the authentication factor. You will use socket programming to build a client-server architecture, ensuring the password is never directly transmitted over the network. The objective of this assignment is to implement a privacy preserving protocol, by using the ZKP circuit to generate and verify proofs, then evaluate the efficiency and usability of the implemented protocol.

Step-by-Step Guide:

1. **Starter Code**: A basic client-server python template has been uploaded to the CMS, currently supporting a simple "hello" message exchange between the client and the server.
2. **Integration with ZKP Circuits:** Choose an open-source ZKP circuit from the resources provided below. Adjust it to integrate it with your project. Modify the client to send the password to the ZKP circuit to generate a proof, which should be sent along with the username to the server for verification instead of the password.
3. **Main Functionalities for the System:**

    a) Enrollment / Sign-up: to handle the commitment process. The client generates an initial proof (commitment) and sends it to the server to securely store it. Feel free to find out how this should be stored on the server (use any database of your choice to store the enrolled proofs alongside usernames).
    b) Authentication / Sign-in: to handle the verification process. The client generates the proof and sends it to the server to securely verify it, ensuring a secure and efficient authentication process.

4. **Report and Metrics**:

    - Collect efficiency and usability metrics, including:

        a) CPU and RAM utilization on both client and server.
        b) Packet propagation and processing time for the entire authentication process.

    - Write a report detailing your implementation steps and metrics analysis.

Open sources for ZKP Circuits:

- Gnark (Golang, most recommended): https://github.com/Consensys/gnark
- Circom (Rust): https://github.com/iden3/circom || https://docs.circom.io/ || https://victoryeo-62924.medium.com/zero-knowledge-rollup-using-circom-for-beginner-276ff1a96d5b
- Bellman (Rust): https://github.com/zkcrypto/bellman || https://trapdoortech.medium.com/zkp-deep-into-bellman-library-9b1bf52cb1a6

Submission Guidelines:

- Submit 1 zip folder containing 3 files: Report.pdf, Client.py, and Server.py. Ensure that your Python files are well-commented.
- You may choose to work on this assignment individually or in pairs.
- Submission Deadline: Sunday 22/12/2024 at 11:59 PM.
- Submission Link: https://forms.gle/3pKeh73323NAtpx76