



# APPLYING HOMOMORPHIC ENCRYPTION IN THE CLOUD

Jesús Antonio Soto Velázquez

jesus.antoniosv@gmail.com

Universidad Autónoma de Nuevo León, México



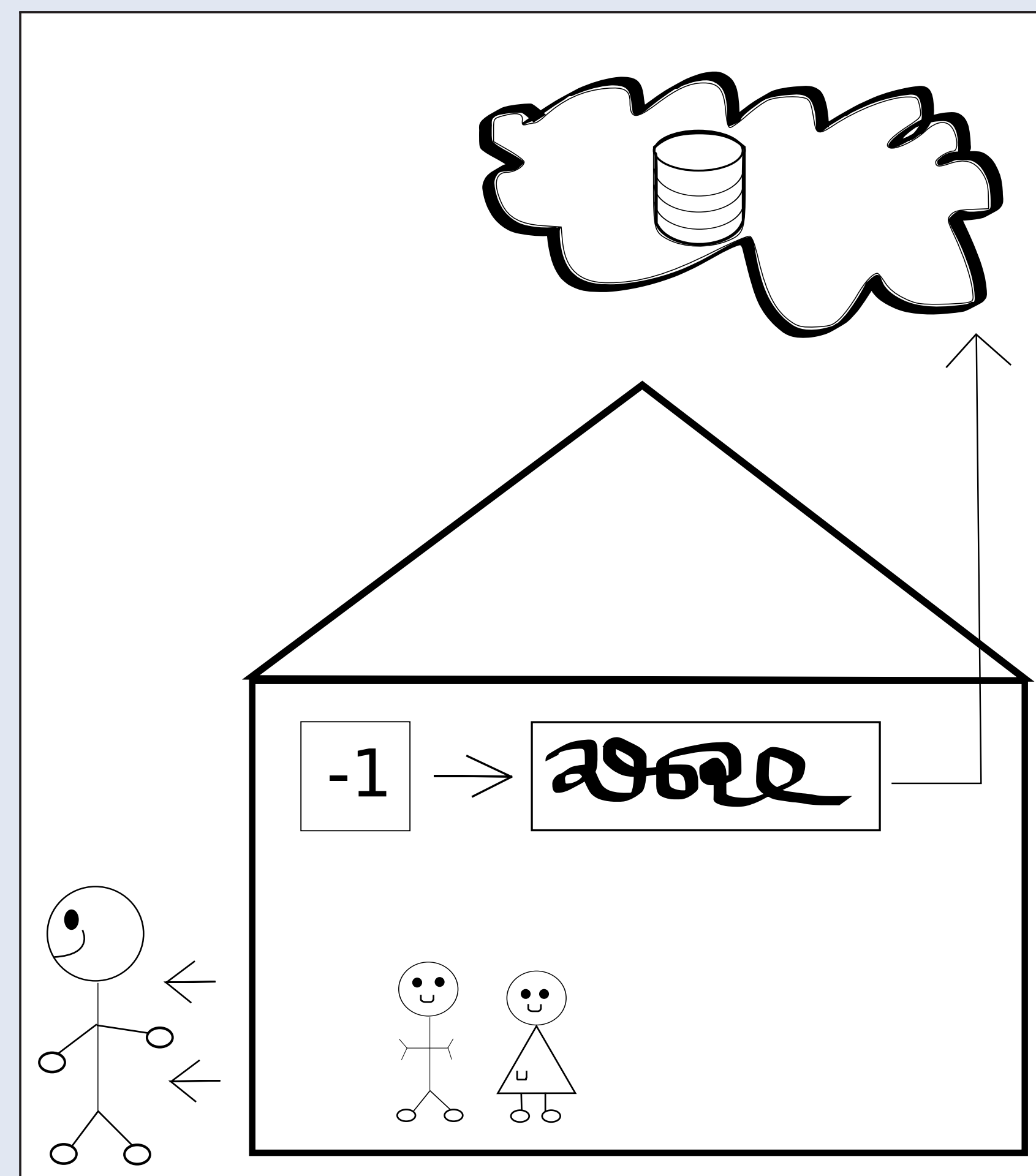
## Contribution

This work proposes the *design* and *implementation* of a client-server architecture based software that uses *HElib* to enable homomorphic encryption and perform computations on encrypted data. The software is used to address the situation in the *case study*.

## Objectives

- > **Establish** a client-server architecture where homomorphic encryption can be applied.
- > **Identify** which factors pose a challenge to apply homomorphic encryption in the cloud.
- > **Collect** performance data on the use of homomorphic encryption.

## Case Study



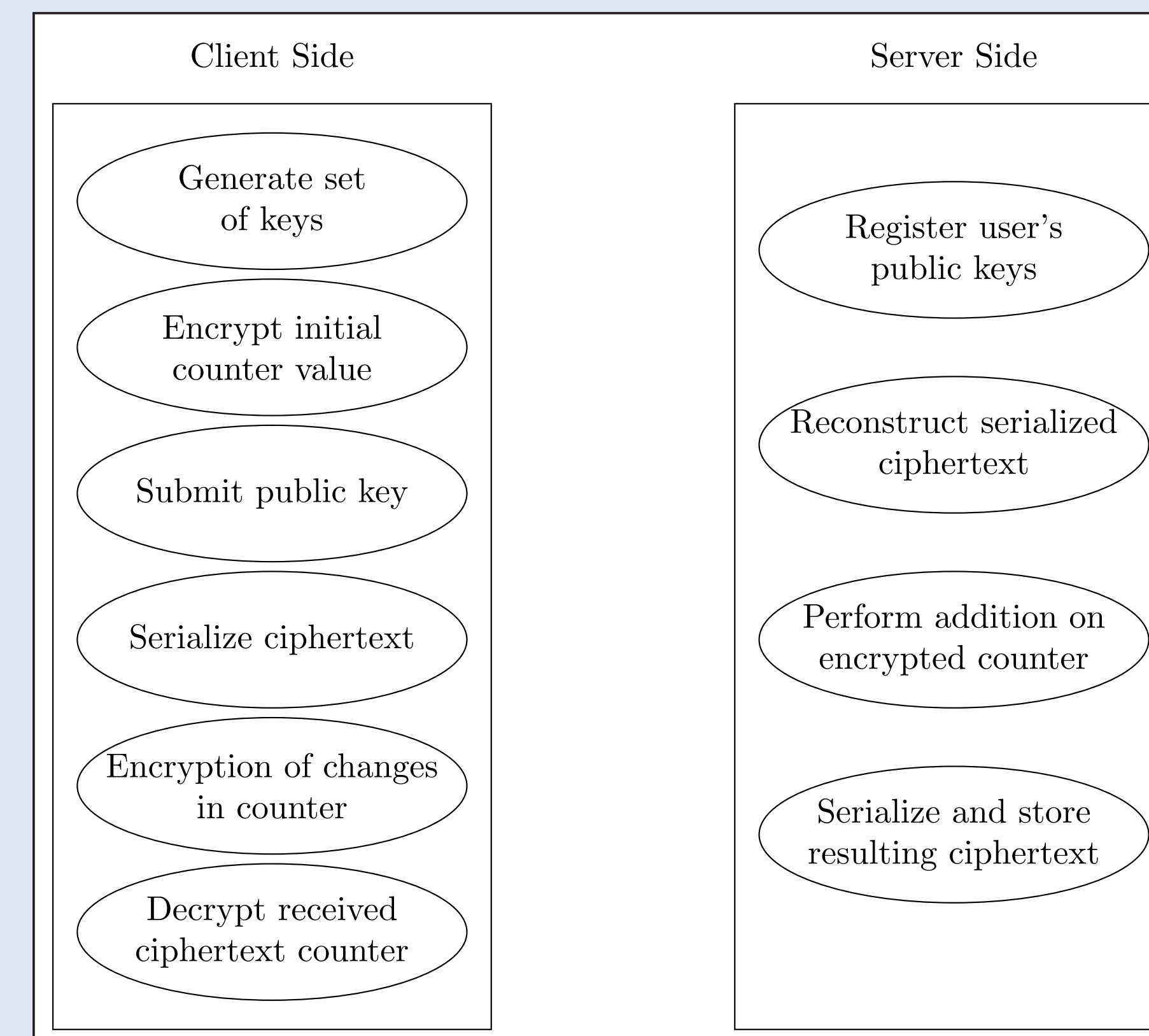
Consider a scenario where a household has an expected pattern of activity, so that the resident seeks to ascertain the number of people inside at any time.

As the resident chooses to **store the value of the counter in the cloud**, he quickly realizes he does not want others to learn of this value, not even the cloud service itself, as to prevent potential burglars to break in when the household is empty.

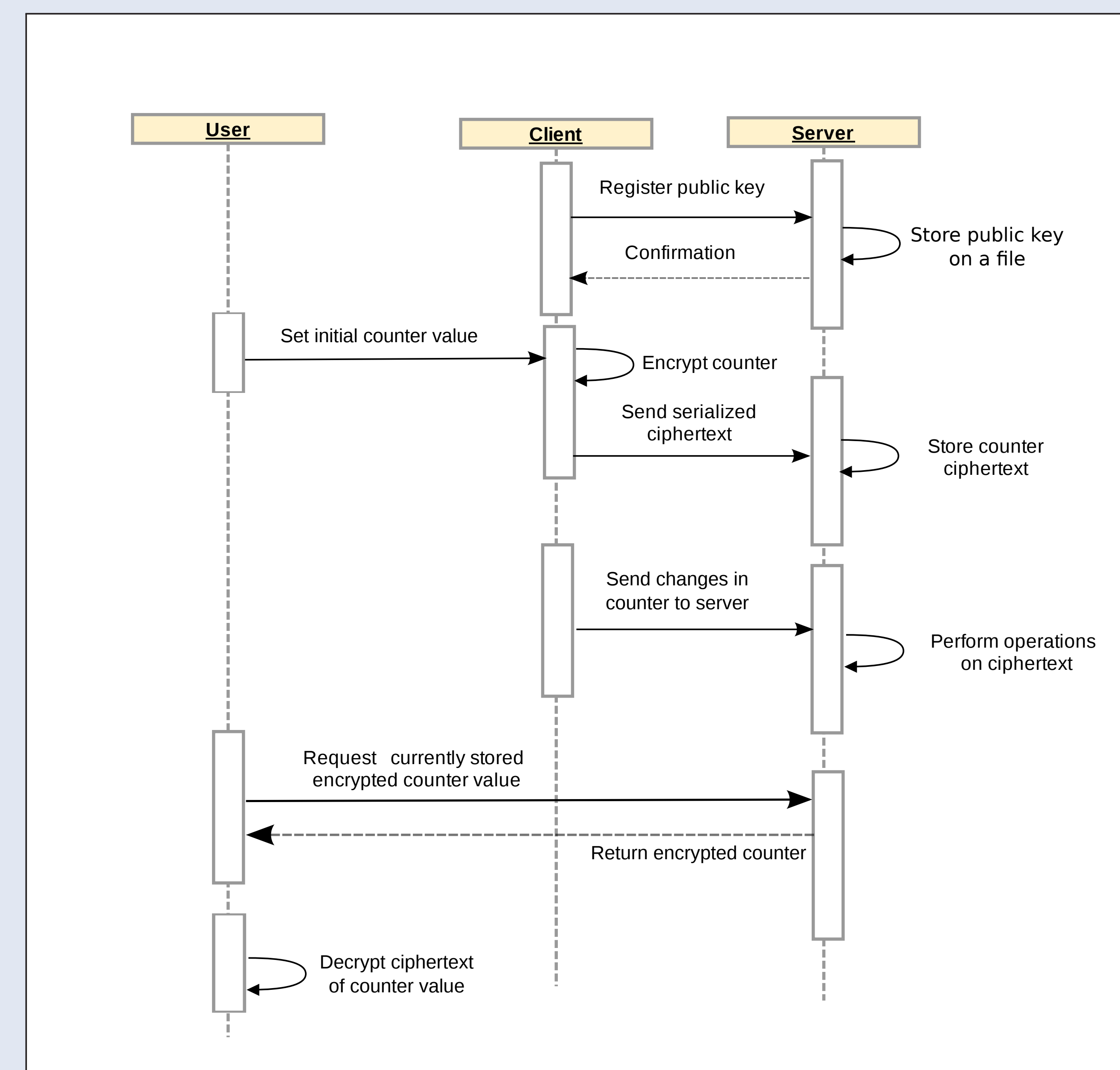
Therefore, a mechanism to ensure that the household activity data is available as well as protected needs to be put in place.

## Methodology

A client-server architecture was designed and implemented in C++ to address a solution for the presented case study. Homomorphic encryption is enabled by using HElib [1]. Relevant tasks were divided between **client** and **server**.



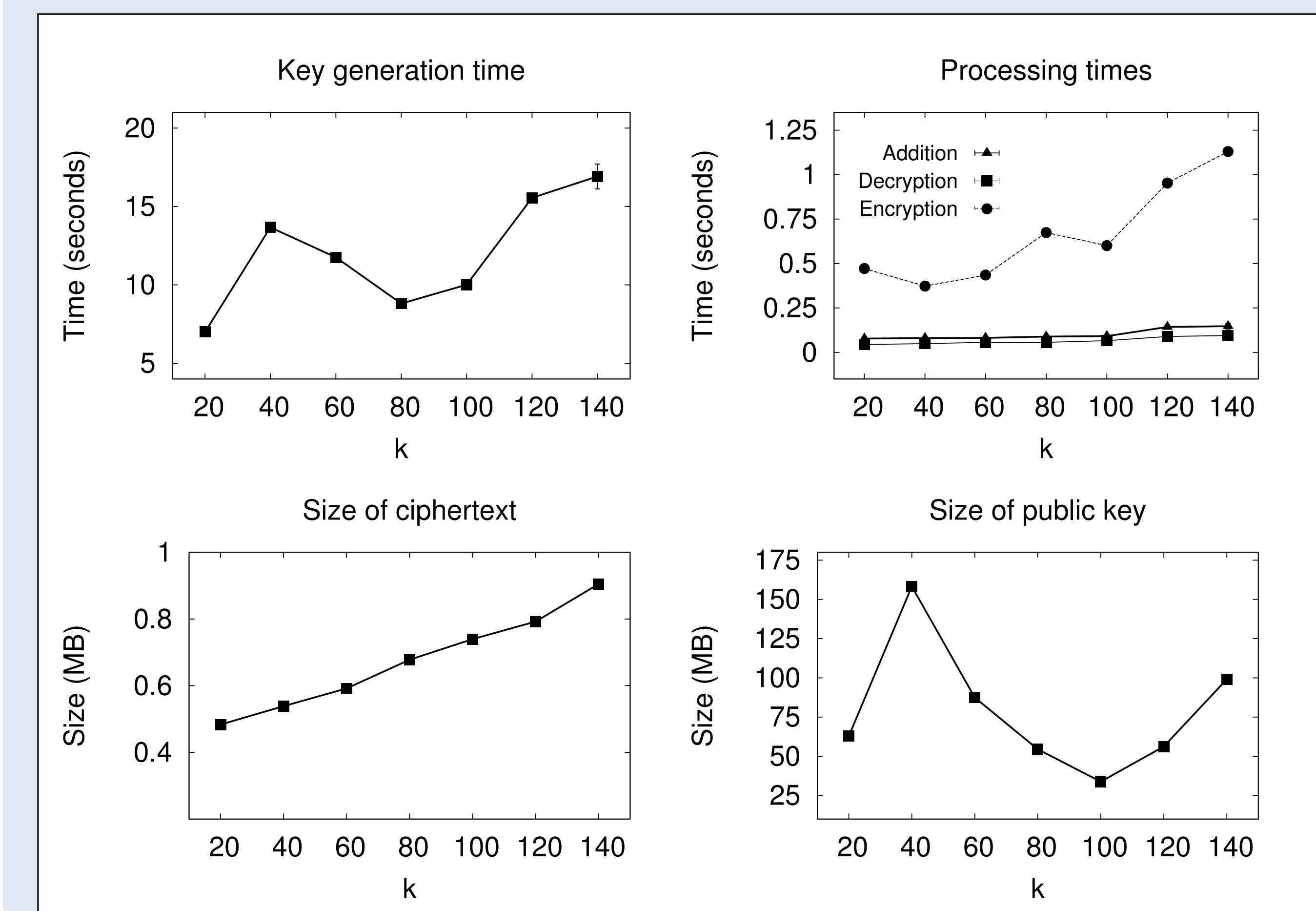
## Operation Flow



## Experimentation & Results

The performance of the system was evaluated by running 20 iterations with distinct values of a security parameter  $k$  needed for the homomorphic encryption scheme.

The system was evaluated in terms of ciphertext and key size, as well as the time needed for key generation and processing (addition, decryption, encryption).



## Future Work

- Perform more experimentation on the size of the ciphertext.
- Secure the communication channel between client and server.
- Design a database scheme to store public key-ciphertext pairs.

## References

- [1] Z. Brakerski, C. Gentry, V.Vaikuntanathan. HElib, 2014. <https://github.com/shaih/HElib>

## Acknowledgements

Special thanks to Prof. Elisa Schaeffer for providing the graphs of the results and Prof. Sara Garza for her support as primary advisor.

## Source Code

The code is publicly available at the following repository:  
<https://github.com/antoniosv/homomorphic-counter>