Client-Server System

Kali Hale and Craig Reinhart, Ph. D
Department of Computer Science, California Lutheran University

INTRODUCTION

- In an increasingly online world, security grows ever more important when communicating over the internet.
- The University of Michigan reported in 2019 that 80% of websites communicate with an encrypted connection (HTTPS)
- Many apps and software programs utilize a client-server system.
- In order to protect information traveling between a client and a server, encryption must be built into the system.

PURPOSE

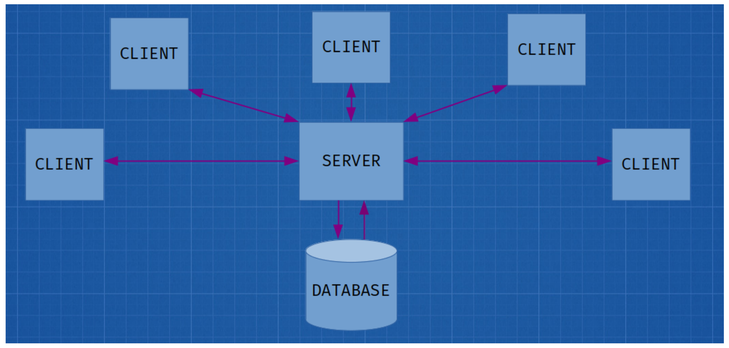
- When information is sent over the internet unprotected, that information can be intercepted. This includes information such as credit cards, social security numbers, etc.
- Additionally, people who want to communicate in countries such as Myanmar are placed at risk when their communications and internet activity are unprotected.
- This project serves to demonstrate that protecting information that travels over the internet can be done relatively easily, potentially protecting information from scammers and governments alike.

RESEARCH QUESTION

What is required to provide a secure experience for an end user of a client-server system?

METHODS

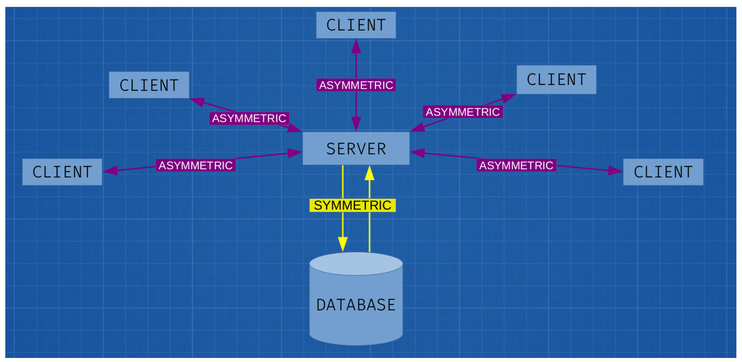
- For the base client-server system, I used the client-server system that I developed with a team in CSC 335: Software Engineering.
  - A client-server system is a system which has one or more clients which communicate with a single server. In this case, the server facilitates communication between clients and a central database.



METHODS (continued)

- I hypothesized that there are two points in a client-server system where information may be compromised – between the client and the server, and where the information is stored in the database.
- In order to protect information at these points, we must encrypt the data.
  - Encryption uses a *key* to encrypt and decrypt information (kind of like a cipher).
  - There are two types of encryption: *Symmetric* encryption and *asymmetric* encryption.
  - Symmetric encryption uses a single key to both encrypt and decrypt. This is useful if the same entity is both encrypting and decrypting the information, but it is not useful for sending information between two entities because the key must be communicated between the entities and there is a potential to be compromised in the process.
  - Asymmetric encryption uses a *public key* to encrypt and a *private key* to decrypt. The public key cannot be used to decrypt the information, only the private key can be used to decrypt. This means that if my friend wants to send me private information, I give them my public key in order to encrypt the information. They can then send me the information knowing that only I have the ability to read what they sent.
- Since the clients and server will be running on different computers in different locations, I chose to implement asymmetric encryption between the clients and the server. On the other hand, the database will only be accessed by the server, so symmetric encryption makes sense here.

ENCRYPTION IN CONTEXT:

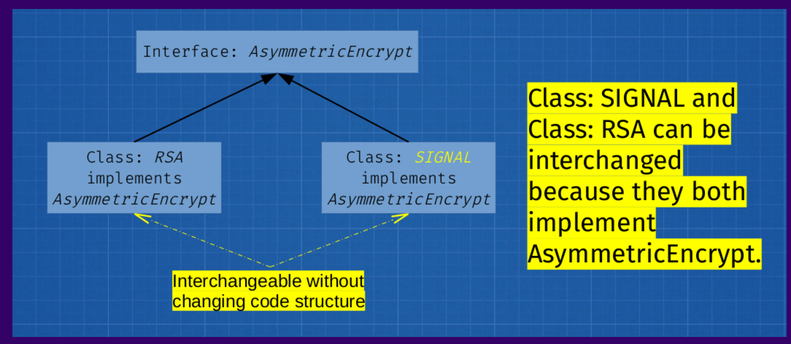


NEW RESEARCH QUESTION

Can multiple forms of encryption be implemented through a single, reusable interface?

RESULTS

- The answer to my new research question is yes, this is possible. An interface is like a “template” for a class – classes that use the interface may work differently from each other, but they still need to implement certain methods.
  - In this case, this means that any class that implements my interface Encrypt needs to be able to set the keys, encrypt, and decrypt the messages given to them.
  - This also means that any class that implements the interface is functionally interchangeable. If I initially used RSA encryption for my asymmetric encryption algorithm and I later chose to add the Signal Protocol, I would be able to swap those by changing only a single line of code.
- My project currently implements RSA encryption for the asymmetric encryption and AES encryption for the symmetric encryption.



FUTURE RESEARCH

- I am currently continuing this project as my capstone project.
- The focus of my capstone is to research and add the Signal Protocol into this project.
  - The Signal Protocol is widely considered to be the most secure encryption protocol currently in existence. It is used by most major messaging services, including WhatsApp, Telegram, Google Chat, and the Signal Messaging App.
- The Signal Protocol is open-source and there is a Java library available for use, so I am currently focusing on studying and understanding the code available before implementing it.