Client-Server System Framework

By Kali Hale

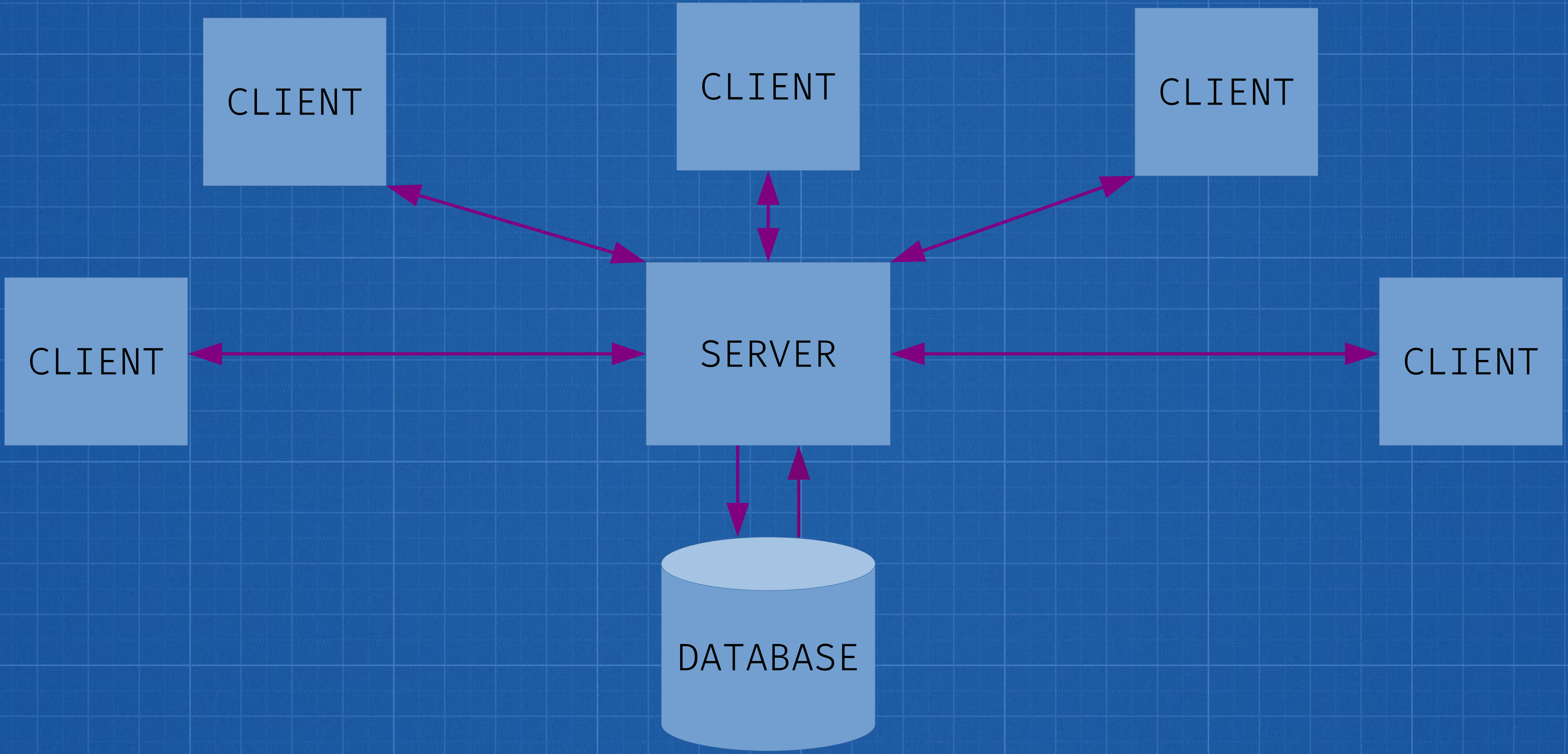
Keeping your digital information safe is essential.

- Unprotected information, such as credit cards, can be compromised at two points: When it’s sent over a network, or where it’s stored in a database.
- Protecting messages can save lives – for example, the ability to message securely in Hong Kong or Myanmar.

Question:

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

What is a Client-Server System?



Hypothesis

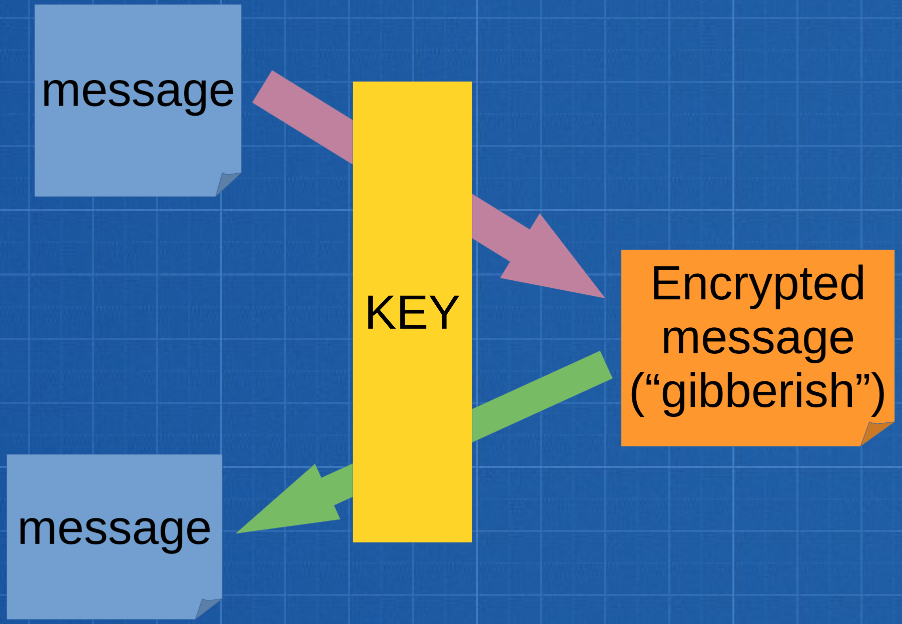
Two vulnerabilities exist:

1) Network communication
2) Stored information

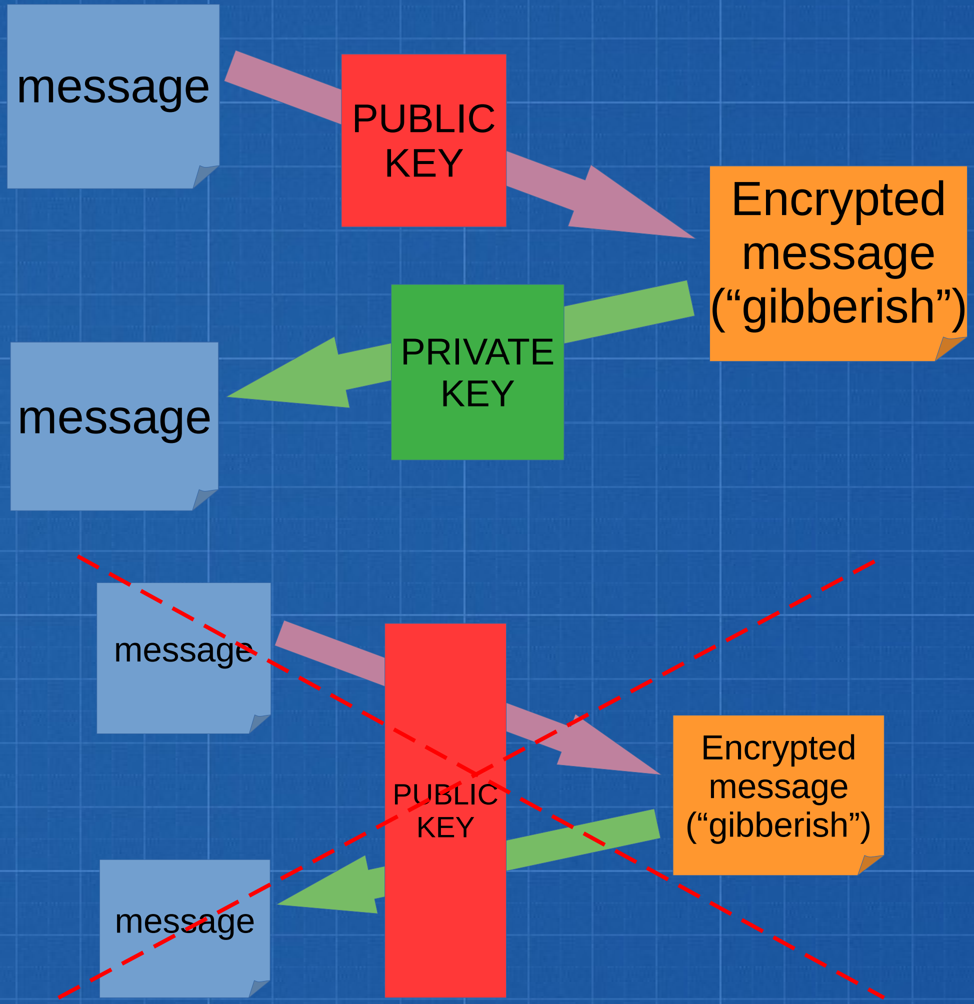
Solution: Encryption

- There are two types of encryption:
  - Symmetric encryption, which uses a single key to encrypt and decrypt information
  - Asymmetric encryption, which uses two keys: A public key to encrypt and a private key to decrypt

Symmetric Encryption



Asymmetric Encryption


message
PUBLIC KEY
Encrypted message (“gibberish”)
PRIVATE KEY
message
message
PUBLIC KEY
Encrypted message (“gibberish”)
message

Unencrypted

(postcards)

- Texting (SMS, MMS, “green text bubbles” on Apple)

Encrypted

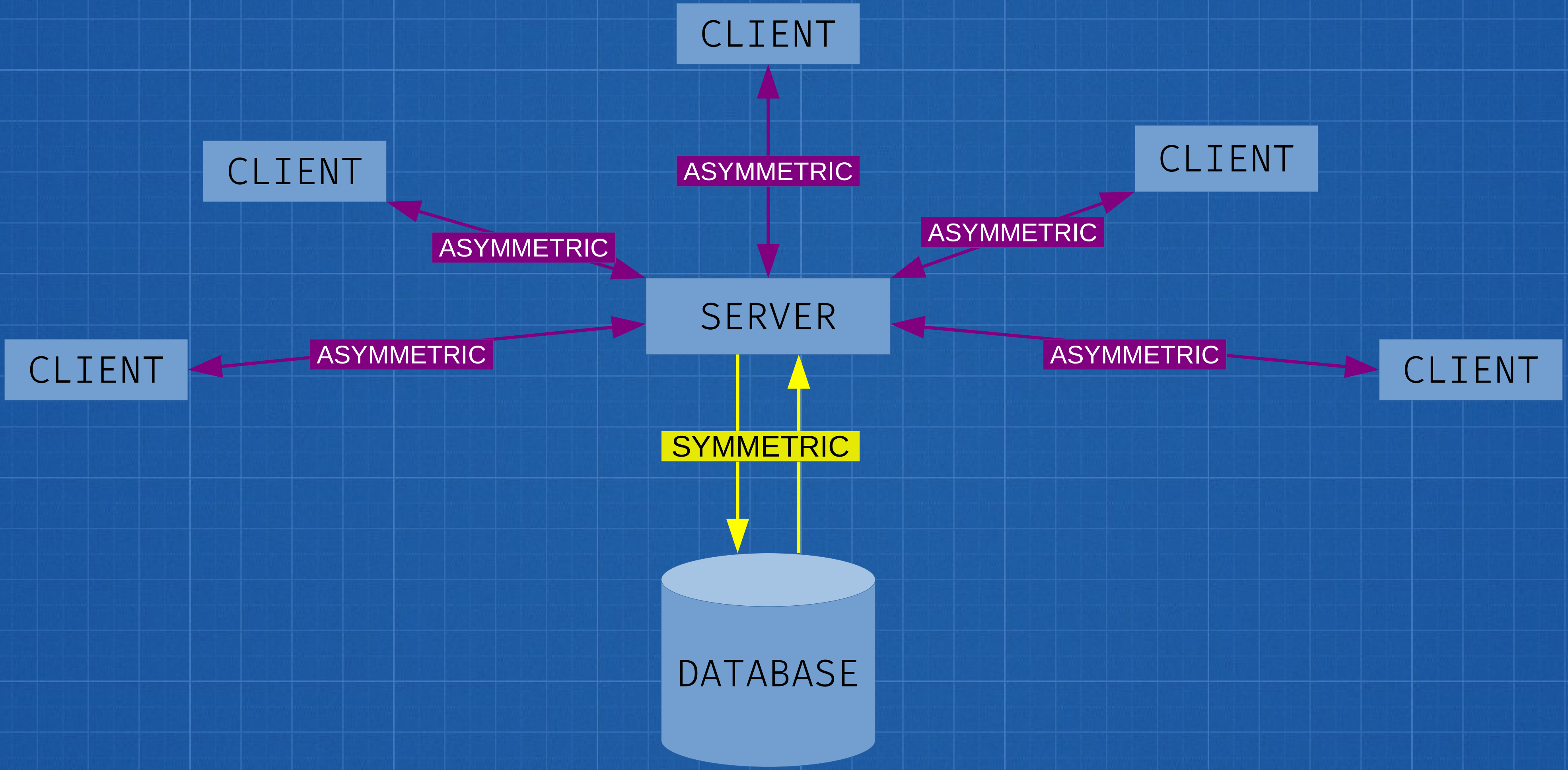
(sealed envelopes)

- Signal Messenger*
- WhatsApp*
- Facebook Messenger*
- RCS messages on Google Messenger*
- Telegram*
- iMessage**

*Uses Signal Protocol

**Uses multiple types of encryption, but can be hacked by those with the resources (therefore allowed in China, while the rest of these are banned)

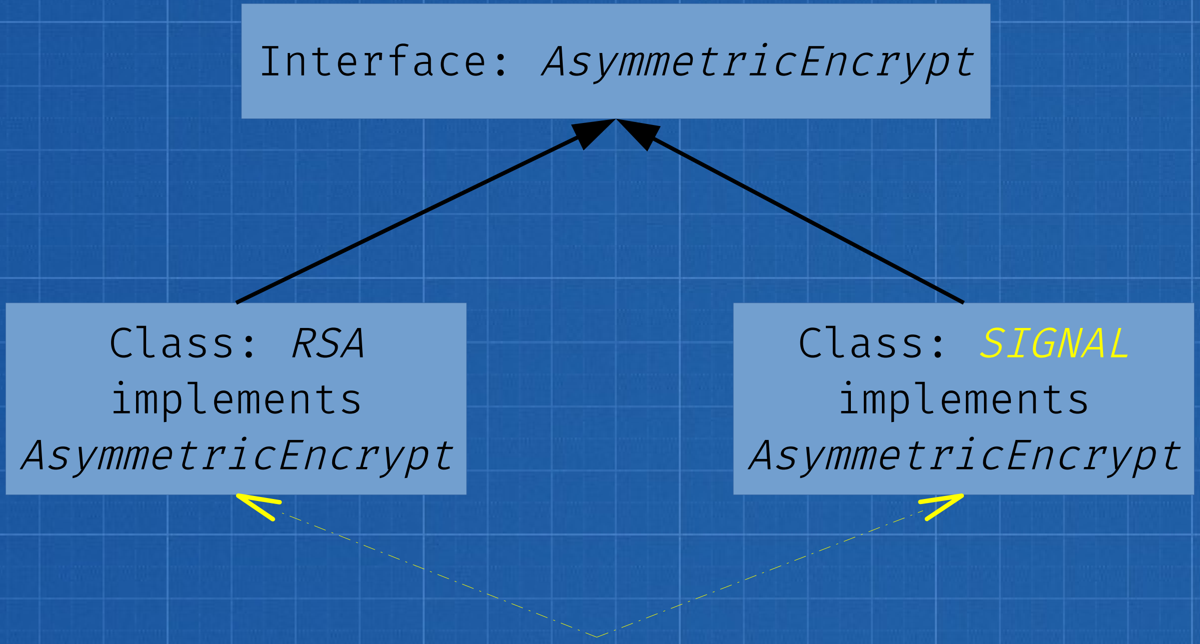
In context...


CLIENT
CLIENT
CLIENT
ASYMMETRIC
ASYMMETRIC
ASYMMETRIC
SERVER
CLIENT
ASYMMETRIC
ASYMMETRIC
CLIENT
SYMMETRIC
DATABASE

New question:

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

Interface example: Encrypt


Interface: *AsymmetricEncrypt*
Class: *RSA* implements *AsymmetricEncrypt*
Class: *SIGNAL* implements *AsymmetricEncrypt*

Interchangeable without changing code structure

Class: SIGNAL and Class: RSA can be interchanged because they both implement AsymmetricEncrypt.

Why use an interface?

- Encryption is constantly evolving; interfaces allow us to implement better and newer encryption without having to significantly alter code.
- Easier to maintain (keep up with new attacks and vulnerabilities) and safer for the end user (update an existing program vs. installing a new program).

Summary

- Symmetric encryption can be used to protect information stored in the database
- Asymmetric encryption can be used to protect information traveling between the client(s) and server
- Interfaces can be used to easily swap out types of encryption, making it easy to upgrade and maintain the system

Acknowledgments

I’d like to thank my mentor Dr. Reinhart, Jim and Sue Swenson for sponsoring the Swenson Summer Research Fellowship, California Lutheran University for the Summer Research program and computer labs, and my family and friends for supporting me in this endeavor.