Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth
---ECE549 Network Security---
Experiment

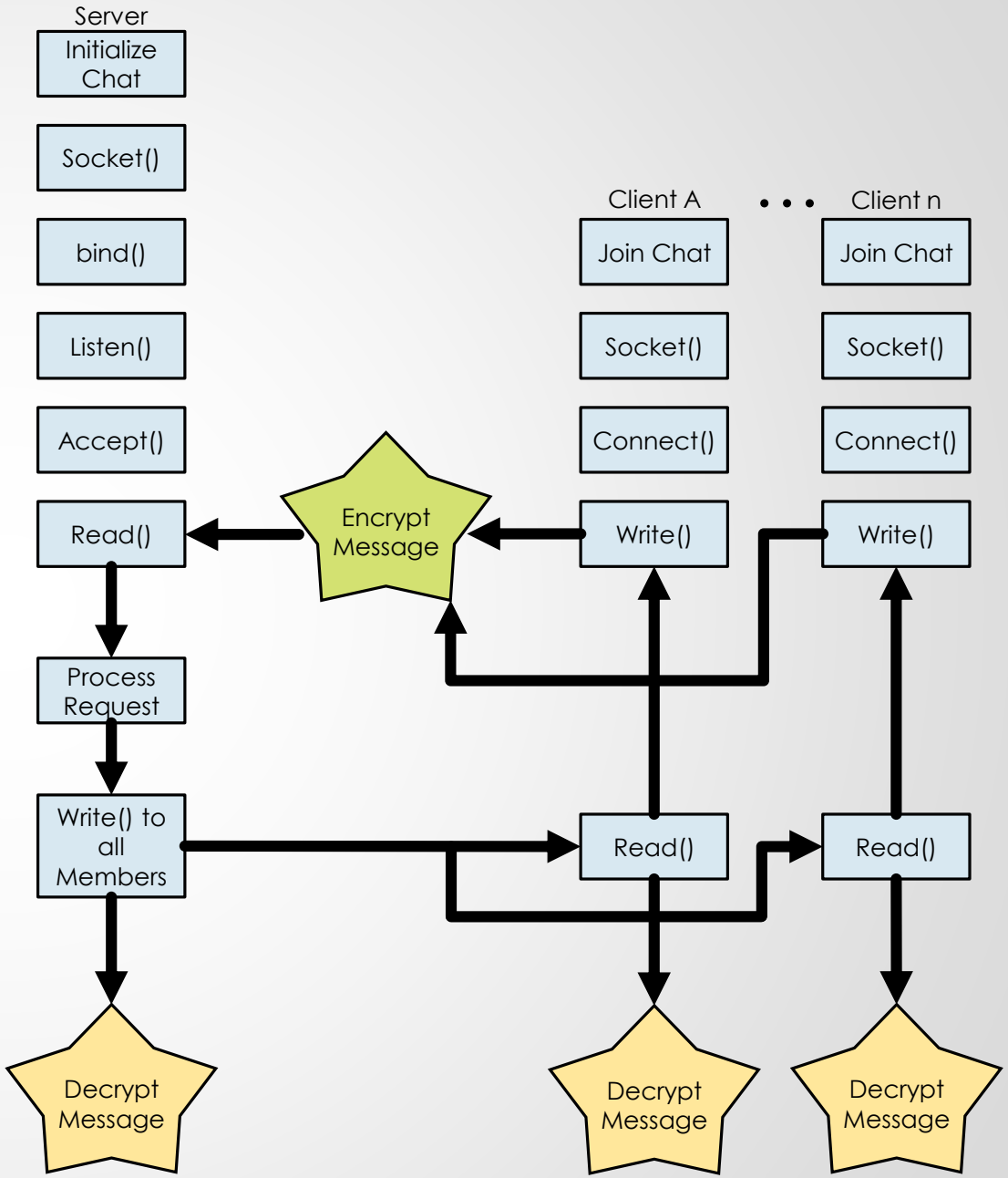
Team Lead: Cameron Whittle

Members: Devaj Ramsamooj, Peter McGrory, James McCarthy

Contents

Outline

- Why Is Messaging Security Important?
  - It protects sensitive information as it passes through unprotected/unsecure media
  - Anyone can see unprotected messages if they look hard enough
  - Dual Key Encryption relies on each member having a shared Public key and a personal Private Key
- Our team designed Amity Communications v1.0 in ECE369 – Computer Networks
- Amity Communication had no implemented security



Amity Communications v1.0 Recap

ACv1.0 – ECE369 Computer Networks

- Most group chats are unorganized and often a legitimate question or important information is often ignored or overlooked amongst other conversation. To better optimize the interface at which people view these chats and respond to them, we will maximize efficiency and make utilization smoother.
- Amity Communications is a group chat messaging system with three separated chats within each group server, which can be launched from within the application.
- Prioritize organization and simplicity in users' everyday lives.
- Built on a client/server framework in C# using windows forms for GUI design

TCP

- Connection Oriented
- Slower than UDP
- More Reliable and has ACK/NACK framework
- Server must be running before client is connected

Introduction

Goals

- Implement an effective encryption algorithm into Amity Communications
- Analyze the different encryption methods available
- Deign an efficient and secure way to handle encryption keys
- Showcase the implementation's effectiveness

Purpose

- Upgrade our chat server to learn how encryption algorithms function in a group chat environment
- To continue work on a previously designed project, getting to see its faults and continue to build it from start to finish

Problem Statement

Scope

- To design a prototyped secure group communication system, inspired by ECE369's Amity Communications
- The system must use an encryption algorithm learned in class and must feature a key distribution method.
- Designed for the everyday use of busy college students (as ACv1.0 was).

C.I.A.

- Confidentiality – Cannot decrypt without the respective private key, public keys are used for encryption.
- Integrity – Cannot change data without the session key to encrypt or the senders private key to decrypt. Session key changes in every chat.
- Availability - Each user profile is protected by a login password, and must TCP connect with the server to join that chat.

Claim

- To implement an RSA encryption algorithm and additional security measures into an existing group chat messenger, creating Amity Communications v2.0.

Market Value

- Market Value is open source for students.

Background

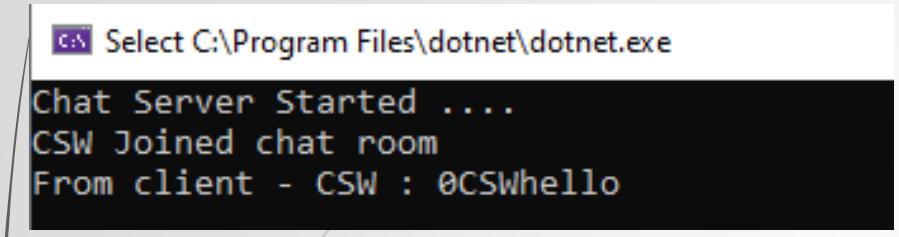
State-of-the-Art

- A user friendly, simple, secure, group chat messenger that features 3 divided sub-chats, adding organization to cluttered chats.
- Dual-Key encryption where a session key is used for encryption and the server uses public keys to send each person a personalized message that must be decrypted using private keys.

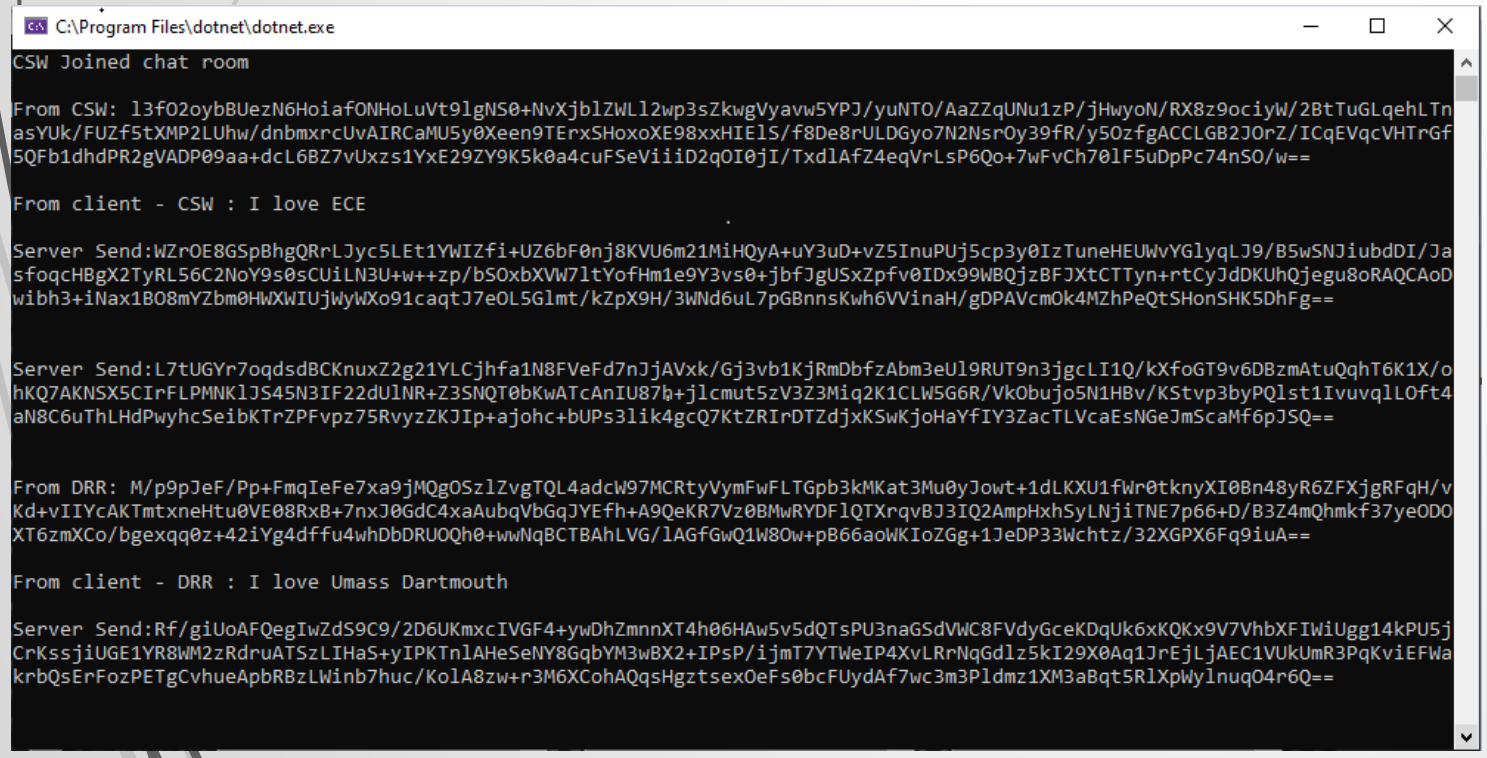
Applicable Background Knowledge

- Computer Networks – TCP Connections, Client & Server Messaging, Group Chats.
- Network Security – Password protecting, sending encrypted data, Cryptography, Dual-key Encryption.

Defending Ability (v1.0 vs v2.0)



Amity Communication v1.0 –
All sent data is viewed and can be intercepted in plain text



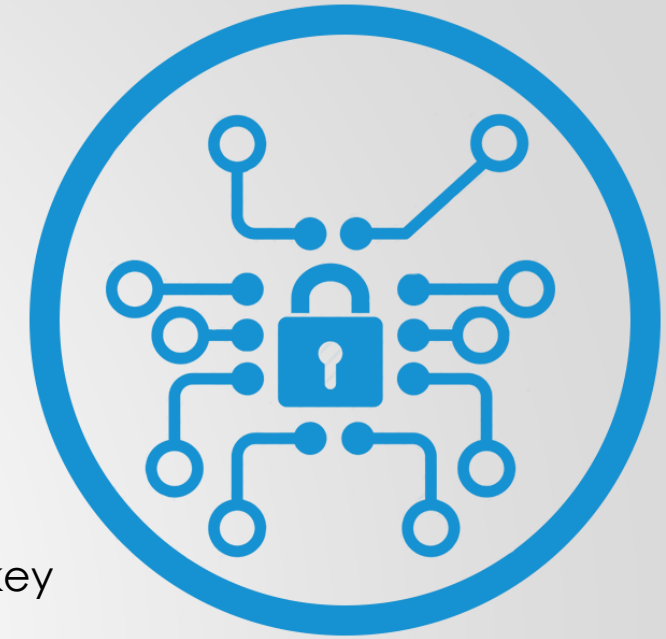
Amity Communication v2.0 –
All sent data is encrypted and can only be decrypted and viewed as plaintext with the private keys

Challenges

Issues Faced

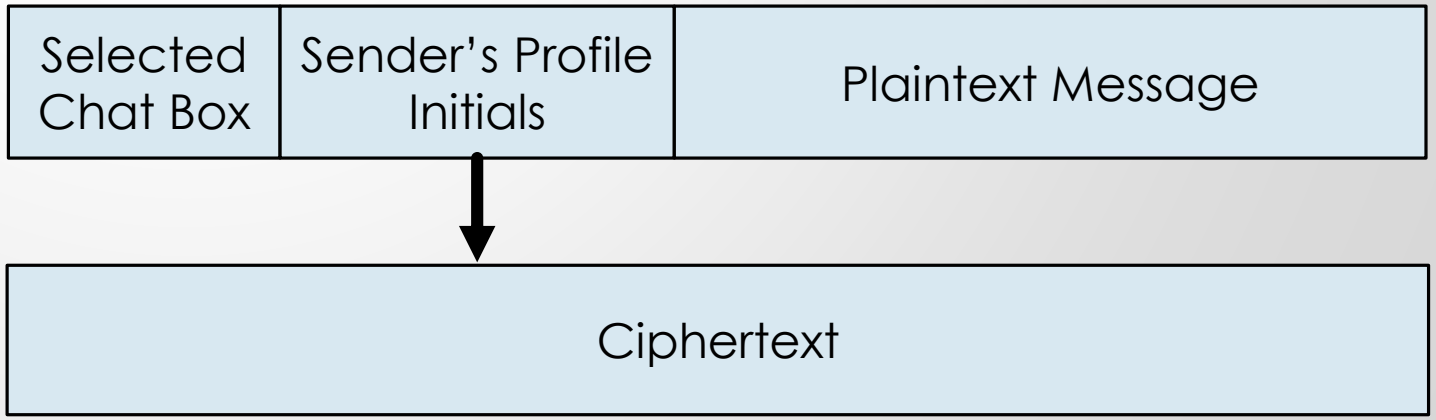
- Incorporating Encryption and Decryption to Amity Communications, lots of changes had to be made to the previous design
- Decrypting the relayed message from the server to each client
- Storing all the public keys in the server along with each TCP connection
- Having the Server re-encrypt the data using public keys
- Deciding on the security framework & how keys would be distributed among users
- Trying to make this as secure as possible with our limited experience and knowledge

Work Conducted

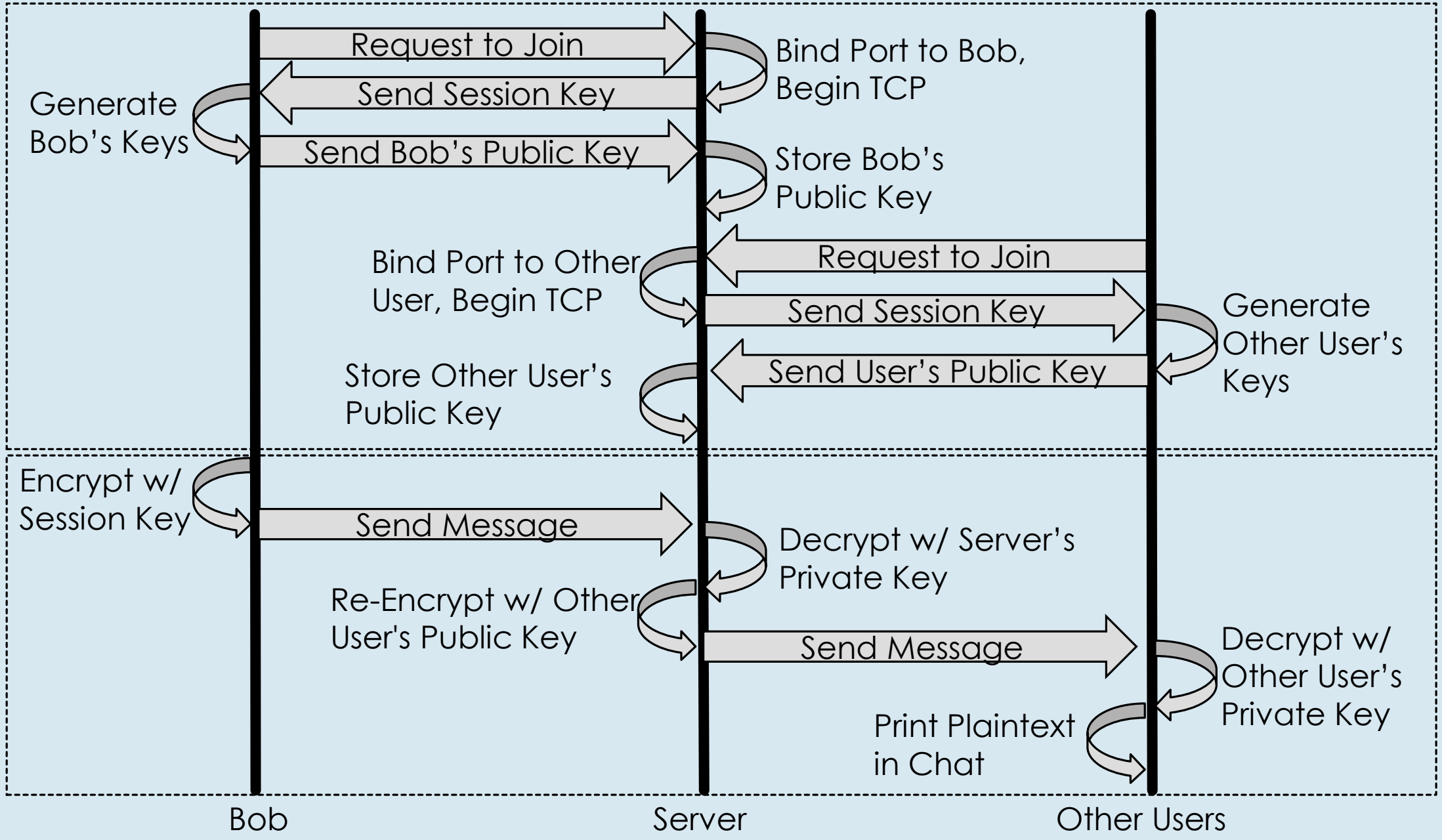


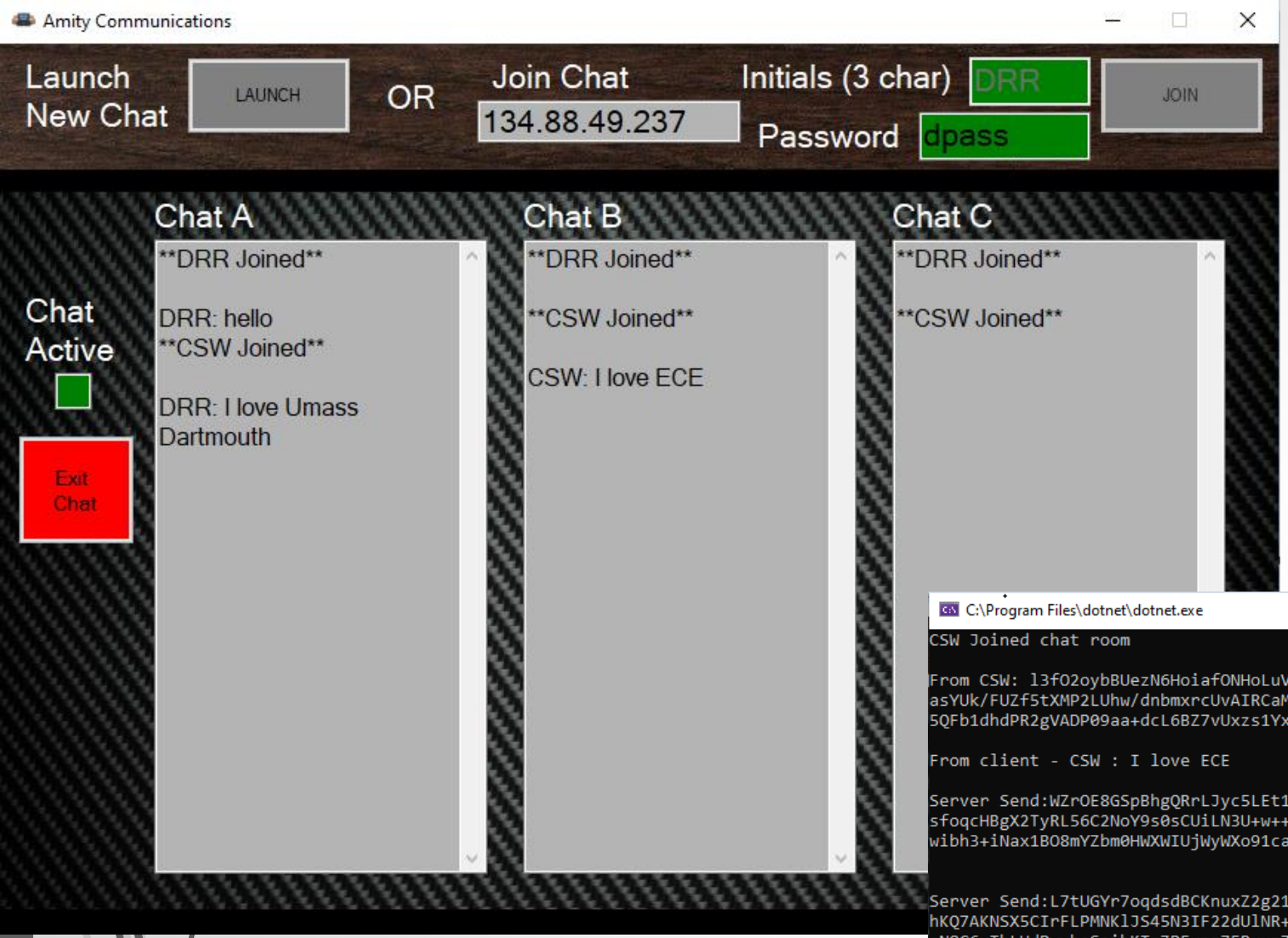
Applied Features

- User Profiles that are password protected
  - Initials with a set password
- RSA data encryption for sent messages to/from server
  - It's hard to crack through mathematics with a 2048 bit key
- Group message key distribution
  - Server's session key and private key
  - User's dual-key (Private and Public)
  - Key distribution architecture **Shown on next slide**

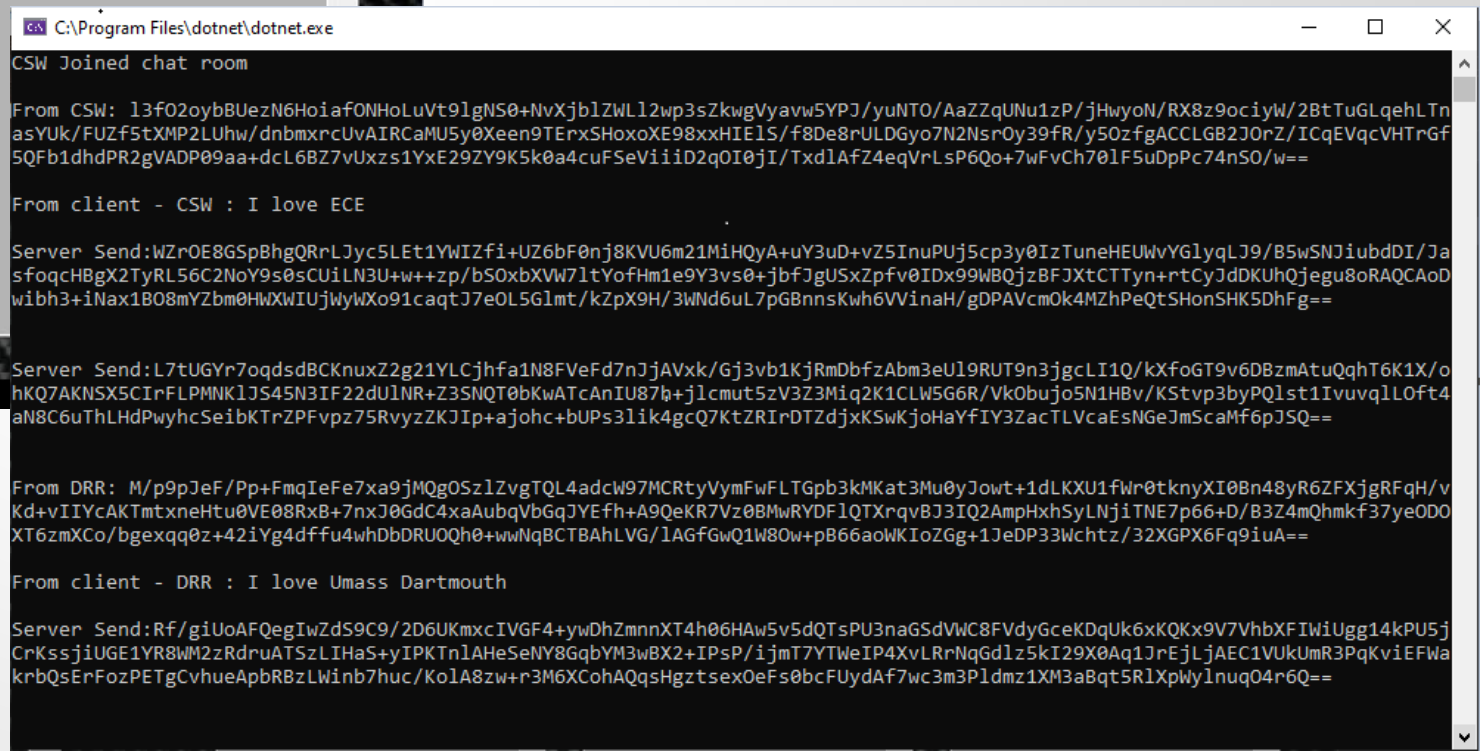


Key Distribution Architecture




Amity Communications
Launch New Chat
LAUNCH
OR
Join Chat
134.88.49.237
Initials (3 char)
DRR
Password
dpass
JOIN
Chat A
DRR Joined
DRR: hello
CSW Joined
DRR: I love Umass Dartmouth
Chat B
DRR Joined
CSW Joined
CSW: I love ECE
Chat C
DRR Joined
CSW Joined
Chat Active
Exit Chat

Results


C:\Program Files\dotnet\dotnet.exe
CSW Joined chat room
From CSW: l3fO2oybBUezN6HoiafONHoLuVt9lgNS0+NvXjblZWLl2wp3sZkwgVyavw5YPJ/yuNTO/AaZZqUNu1zP/jHwyoN/RX8z9ociyW/2BtTuGLqehLTn
asYUk/FUZf5tXMP2LUhw/dnbmxrcUvAIRCaMU5y0Xeen9TErxSHoxoXE98xxHIElS/f8De8rULDGyo7N2NsrOy39fR/y5OzfgACCLGB2JOrZ/ICqEVqcVHTrGf
5QFb1dhdPR2gVADP09aa+dcL6BZ7vUxzs1YxE29ZY9K5k0a4cuFSeViiD2qOI0jI/TxdlAfZ4eqVrLsP6Qo+7wFvCh70lF5uDpPc74nSO/w==
From client - CSW : I love ECE
Server Send:WZrOE8GSpBhgQRrLJyc5LEt1YWIZfi+UZ6bF0nj8KVU6m21MiHQyA+uY3uD+vZ5InuPUj5cp3y0IzTuneHEUWvYGlyqLJ9/B5wSNJiubdDI/Ja
sfoqcHBgX2TyRL56C2NoY9s0sCUiLN3U+w++zp/bSOxbXVW7ltYofHm1e9Y3vs0+jbfJgUSxZpfv0IDx99WBQjzBFJXtCTTyn+rtCyJdDKUhQjegu8oRAQCAoD
wibh3+iNax1BO8mYZbm0HWXWIUjWyWXo91caqtJ7eOL5Glmt/kZpX9H/3WNd6uL7pGBnnsKwh6VVinaH/gDPAVcmOk4MZhPeQtSHonSHK5DhFg==
Server Send:L7tUGYr7oqdsdBCKnuxZ2g21YLCjhfa1N8FVeFd7nJjAVxk/Gj3vb1KjRmDbfzAbm3eUl9RUT9n3jgcLI1Q/kXfoGT9v6DBzmAtuQqhT6K1X/o
hKQ7AKNSX5CIrFLPMNKlJS45N3IF22dUlNR+Z3SNQT0bKwATcAnIU87h+jlcmut5zV3Z3Miq2K1CLW5G6R/VkObujo5N1HBv/KStvp3byPQlst1IvuvqlLOft4
aN8C6uThLHdPwyhcSeibKTrZPFvpz75RvyzZKJIp+ajohc+bUPs3lik4gcQ7KtZRIrDTZdjxKSwKjoHaYfIY3ZacTLVcaEsNGeJmScaMf6pJSQ==
From DRR: M/p9pJeF/Pp+FmqIeFe7xa9jMQgOSzlZvgTQL4adcW97MCRtyVymFwFLTGpb3kMKat3Mu0yJowt+1dLKXU1fWr0tknyXI0Bn48yR6ZFXjgRFqH/v
Kd+vIIYcAKTmtxneHtu0VE08RxB+7nxJ0GdC4xaAubqVbGqJYEfh+A9QeKR7Vz0BMwRYDFlQTXrqvBJ3IQ2AmpHxhSyLNjiTNE7p66+D/B3Z4mQhmkf37yeODO
XT6zmXCo/bgexqq0z+42iYg4dffu4whDbDRUOQh0+wwNqBCTBAhLVG/lAGfGwQ1W8Ow+pB66aoWKIoZGg+1JeDP33Wchtz/32XGPX6Fq9iuA==
From client - DRR : I love Umass Dartmouth
Server Send:Rf/giUoAFQegIwZdS9C9/2D6UKmxcIVGF4+ywDhZmnnXT4h06HAw5v5dQTsPU3naGSdVWC8FVdyGceKDqUk6xKQKx9V7VhbXFIWiUgg14kPU5j
CrKssjiUGE1YR8WM2zRdruATSzLIHaS+yIPKTnlAHeSeNY8GqbYM3wBX2+IPsP/ijmT7YTWeIP4XvLRrNqGdlz5kI29X0Aq1JrEjLjAEC1VUkUmR3PqKviEFWa
krbQsErFozPETgCvhueApbRBzLWinb7huc/KolA8zw+r3M6XCohAQqsHgztsexOeFs0bcFUydAf7wc3m3Pldmz1XM3aBqt5RlXpWylnuqO4r6Q==

DEMO

Conclusion

Lessons Learned

- How to implement dual-key encryption
- How the RSA algorithm can be used on a group chat messenger
- How easy it is to intercept and enact replay attacks on an un-protected messenger
- There are many different ways to defend and attack when it comes to Network Security
- A recap and improved understanding of computer networks and TCP communication
- How important data and message security is

References

1. "Easy Guide to Encryption and Why It Matters." *Easy Guide to Encryption and Why It Matters | Amnesty International*, https://www.amnesty.org/en/latest/campaigns/2016/10/easy-guide-to-encryption-and-why-it-matters/.
2. Upganlawar, Sooryavanshi, Bhosle, et al. "Digital Signatures for Secure Communication." *Digital Signatures for Secure Communication - IEEE Conference Publication*, 21 June 2018, https://ieeexplore.ieee.org/document/8389291.

Thank You

Questions?