# CS 4153/6153: Computer Security: Project #3

**Ryan McCarthy and Duc Tran**

Friday May 1st, 2015

# Contents

# Introduction

The goal of Project 3 was to implement the Oblivious Transfer Protocol. The catch was to be sure and implement the protocol in a manner that ensured no cheating could occur. We were given the majority of the code to implement the protocol, only being left with implementing half of the protocol and inserting various countermeasures to detect any scenarios in which the other party attempts to cheat.

# Methodology

## Oblivious Transfer Protocol Implementation

To implement the OTP, we looked at the diagram included with the project and went through it step by step. By looking at the code already completed for Alice's portion of it and the help from the diagram, we were able to successfully figure out how to implement the OTP.

## Code Analysis for Flaws

To try to stop Alice from cheating, we looked at the OTP implementation as if we were trying to exploit it ourselves. We looked for places we could have sent faulty information or leveraged known information to produce a desired output.

# Protocol and Cheat Overview

## Protocol Overview

In Common.java, the encryptKey algorithm was flawed. The algorithm was not inserting random bytes correctly. We fixed this by adjusting the block length so it would be correct and adjusted the secureRandom to randomize the key.

## Cheat Overview

Some of the various checks against cheating we performed are:

- If both private and public key pairs are the same
- If both public keys sent are the same
- If both private keys are the same
- If the private key actually decrypts the information
- If the keys are formatted as anticipated

# Implementation

The hardest part of the project was finding the issue with Common.java. The encryption key generated only had 16 bytes of entropy which could allow an attacker to brute force the key, unlock both public keys and correctly determine the winning key. After we figure that out, implementing a fix was not overly difficult.