Secure Communication Protocols

Abstract

In this assignment, we implement a secure communication protocol consisting of three consecutive steps. First, we use Diffie-Hellman key exchange to craft a shared private key between two actors using public key cryptography. We then use an implementation of the Blum Blum Shub algorithm to generate a cryptographically secure pseudo-random shared private key, using the previously exchanged key as seed. Finally, we use the generated random key to encrypt and decrypt messages between the two actors using 128-bit AES ciphers.

1 Introduction

Our main goal is to be able to communicate secret data over an unsecure communication channel. To do so, we have to encrypt our data before sending it, and decrypt it after recieving it, so that it only appears encrypted on the communication channel. To perform these encryptions, we want to use a symmetric cipher allowing us to efficiently compute ciphertext from plaintext and vice versa. Before using symmetric encryption, we have to solve the key exchange problem, because we still need to communicate the private key between the two actors over an unsecure channel. One answer to this first problem is to use public-key cryptography, and most specifically the Diffie-Hellman key exchange protocol.

1.1 Diffie-Hellman

This key scheme method allows us to create a shared private key by only communicating public data. It relies on the computational difficulty of computing discrete logarithms. It has been formulated in 1976 by Whitfield Diffie and Martin Hellman. We use it to generate a first shared private key between two actors, Alice and Bob.

1.2 Blum Blum Shub

1.3 Advanced Encryption Standard

2 Design and Implementation

3 Test Results

4 Discussion

5 Conclusion