Benha University
Faculty of Engineering at Shoubra
Electrical Engineering Department
Computer Systems Engineering
Academic Year 2021/2022

Secure large-size data sharing over unsecure channels

Submitted By
Ahmed Amr Mohamed
Ahmed said Abd Al Raheam
Aya Nashaat
Mohamed Atef Ali
Mohamed Said Abd Al Raheam

Supervised By
Prof: Sahr

Tables of content

Abstract

in this project we demonstrate how two or more peers can share large-sized and secret information with each other without any prior communication and over an insecure channel. we start by introducing two main cryptography algorithms with their relevant advantages and disadvantages and why neither category is enough by itself to solve the problem. We move on to a suggested solution to our problem. Then we propose a relaxed version of the problem to reduce the projects scope to the core issue. Then we discuss the proposed solution in more detailed manner.

1.1 Introduction

Symmetric algorithms like DES and AES provide fast and secure way to exchange data between peers. The only requirement is to have a shared a key that is only shared between the intended peers. Which requires a secure channel to establish such code, which is not always feasible.

Asymmetric algorithms on the other can share data over an unsecure channel between peers. The downside is that these algorithms are typically slow and resources demanding in terms of computational power and transmission. Which makes them impractical for lengthy back and forth data exchange or sending large-sized data.

Using only one of these solutions is either not efficient or not secure for transmitting large-sized data over an unsecured channel.

1.2 problem statement

How to share large sized data over an unsecure channel between two or more peers without any prior communication between them?

1.3 Suggested Solution

Hybrid solutions offer the best of both worlds. It uses the asymmetric algorithm to establish a shared-secret key between the peers over the unsecure channel, then uses that shared key as the secret code for the symmetric algorithms to efficiently encrypt and decrypt the large sized data.

In this paper we implement a relaxed version of this system to focus only on the cryptography part and not the networking part of the problem.

We start by discussing the relaxed version. Then we discuss the workflow of the shared key establishment. Then we discuss the workflow of the actual data encryption and decryption.

1.4 Relaxed version of the system

We only consider to peers: the sender peer (SP) and the receiver peer (RP). We use the local OS directories to simulate three channels: the sender channel, the receiver channel and the public channel. We consider the first two channels to be secure as they represent the content that only their respective peer has access to. The third channel represent the medium onto which the data is transmitted, we can assume that both peers and any third-party peer - attacker or otherwise -has access to all data in this channel.

1.5 Key Establishment

The RP use the RSA algorithm to generate two keys, a private and a public key. Messages encrypted with the public key can only be decrypted using the private key. The public key is broadcasted over the public channel. the SP generates a random code referred to as the session code, this code is then encrypted using the public key and broadcasted over the public channel. the SP uses its private key to decrypt the encrypted session key.

1.6 AES

The session key is then used as a secret code, since it is only known to the SP and RP. The secret code is used as the encryption and decryption key by both peers. For efficiency reason the encrypted session key is broadcasted at the same time as the encrypted message. The RP uses the private key to decrypt the session key, the uses the session key and the inverse AES to decrypt the message.

References

Understanding Cryptography by Christof Paar and Jan Pelzl

Daemen, Joan; Rijmen, Vincent (March 9, 2003)

Kaliski, Burt (October 22, 1997). "Growing Up with Alice and Bob: Three Decades with the RSA Cryptosystem".

pycryptodome https://pycryptodome.readthedocs.io/en/latest/src/license.html#public-domain