Decentralized Systems Engineering Fall 2019

Project Phase 1

Hrusanov Aleksandar, Lanzrein Johan, Rinaldi Vincent

Project Motivation

In the implementation of Peerster, nodes can exchange private messages around the network. However there is no protection against malicious nodes who would eavesdrop or modify the private messages. This is why we decided to focus our project on adding encryption. For encryption, we decided to use the Integrated Encryption Scheme. Additionally, we want the nodes to be sure that they are communicating with the intended receiver of the message, to this end we will add a Web-Of-Trust service for the key dissemination. This allows peers to share their keys with the network without the need to trust a centralized service. Optionally we would like to add some privacy friendly messaging. In the proposed setting, we only encrypt the content of the messages, but an eavesdropper can still see who the communicating parties are. Our project would encrypt the metadata of the private messages, providing privacy friendly messaging.

Techniques to use

The techniques used for encryption will use elliptic curve cryptography, namely the Elliptic Curve Integrated Encryption Scheme. We will build upon the *crypto* package from *Go*, using the package *elliptic* to generate keys. We will use *hkdf* for the key derivation function. For symmetric cryptography, we plan on using *AES* encryption. As the integrated encryption scheme can be adapted to have MAC, we will build on it. The standard SHA256 will be used for the MAC. To implement the Web-Of-Trust, we will consider a naive approach which consists of disseminating the keys in a similar fashion to the DSDV of Homework 2. When a node receives the same public key from the same origin, it will, over time, build confidence score. The confidence score is simple : every time you see the same key from the same host, you add one point to this score, and after reaching a given threshold, the node can decide to trust the key.

Finally for the privacy friendly messaging, we would have the node encrypt the entire message including the metadata (origin and destination for instance) with the public key of the receiver. Then the message would be propagated following a *rumor mongering* protocol. Each receiving node will try to decrypt the message, and if its secret key doesn't correspond for decryption, it rumormongers the message to the other nodes, otherwise it stops the protocol and reads the received message. This protocol is less efficient than direct messaging but ensures anonymity and deniability of the sender and receiver.

Expected System Performance

Concerning the expected system performances, on startup, every peers will have some overhead in order to generate their public and secret key. From then on we will only use symmetric cryptography which has proven to be fast and efficient. In case we decide to add a simple scheme to preserve full message privacy (e.g. encrypting the whole message metadata) we will need to broadcast or monger every private message across the network, until it eventually reaches its destination. This is expected to decrease the overall application performance.

Division of the Tasks

The project work can be divided into the following major components : the implementation of the Web of Trust public key dissemination (Aleksandar) ; the local key generation and the encryption/decryption of messages in addition to the use of digital signatures (Johan); the communication between peers and the message handling (Vincent). Since our group is composed of three students, each member would be responsible for the completion of one of those three tasks.