

## A brief note on Secured P- Grid

This document talks briefly about making the current P- Grid implementation secure. We realize the following two requirements which we feel are essential for making communication on P-Grid the most secure one.

1. **Authentication:** Both peers on either side have to authenticate each other before actually starting exchanging any P-Grid related messages. This authentication is done using digital certificates.
2. **Encryption:** Once authentication is done, all the communication should be done in a secure way. Any unauthorized person should not be able to listen to the messages. This can be done by encrypting the messages before transmission and decrypting after received on the other end.

After a decent study of various security mechanisms (like Password-Based Cryptography Standard- PKCS#5 ...), we found that SSL Sockets handle both the above tasks perfectly. Moreover, it is very much integrated into Java. It is a standard technology and widely used because it has lot of built-in security related functionalities, which otherwise have to be implemented manually which might be error-prone.

All the communication in the secured P-Grid takes place on top of Java SSL Sockets. In the following, we discuss about how the SSL Sockets work and how they meet our requirements and what a participating peer in P-Grid has to do before running the client.

### The Secure Socket Layer (SSL) protocol

The primary goal of the SSL protocol (which is renamed to Transport Layer Security (TLS)) is to provide privacy and reliability between two communicating parties. SSL provides a secure alternative to the standard TCP/IP sockets protocol. The applications using SSL have to specify additional cryptographic information to the SSL layer. SSL provides

1. **Authenticity-** An SSL session involves server authentication and optional client authentication. Mutual authentication implies that information is guaranteed to be exchanged only between the intended parties. The authentication mechanism is based on public-key signatures and digital certificates. These digital certificates can be either self-signed certificates or certificates obtained from external third party certificate authorities (CAs). **For P-Grid**, we use self-signed certificates because it avoids the need for contacting the CAs and it is sufficient for the initial said requirements for secured P-Grid.
2. **Privacy-** After initial handshake, a secret key is defined which will be used to encrypt and decrypt all the messages on the connection. Since it is based on symmetric key, cryptographic operations are much faster.

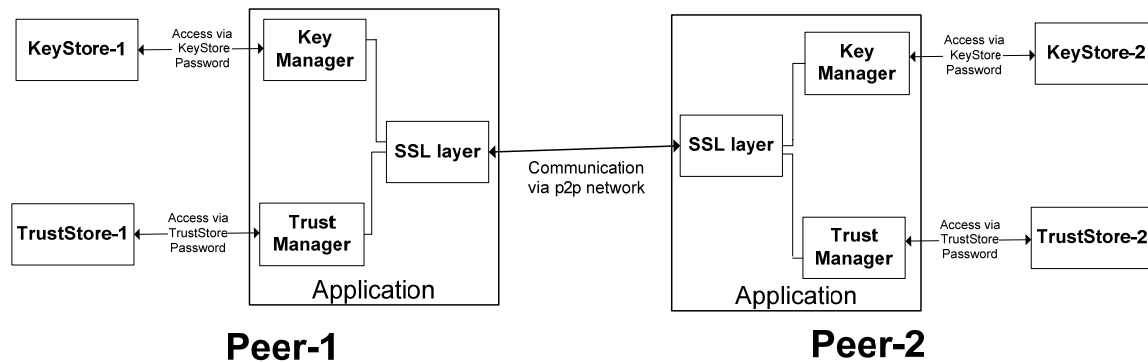
3. **Data Integrity-** The SSL connection is reliable. The message transport includes a message-integrity check based on a secure hash function. So there is virtually no possibility of data corruption without detection.

### Implementation details

A node participating in communication through SSL has to provide digital certificates and the public-private key pairs to the SSL layer which will be submitted to the remote peers during authentication phase. More over, it has to specify the certificates in which it trusts in which helps the SSL layer to authenticate the remote peers. Authentication of remote peer will be successful if and only if at least one of the certificates in the certificate chain provided by remote peer match local peer's trusted certificates.

Digital certificates and public- private key management of a user on a host is typically done through **KeyStores**. A **TrustStore** is a regular keystore to specify what kind of certificates and CAs the peer trusts in. Thus, each peer initializes its own key store and trust stores before any application using SSL is run and specifies the locations of these files to the application along with the passwords to access them. Keystores and Truststores are created by a tool called *keytool*. Each keystore is protected by a single password. Either the same or a different password can be used for private key information in the store.

The working of SSL and the concept of keystores is demonstrated in Figure. 1.



**Figure 1 System setup for SSL**

Without loss of generality, assume that peer-1 starts the communication first to peer-2 (peer-1 as the client and peer-2 as the server). The following sequence of activities takes place in the setup as part of the authentication and communication.

1. Peer-1's KeyStore (i.e., KeyStore-1) is accessed via the password and certificate is retrieved and sent to peer-2.
2. Peer-2 checks the certificate against its TrustStore (i.e., TurstStore-2). If it matches, peer-1 is authenticated by peer-2.
3. Then peer-2 is authenticated by peer-1 similarly. A certificate from Keystore-2 is sent to peer-1 and is checked against TrustStore-1 for verifying the authentication.

4. Once mutual authentication is done, a secret key is shared between the peers, which is used for encryption and decryption of the following communication messages. This is done automatically by the SSL layer.

### **Why it is secured?**

A digital certificate contains the public key information along with some identity information (like name, institution name, etc). It will also have the **digitally signed** hash of the contents (including the public key) at the end i.e., a hash function is applied on the contents and the resulting hash is encrypted using the **CA's** private key (OR by its own private key in case of self-signed certificates) to create the digital signature of the certificate. On the receiving end, this encrypted hash is decrypted using the said public-key and is checked with the hash of the received contents (the same hash algorithm is applied at the receiving end too). If the check fails, the certificate is rejected.

- **Attack-1: An outsider sniffing the traffic obtains the certificate.**  
So since private key is secret and never transmitted on network, there is no way to create such a signature knowing a certificate.
- **Attack-2: Outsider obtains both the certificate and its signature just by sniffing the traffic.**  
There is no way one illegally takes advantage, because the SSL requires the client to send a random message along with its digital signature to the server (in addition to the digitally signed certificate) as part of authentication. And generating such a signature requires the private key which is never transmitted on network.
- **Attack-3: Outsider, after obtaining the certificate and the signature, may advertise that he is already member of the secured network expecting others to join to him.**  
This does not pose a significant threat as the later part of connection setup needs sharing some random numbers secretly. Then the server has to encrypt those numbers with its private key which will be decrypted by the client with the server's public key. Since an outsider can never get the private key, this step fails. In SSL, failure of even a single step in the connection establishment phase rejects the whole connection request.

### **How it is done for P-Grid**

P-Grid security is based on self-signed certificates. So a super user or administrator of the secured network creates a KeyStore file running keytool. This file along with the password is shared to all the participants by the creator out-of-band either via email or some other mechanism. The same store acts as both key and trust stores on all peers i.e., KeyStore-1=TrustStore-1=KeyStore-2=TrustStore-2 with the same password for all. So to participate in the private P-Grid network, a peer should have above store file and the password to open that. Both should be supplied to the P-Grid client running on the peer. It is to be noted that all peers will have same public and private key pairs in this setup.