Authentication

It's not secure to transmit the password and pattern in plaintext through the network. So we need to protect the password and pattern by encryption or message digest. In our implementation, we use one handshake protocol instead of multiple handshakes for better efficiency. The format of authentication message is Key{Hash(TS | Pattern) | TS | Pattern}. The Key is 128bit generated from MD5 mesage digest of password. We generate Hash(TS | Pattern) using SHA-1, which provides intergrity protection for the message.We encrypt the Hash(TS | Pattern) | TS | Pattern by AES algorithm. Moreover, we add a salt to for better diversity. With the timestamp and salt, it's invulnerable to rainbow attack.

How to prevent replay attack? In our implementation, the server record the timestamp. If the server receives a message with the same timestamp which already exists in the time window, the server will refuse the message. Of course, the server will also drop the message with invalid timestamp.

However, there is some drawback in the protocol. Since the client couldn't authenticate the identity of the server, it's vulnerable to in-the-middle attack. Besides, the protocol is not secure when it comes to server break-in attack. The attacker could also modify the time in the server.