How to Implement a new Protocol

For convenience we have included to empty classes and methods in the simulator code for you to implement a new round based protocol.

We call this protocol `SampleProtocol` any class and methods starting with `SampleProtocol` is the empty classes and methods needed to implement for a new protocol and we will summarize all the requirements below:

Implement the major classes extending from generic class

More detailed on what those classes do and their parent class methods please refer to `Javadoc`.

```
// Configuration class that stores the protocol configurations from config.json
class SampleProtocolConfig;

// IO methods converting configurations, message_trace, and state_trace from JSON
to Object or from Object to JSON
class SampleProtocolJsonifier;

// Message class that holds info to be passed during protocol execution.
// This should compatible for both input and messages in rounds
class SampleProtocolMessage;

// Player object that holds states of a single player
// to be accessed by NetworkSimulator to receive message and for PlayerController
to query state
class SampleProtocolPlayer;

// Major class that implements methods to generate messsages during protocol
execution.
// This is the major class to implement honest player execution and adversary
attack
class SampleProtocolPlayerController;

// Protocol execution procedures
class SampleProtocolRoundSimulator;
```

Implement a FAuth() for new protocol

Inside `class CryptographicAuthenticator` there is an empty method to implement for this protocol. New protocol can call this method to append a signature for the message.

```
/**
 * TODO: Implement the authentication function to generate a unique signature and
append that into message's signature
 *
```

```
     * @param message
     */
    public void sampleProtocolFAuth(final SampleProtocolMessage message) {
    }
```

Register the protocol in App class

In the `App` class you need to add the new protocol to be recgnized for initializing the correct child protocol class. We already implemented this for `SampleProtocol` as below. Except for major change in simulator, you do not need to change this:

```
else if (protocol.equals("sample_protocol")) {
    SampleProtocolRoundSimulator simulator = new
SampleProtocolRoundSimulator(args[0]);
    simulator.run();
}
```

Any additional helper methods

After you implement any helper message to get this protocol up and running, you are ready to go!