Antigone is an enforcement tool for dynamic group policies. These group polices are an agreement between participate of the Antigone system. The group polices enforce how security drives participant behavior, which security mechanism will be used and what entities will be allowed into to participate. The Antigone system uses three entities to adjust a shared group policy; these entities are a policy issuer, an initiator and a participant. I should note that a participant can be an initiator or a policy issuer. The policy issuer states session requirements for the group and the requirements for future sessions. The initiator generates the policy that the issuers distribute. The initiator can create either a group or local policy. Each participant defines a set of acceptable security parameters in their local policy. Using the local policies from the each participant and group policies created by the initiator, each entity in the group collaborates with each other to achieve a set of security parameters and behaviors that are agreed upon by all participants.

The developing group of Antigone has implemented a basic API (30,000 lines of C++). This API can accept new code additions for features like auditing and fail detection. The Antigone system itself consists of 4 layers; a group interface layer and mechanism layer a policy engine and a broadcast layer. The group layer interfaces an application with components in the mechanism layer and the policy engine. The mechanism layer is at the heart of the Antigone system since this layer is the software that is used to implement secure groups. The broadcast layer is need since multicast is not globally available. Therefore, Antigone uses its own broadcast layer so that it can be deployed in various network environments.

I was interested in reading this paper because of work I have done trying to secure a peer-to-peer network. I am also working on a project for University that polls windows workstation to see if they have any new data for a calling database (called Mosquito). I have been having trouble with both creating secure groups efficiently. This helped me to come up with some better ideas for both projects. While I will not be implementing something on the scale of this work, I hope to use some of the idea presented in the paper. The API and policy complier seem very easy to use and very easy to understand syntactically.