Rough Idea/Abstract: There are many cryptographic-based messenger applications that exist today. Goals of earlier protocols such as OTR were to provide encrypted communications with perfect forward secrecy, plausible deniability, and real-time, synchronous communications. Newer protocols such as the Signal Protocol (based on the Double Ratchet Algorithm), which was recently implemented in the popular messaging application *WhatsApp*, aim to provide the benefits of OTR along with the ability to push group messaging, multimedia, and asynchronous communication for devices such as in a mobile environment, where users can go offline or disconnect from a stream. Furthermore, protocols like the Signal Protocol aim to use strong handshakes (Triple DH), and modern Elliptic Curve Cryptography based on the fast, patent-free and very secure Curve25519. The use of a Double Ratchet Algorithm provides the benefit of key renewal without any live communication with a user, as well as providing additional steps to maintain message ordering. The Signal Protocol is very strong: the protocol provides confidentiality, integrity, authentication, participant consistency, destination validation, forward secrecy, backward secrecy (aka future secrecy), causality preservation, message unlinkability, message repudiation, participation repudiation, and asynchronicity[1]. However, the implementation requires servers for relaying messages and for the storage of public key information of all users.

The goal is to develop a decentralized application with the strong security properties of the OpenWhisper/Signal Protocol while maintaining a ***decentralized*** and ***inherently anonymous*** platform. Among other things, decentralization and inherent anonymity provide the user with complete privacy. Coupled with a strong messaging protocol (such as Signal), we can achieve a platform which provides a minimal to zero trust in "public" infrastructure. This is akin to a decentralized encryption system (such as PGP, which does not rely on any central authority) vs. a PKI system where a "trusted" agency provides rendezvous and storage services – such as in all applications currently utilizing the Signal Protocol (Signal, WhatsApp, etc.).

An example of a transport protocol that is decentralized and inherently anonymous is the Tor Protocol, which has widespread use. Tor provides a sound transport layer; that is we are able to securely transmit data through the Tor Network between two nodes (hidden services) without any knowledge of the identity of either node, and utilizing onion routing to establish a secure rendezvous point for nodes to meet. The transport protocol of Tor is fairly strong; utilizing proper cryptographic protocols for the secure transmission of data. Properly implemented, Tor also provides inherent protection against metadata leaks, does not utilize public name systems like DNS (and hence providing additional security), as well as decentralization with no single entity responsible for the transport of data.

An example of a messaging application that utilizes Tor is *Ricochet*, which utilizes the Tor Protocol to create a local hidden service and connect to the Tor Network. A user's contact name is derived from the Tor specification, as a hashed value of a public key corresponding to a private key for a hidden service. An example of this would be: rs7ce36jsj24ogfw. As with Tor addresses, contacts are identified by their unique identifier, and the Ricochet application provides an additional RSA signature handshake with HMAC-SHA256 to provide authentication and proof that a contact is both a) the owner of the private key, or hidden service, and b) able to provide a unique, verifiable hash value (done by the HMAC-SHA256) for each individual contact request. In this way, Ricochet provides authentication and integrity for contact requests: we can be sure that a user is the owner of their private key and that the initial contact request is made uniquely with a verifiable signature. With this, we can now communicate end-to-end through the Tor Network, anonymously sending messages through the network directly to a contact that is authorized. Messages are streamed directly from one party (or hidden service) to the

other, via. the standard Tor relays (a 3-onion routed relay to rendezvous point for each client), thereby accomplishing anonymous messaging through the already-existent Tor Network.
While Ricochet provides a lot of benefits (namely the benefits of the Tor Network: inherent anonymity, protection against metadata leaks, no PKI system/decentralization, and no utilization of public name systems like DNS), it is not a perfect solution to messaging. Messaging (such as instant messages, multimedia in a chat setting, etc.) The Tor Protocol is a strong transport protocol, but for *true* perfect forward secrecy and stronger plausible deniability we must implement further cryptographic protocols. With Tor, we inherit the forward secrecy of the (EC)DH handshake during the transport. Building upon Ricochet, the goal is to blend the very cryptographically secure properties of the Double Ratchet Algorithm / a modified version of the Signal protocol with peer-to-peer, decentralized communications through local hidden services through the Tor Network (as Ricochet does). This will help separate the *identity element of the user* from the *transport (hidden service)*, which will accomplish perfect forward secrecy and build stronger deniability, among other privacy benefits. A resultant messaging application utilizing the hybrid encryption scheme for ***one-to-one, synchronous messaging*** is not difficult to implement.

The challenges arise when we want to provide *asynchronous* capability for communications (which comes in hand on mobile applications, or when we wish to maintain cryptographically secure key derivation and management for offline clients), as well as group messaging without the usage of a somewhat PKI derived structure (such as WhatsApp) to relay messages groupwise/multicast and maintain public-keys centrally.

Performing these tasks through the Tor Network pose unique challenges, and the goal of this paper is to explore methods to accomplish stronger characteristics derived from the Signal/Double Ratchet Protocol (confidentiality, integrity, authentication, participant consistency, destination validation, forward secrecy, backward secrecy (aka future secrecy), causality preservation, message unlinkability, message repudiation, participation repudiation, and asynchronicity) as well as a basic algorithm for group messaging using the multicast scheme (user sends a *single* message once and it is distributed to all N participants in a group conversation N times vs. sending message N times to N group members directly). The challenge lies in *distribution*, as there is no central server with Tor that would normally distribute such messages and associated keys, KDF Ratchets, etc. We also need a multicast scheme as described for strong deniability, a significantly more advantageous forward secrecy guarantee, and importantly to ensure ordering of messages (again somewhat relating to the asynchronicity problem, but also with a decentralized system with inherent anonymity we need to ensure all group members receive messages in order).

The remainder of this paper will be split into three main sections: firstly to discuss an implementation of the Double Ratchet Algorithm over the Tor Network utilizing Ricochet-style hidden services as base contact points for synchronous one-on-one conversations, secondly to discuss a method for asynchronicity over the network followed lastly by a method of establishing group messaging communication through the resultant protocol.