So there is the potential in this space for DKG to become quite useful, and so a number of people in the space have been looking into creating implementations of this thing. Maybe it's a good time to try and standardize this?

I know of Parity's Secret Store, based off of this ECDKG paper. In addition to the base DKG stuff, they also include the encryption shadows use case as a first class citizen in their system having to do with "document keys" if I interpret their docs correctly. Their system uses the secp256k1 elliptic curve, and they have two ways of specifying the DKG nodes:

1. A listing of the nodes' public addresses and IP addresses placed in a node's configuration.
2. Contracts which exist on the (Kovan-only, as of the time of this post) chain, which implement a specific interface. I am unclear on how nodes find and private message each other in this scenario though, as I am looking at the contract interface and don't see any dialing info.

Also, each DKG node in Secret Store specifies its own "ID", which is necessary for generating secret shares for key recovery, in that the ID is the parameter for the secret polynomials of a node. This means that node IDs have to be unique to every instance of DKG / within a set of nodes generating a keypair.

I've prototyped a DKG implementation based on the same paper as before in Python. It's not complete, and it's been abandoned, but there were a couple of major differences from Secret Store:

1. The only configuration supported is manually, similar to Secret Store.
2. The IDs are assigned the value of a node's Ethereum address.

Dfinity also has a DKG implementation in the works. Theirs is being written in JS.

There are implementations of DKG in a golang crypto library called kyber. It is backed by DEDIS, which is a research lab at a uni. This is not to be confused with Kyber Network, which seems to be a completely separate entity.

They have two different implementations of DKG which they have marked as Pederson and Rabin, probably after the authors of the papers on which the implementations were based.

There's also a DKG implementation in the works at Gnosis (full disclosure: I'm at Gnosis). It is being written in Golang, and may be seeking to target the pairing curve that has opcodes on EVM (for SNARKs).

Honey Badger BFT has an implementation of DKG based on a paper Distributed Key Generation in the Wild. Unlike the other implementations in this list, theirs uses bivariate polynomials.