Graph Commitment to Bit Vectors

1 Intuition for Committing to Bitstrings using Graph States

The basic idea is as in Minh's original scheme. That is, to commit to a bitstring m, we encode the message as some graph G and generate the graph state $\rho(G)$ as our commitment. A verifier will perform corresponding stabilizer measurements based on the message which the committer reveals.

The problem with this is that $\rho(G)$ does not hide the graph G, and therefore does not hide the corresponding message m. To remedy this, the idea is to embed G in a larger graph G' in some kind of randomized manner and have $\rho(G')$ as the commitment. The reveal/verification stage involves sending the sequence of measurements detailing how to get back to the $\rho(G)$.

The problem with this now is that the scheme will not be binding. Alice can reveal a different set of measurements which will get Bob to some other subgraph G'' rather than G'. This allows Alice to open to multiple messages, breaking binding.

2 Possible Fixes

Ideally, we would like a commitment mechanism to work as follows:

Committing: To commit to a message string m, some large graph G' should be generated in a randomized way (this will give us the hiding property). We output the graph state $\rho(G')$ as the commitment.

Reveal/Verify: The large graph G' should have some small fixed graph G such that $\rho(G)$ can be generated after applying some Z and Y measurements. These measurements should depend on the message somehow. Only this sequence of measurements should be able to get us back to the $\rho(G)$ graph state. This will ensure binding.

Attempt: Say we want to commit to a single bit message 0 or 1. Lets say the fixed public graph (state) G which the verifier is supposed to end up with after measuring is the graph with nodes labelled 1 and 2 and an edge connecting them. Our commitment to the message bit 1 could be the graph in figure 1:

After the committer reveals the message, the verifier could measure some fixed vertices in Y or Z (based on what the message is) and check that the measurements result in $\rho(G)$. In the case of $m = 1$, the verifier could do a Y measurements on vertex 3 in $\rho(G')$. This results in the graph state for the graph in figure 2 . In the case of $m = 1$, the verifier could measure vertex 3 and get a graph state for a complete graph with 2 vertices as desired. There is no way to obtain the graph state for G using any other (single) measurement. However, I think the commitment is not hiding since this is the only G' is the only graph which works as a commitment when our message bit is 1. So there is no randomization.

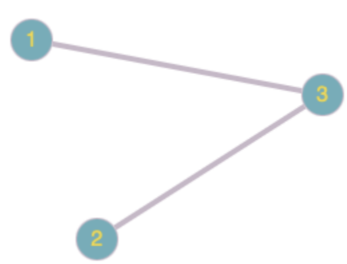

Figure 1: Caption

Figure 2: Caption