***Question 1***

|  | A | B | C |
| --- | --- | --- | --- |
| Y | 0.5 | 0.5 | 0 |
| Z | 0.25 | 0.25 | 0.5 |
| W | 0.25 | 0.25 | 0.5 |

1. If W is revealed to be controlled by Bob, then it reveals that Z is Carol and Y is Alice.
2. Following the diagram below, the first mix will be of equal distribution, allowing U and X to both be 0.5A and 0.5 B.

Taking X, we mix it with C, to give V and W. V would be a distribution of ⅔ C and ⅓ X, while W will be ⅓ C and ⅔ X. Expanding X in W, we ⅓C and ⅔ (½ A and ½ B), leading to ⅓ C, ⅓ A, and ⅓ B, thus giving us equal distribution for all 3 parties in W.

Lastly, we mix U and V together with equal distribution. This would give Y and Z, where they would both have ½ U and ½ V. Expanding it, we would get ½ (½ A and ½ B) and ½ (⅔ C and ⅓ X), which gives us ¼ A and ¼ B from U mixed with ⅓ C, A and B from V. Adding them all together, we get ⅓ C, ⅓ A, and ⅓ B, thus giving us equal distribution for all 3 parties in both Y and Z as well.



***Question 2***

1. where B is the Bernoulli distribution.
2. where B is the Bernoulli distribution.
3. where B is the Bernoulli distribution.
4. If no block is proposed then it may take longer to validate and finalise a transaction, thus reducing throughput. If multiple blocks are proposed then there will be an issue of how to choose a leader/winner amongst those blocks. However, with multiple blocks, designers can use a tie breaker, such as choosing the coin with the largest hashed VRF output. Hence, multiple winners may be slightly more favourable.

***Question 3***

1. Assuming 70 transactions in a block with 1024 bit public keys and 256 bits signatures, we would be saving on 69 x 1024 x 256 = 18087936 bits worth of bandwidth. Savings could vary if we store the hash of the public keys instead of the full key.
2. Assuming 70 transactions in a block with 256 bits signatures (since Ethereum doesn’t have public keys), we would be saving on 69 x 256 = 17664 bits worth of bandwidth.
3. This could allow for faster verification of blocks in BitcoinNG, thus allowing for higher throughput.