Frost	CLSAG	Use of Symbol
$\mathbb{G}$	$\mathtt{G}$	Denotes a group
$\mathtt{p_i}$	$\mathtt{pk_i}$	Denotes some public key
$\mathtt{s}$	$\mathtt{sk}$	Used to denote secret key
$\mathtt{s_i}$	$\mathtt{sk}$	Denotes the i^{th} secret share, assigned to the P_i^{th} participant
$\mathtt{t}$	-	Number of Threshold Participants
$\mathtt{n}$	-	Number of Total Participants
$\mathtt{P_i}$	-	Denotes the i^{th} participant
$\overrightarrow{C}$	-	Public Commitment of the form $\langle \phi_0, \phi_1, ..., \phi_{t-1} \rangle$
$\mathtt{SA}$	$licshares\mathtt{Sign}$	Signature Aggregator
$\mathtt{Y_i}$	-	Public Key Shares assigned to P_i
-	CO	Denotes the corruption oracle.
ρ	ρ_i	Queries to an oracle

Table 1: Notation employed for cryptographic components.

It should be noted that, when reading both the Frost and clsag papers, if the variables or notation denoted below is not found in the table, then it likely stands for a generic value which is defined immediately within the same sentence. These will not be included in the Stone below.

1. It should be noted that oracle queries of all types may or may not possess a number of parameters specified in the same line. Their 'type' is also specified, albeit in a sort of non-obvious way. Regardless of the inconsistency, ρ can be safely interpreted as a vector of some length, with ρ_i or $[\rho]_i$ denoting the i^{th} entry, or the i^{th} query to the oracle. Both are equivalent under the vector interpretation.

2. It should be noted that random oracles and Hash functions seem to have extremely similar notation, so caution should be warranted. In theoretical proofs, it seems to matter little, but universally, H does not necessarily mean a hash function.

3. The notation is fairly distinct between the two papers, with the exceptions noted above. The main difference seems to be the bulk of the notation used in the proofs, particularly in paramaterization of quantities with numerous parameters. (If it is helpful, I can parse through and find the notation with numerous parameters, and detail the nature of the parameters throughout the two papers. This would be more or less just a little cheat sheet that may help with reading.)

1 Notes for the Frost Paper

1. In Frost KeyGen, the notation is very consistent. The only clarifying factor would be that the subscripts that are indexed as a_{ij} should have some separating symbol, like $a_{i,j}$.
2. In round 2, modulus operator in the exponent should be contained in round braces. The comment from above also holds for the product indexing.
3. In the signing protocol on page 13, it looks like set theory notation mixed in with casual notation. Recommend replacing the ':' with the verbal equivalent of 'such that' or encasing in brackets. Other instances occur, so it is recommended to fix those as well.
4. On page 15, the summation in 7.c is not indexed. Shoddy notation.
5. I think that the proof for Theorem 6.1 can be streamlined a bit.
6. Lemma A.1, needs left(, right). The entire paper is beautifully written. It would benefit from some elaboration throughout and an adjustment in formatting to increase readability. Beside the comments above, it is a shining example of a well written paper. It might be worth leaving proof sketches out of the upper section and just moving the whole proof up.

2 Notes on the original clsag Paper

1. I would recommend a slight change in the notation in the preliminaries. Just cleaning up the notation.
2. 3.1, citation needed. Not sure where to find this result. I believe it, but I would like a paper.
3. The linkable ring signature definition is a beast and a half.
4. Typo on page 8, 'we' shows up twice in the first paragraph.

3 Linkability and Forgeability Definitions

3.1 Linkability

1. A careful thing to note, appearing in the clsag definition of *pigeonhole linkability*, is that the adversary does not require key generation, corruption or signature oracle access. It is the case that the adversary

behaves exactly the same with or without them? Where does this independence come from?

2. ACST linkability allows certain oracle queries, under some constraints. My immediate question is: does the adversary from pigeonhole linkability possess the ability to retrieve the same information as the in ACST without an oracle? Presummably, if the adversary from the pigeonhole linkability does not need them, but does this imply that he has them and chooses not to use them? Would it not make for a more powerful attacker potentially, giving oracle access regardless? I think clarification should be made regardless, as the ambiguity leads to two attackers, distinct in their capability, but equal in their attack power.

3. There is a claim in Definition 6 that ACST linkability and Pigeon hole linkability are not the same, due to differing success conditions. However, is the overlap frequent enough that they only differ in a small number of cases? (Cases that will never see practical use?)

4. A common theme I see in the papers I've read is the use of common venacular in one subfield placed into another unfamiliar field. This can lead to confusion during an initial read through and needlessly obfuscates the process. For example, in point (30 on page 17, it is stated that M picks two random tapes, **h** and **h'** to simulate an oracle response. But are these not just vectors of some length with entries in a familiar field? The word tape could be simply elaborated on. A change as simple as, 'let h and h' be vectors of length n, in field f, which are used as tapes to simulate oracle responses.' This sort of proof modification would benefit, I think, readers from all familiar backgrounds.