We thank the reviewers for their valuable comments and address the main concerns raised in review order.

R2 What is the break-down of the runtime, mostly communication or computation? More thorough discussion of runtime of ReLUs: The runtime of CryptoNAS is dominated by the cost of securely evaluating ReLUs, for which we used the ABY library [3]. Unfortunately, ABY does not break down total runtime by communication and computation and hence we only reported the end-to-end runtime. We will instrument ABY to measure the two separately and report breakdowns in the appendix. It is worth noting that reducing ReLU counts translates directly in reductions in *both* communication and computation costs, since the bottleneck for both are ReLU evaluations. A discussion of the relative costs will be added to the paper.

R3 Better reference for garbled circuits is Bellare et al. at CCS '12: We thank our reviewer; citation will be fixed.

R3, R4 In Algorithm 1 how is β defined? This was an error on our part. In line 198 β was missed, it should be $\alpha^2 = \beta = 4$, where β is the channel scaling across layers (conventionally set to 2).

R4 The use of ENAS may be superfluous. Larger models, properly regularized, are often more accurate:

We thank the reviewer for this suggestion. We ran the suggested control experiments and the results are shown in Table 1. We observe that the largest network (5×5 filters and all skips) for CNet3 ReLU balanced had an accuracy loss of 0.24% (CiFAR-10) and 0.48% (CiFAR-100) compared to the models found by ENAS. That is, the models returned by ENAS are more accurate than the largest all-skip models. This also reaffirms the results reported in [26], where models with all possible skip connections showed an accuracy drop.

Table 1: Comparing models discovered by ENAS to largest models in search space (all-skip, 5×5 filters).

Model (Base-Dataset)	ENAS Arch		Largest Arch	
	Params	Acc	Params	Acc
CNet3-C10	166M	95.55%	311M	95.31%
CNet3-C100	149M	79.59%	311M	79.11%

We note that the models (including largest all-skip models reported in Table 1) are already regularized using dropout and L2 regularization. While it is possible that there is an even better regularizer for the largest model, this would be a significant research problem in itself. In this context, ENAS search can itself be viewed as akin to a regularizer on the largest model since it drops a subset of its skip connections to increase accuracy.

R4 Baselines accuracy and ReLU counts of models without shuffling and pruning (Section 4.2 and Table 1): The ReLU counts for the baseline models (385K, 1.92M, and 3.89M) were reported in the text on Page 6, line 241 but we will add these to Table 1 to highlight them. The accuracy of the baseline is the same as the model with shuffling.

R4 The authors currently don't provide the experiments to empirically justify the added complexity of using ENAS: We thank the reviewer for this point, and will add the control experiments to our paper (as noted in Table 1).

R4 Is the search space identical to the one in the original ENAS paper? Releasing searched architectures and code would also improve reproducibility: Yes, the search space is the same as ENAS' macro-search space. We clarify this in the paper and publicly release our architectures and code.

R5 This work is not totally orthogonal to [15], it would have been nice to also incorporate those ideas: We agree with the reviewer that CryptoNAS and Delphi's [15] optimizations are not entirely orthogonal. Our intent was to note that Delphi can be applied on top of the models found by CryptoNAS for further benefit. But, as the reviewer notes, the two can also be combined in more sophisticated ways which would be an interesting direction for future research. We will update the paper with this note.

R5 The experiments are done on Cifar10/100 which are very similar datasets. It would have been nice to see the effects of this optimization on other application domains: Most prior work on private inference experiments on easier datasets such as MNIST and Cifar10. We included the more challenging Cifar100 dataset in our experiments (Delphi is the only other work to do this), but we concur that experiments on a more diverse range of application domains will be very valuable. We will seek to do this in future work and add a note to this effect in the paper.

R5 Why choose MiniONN for the experiments, as both Gazelle and Delphi outperform MiniONN significantly: The cost of privately computing ReLUs is the shared bottleneck in all three frameworks, and therefore CryptoNAS' benefits will transfer equally to Delphi and Gazelle. We could not use the Delphi protocol since the paper was published only recently and concurrently with our research. Therefore, updating CryptoNAS' implementation from scratch with the Delphi crypto protocol would not have been possible in the available time (although, as shown in Fig. 3, we were able to estimate the runtime of Delphi's architectural optimization using the MiniONN crypto protocol). However, since Delphi crypto protocol improves upon MiniONN's, the runtimes of the CryptoNAS models using Delphi's protocol should be even smaller.

R5 In the impact section, you could also discuss the dark side: We thank the reviewer for this point and will discuss it in the impact section.