What devices for the works?

FHE requires several non-SIMD operations, such as number-theoretic transforms, that are inefficient on CPUs and GPUs. But these operations can be accelerated by specialized functional units, avoiding these inefficiencies. As a result, FPGA and ASIC-based accelerators are selected for this work. Prior accelerators are efficient only on a limited × on FHE programs compares the performance of CraterLake, CPU, and F1+ on deep and shallow benchmarks. It shows execution times and CraterLake's speedups over the CPU and F1+.

GPUs are inefficient on FHE: Prior work has studied the use of GPUs to accelerate FHE. While the data-parallel nature of GPUs may seem a good fit for FHE, these efforts achieve modest speedups over multicore CPUs. This is because GPUs lack modular arithmetic, cannot implement all-to-all operations like number-theoretic transforms and automorphisms efficiently, and their on-chip memories are too small to enable sufficient reuse, despite their use of HBM. Specifically, state-of-the-art GPU approaches carefully tune algorithms to achieve high off- and on-chip bandwidth utilization; however, this prior work is 200× slower than CraterLake. This shows that CraterLake's high reuse is crucial: to achieve the same throughput as CraterLake, a GPU would need over 100 TB/s of memory bandwidth.

The main problem on the designs which are FPGA-based, is that they are small and miss the data movement issues facing an FHE ASIC accelerator. These designs also overspecialize their functional units to specific parameters, and cannot efficiently handle the range of parameters needed within a program or across program.