

Towards Verifiable FHE in Practice

Proving Correct Execution of TFHE's Bootstrapping using plonky2

Louis Tremblay Thibault and Michael Walter
Zama

Abstract

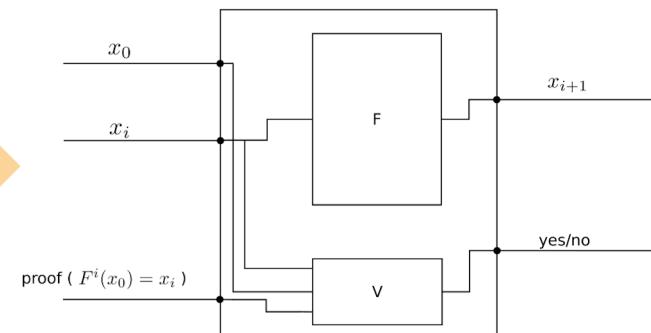
We explore the integration of FHE and SNARKs to prove the correctness of FHE operations. Despite the challenges of overhead in both FHE and SNARKs, we demonstrate progress towards a practically verifiable bootstrapping operation. Our findings suggest practical feasibility, offering promising implications for secure and efficient computation outsourcing.

Why Plonky2 ?

- Small base field
- Transparent setup
- Efficient recursion (verifier circuit is small)

Recursion-based IVC

```
var x = input
for i = 0 to n {
    x = F(x)
}
return x
```



How to prove a loop using recursion. The prover repeatedly proves the circuit for an iteration. "V" is the verifier circuit.

Parameters

n	Q	N	k	B	I
728	Goldilocks prime	1024	1	32	4

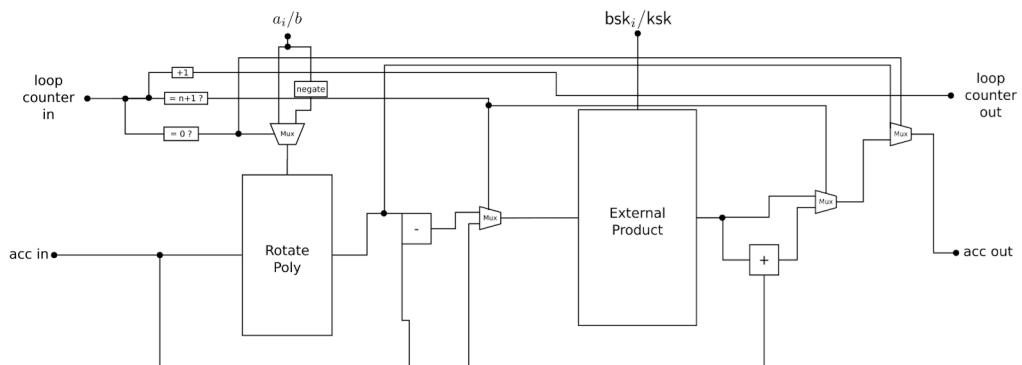
Highlights

- Recursion-based IVC
- About 20 minute proving time, <200kb proof size, <10 ms verification time
- 100 - 1000 times faster than zkVM-based systems

Motivation

- Aim to replace trust in hardware vendors with cryptographic assumptions
- Bootstrapping is the main building block of TFHE

IVC Circuit for PBS



An illustration of the circuit for a loop iteration of TFHE's PBS (denoted "F" on the left diagram). The dominating subcircuits are polynomial rotation and the external product.

Performances

	Prover time	Verifier time
M2 MBP (8 cores, 24GB memory)	~48 min	<5 ms
C6i.8xlarge (32 cores, 64GB memory)	~39 min	<10 ms
C6i.metal (128 cores, 256GB memory)	~21 min	<10 ms