Speedup via batching

- A ciphertext encrypts an array of values
  - E.g., each is a bit or a small integer
- Array size determined by other parameters
  - E.g., 378, 600, 682, 720, 1285, …
- Homomorphic operations include:
  - Element-wise addition/subtraction, multiplication
  - Addition/subtraction, multiplication by constants
  - Cyclic/non-cyclic shifts
  - Also SELECT(A_1,A_2, pattern)
    $= \text{pattern}\times A_1 + (1\text{-pattern})\times A_2$

Performance

- In Jan-2012 we had an implementation that evaluated the AES-128 circuit in 36 hours
  - Note: AES does NOT support homomorphism, we just used the circuit that computes AES as an example
- With parallelism, we can encrypt ~20 blocks in one operation
  - vs. 20x200 cycles (approx. 2ms) for doing the same thing in the clear (in software)
  - "Only" 10 orders of magnitude slower

Recent Performance (Dec 2012)

- Security parameter=80, circuit width=4 arrays

	Circuit “depth”	Array size	Time (hrs:min:sec)
	7	**224**	**0:00:38**
	14	**480**	**0:02:49**
	35	**512**	**0:19:05**
(∗)	**70**	**720**	**3:01:51**
	84	**2048**	**5:24:47**

(∗) maybe similar work to homomorphic AES

– If true, ~12x speedup on our previous implementation