This module implements the base sVole protocol, Single-point sVole, and the sVole extension protocol presented in Figure 5, Figure 7, and Figure 8 of the write-up <https://eprint.iacr.org/2020/925> respectively. Although I have added the optimizations suggested in the paper, the current state of the implementation, in particular, the extend the functionality of the sVole extension protocol doesn’t meet the required performance presented in Table 3. However, the setup phase of the protocol does.

**Probable next steps:**

When I profile the VOLE implementation using Xcode instruments, I realized that there is a bottleneck in the following function:

* Lpn\_mtx\_indices
  + rng: &mut AesRng
  + rand\_core::Block\_rng::generate and set
* AesRng::from\_seed(seed)

**Performance using an example:**

Field: F61p

[642048(k+t+r+52287)] Send time (init): 573 ms

Sender init communication (read): 7.50 Mb

Sender init communication (write): 145.42 Mb

[642048(k+t+r+52287)] Receive time (init): 573 ms

Receiver init communication (read): 145.42 Mb

Receiver init communication (write): 7.50 Mb

[10214168(n-n0(k+t+r))] Receiver time (extend): 4384 ms

[10214168(n-n0(k+t+r))] Send time (extend): 8201 ms

Sender extend communication (read): 4.81 Mb

Sender extend communication (write): 31.83 Mb

Receiver extend communication (read): 31.83 Mb

Receiver extend communication (write): 4.81 Mb

Total time: 8775 ms

**Current Benchmarks:**

lpn\_svole::init::F61p time: [472.81 ms 477.10 ms 481.40 ms]

Found 1 outliers among 100 measurements (1.00%)

1 (1.00%) high mild

