| **Name** | **Parties** | **Description** |
| --- | --- | --- |
| [Obliv-Java](https://github.com/Calctopia-OpenSource/jdk10u) | 2PC | Obliv-Java combines MPC and proof-carrying code to solve DAO and Gyges attacks. Reimplementation of Oblic-C. |
| [ABY](https://github.com/encryptogroup/ABY) | 2PC, Semi-honest | ABY supports arithmetic, boolean, and Yao garbled circuit sharing, and also secret sharing. ABY can also convert between the different types. |
| [ABY3](https://github.com/ladnir/aby3) | 3PC, Semi-honest | ABY3 uses mixed protocol secret sharing for secure machine learning and database unions. Contains regression algorithms, database functions, and set comparison. 50x improvement in throughput and 24x improvement in latency over optimized boolean circuit. Fixed point technique is exponentially faster for some machine learning operations. |
| [BatchDualEx](https://github.com/osu-crypto/batchDualEx) | 2PC | BatchDualEx is attempted from Kolesnikov et al. (2015). The strategy is to repeatedly perform the same function to compute as much as possible offline before inputs can become known. Cost per evaluation is 5.1ms offline + 1.3mx online. Further steps are tuning it to trade performance for leakage. We can make significant performance strides if we allow adversaries to learn a single bit of input with probability 2e-20. |
| [EMP-toolkit](https://github.com/emp-toolkit) | 2PC, Semi-honest | EMP-toolkit works with garbled circuits and has variants secure against malicious parties. 700x time complexity improvement over prior work, and works with 128 parties across 5 continents in under 3 minutes. Largest scale demonstration of secure computation to date. |
| [Fancy-Garbling](https://github.com/spaceships/fancy-garbling) | 2PC, Semi-honest | Fancy-Garbling presents new techniques of weighted threshold gates and exponentiation for arithmetic circuits and also threshold gates for boolean circuits. |
| [Moose](https://github.com/tf-encrypted/moose) | 3PC, Semi-honest | Moose uses secret sharing and focuses on encrypted data processing and machine learning, written in Python |
| [Obliv-C](https://github.com/samee/obliv-c) | 2PC, Semi-honest | Obliv-C uses garbled circuits to enable embed secure MPC based in C. Oblic C is a new language that allows experts to code secure computations without cryptography knowledge, enabling ORAM and width-limited int abstractions. |
| [Rosetta](https://github.com/LatticeX-Foundation/Rosetta/) | 3PC, Semi-honest | Rosetta uses secret sharing and reuses Tensorflow API and tradition. Rosetta integrates MPC, cryptography, and federated learning. Roetta provides secure AI solutions, decoupling Tensorflow AI and privacy technology to enable compartmentalization. |
| [Sharemind](https://github.com/sharemind-sdk) | 2PC/3PC, Semi-honest | Sharemind uses secret sharing to enable end-to-end data protection in data-driven services, classifying user information to the company and to Sharemind itself. |
| [TF Encrypted](https://github.com/mortendahl/tf-encrypted/) | 3PC, Semi-honest | TF Encrypted focuses on Tensorflow applications and uses secret sharing. |
| [TNO-MPC](https://github.com/TNO-MPC/) | - | TNO-MPC contains a large number of generic software tools in 20 repositories that can be used for MPC systems. Uses Keras API with the goal of making secure machine learning available without user expertise in privacy or distributed technology. |
| [Carbyne Stack](https://github.com/carbynestack/carbynestack) | - | Carbyne Stack is a cloud stack for client-side MPC models. |
| [CrypTen](https://github.com/facebookresearch/CrypTen) | Semi-honest | CrypTen is built on PyTorch and enables secure machine learning based on arithmetic and binary secret sharing. |
| [FRESCO](https://github.com/aicis/fresco) | Semi-honest | FRESCO is written in Java and provides a library of many secure functionalities. FRESCO has been demonstrated in secure data science, financial company performance, and credit ratings. Supports TinyTables & SPDZ protocols. |
| [HoneyBadgerMPC](https://github.com/initc3/HoneyBadgerMPC) | T < n/3 malicious parties | HoneyBadgerMPC is uniquely suited for blockchain integration and provides guaranteed output delivery despite Byzantine faults. First, collect client input and submit that input to servers via COPS15. Secret share with random mask—client retrieves shares of this mask and reconstructs the mask. Then, servers obtain their share of the input. Then, the protocol computes the MPC program layer by layer via linear operations by default. Share reconstruction is aggressively batched. Using RanDouSha, random values are created at the same rate before consumption. |
| [JIFF](https://github.com/multiparty/jiff/) | Semi-honest, dishonest majority of non-server parties | JIFF is a JS library for building applications that doesn’t require users to understand MPC. JIFF uses Beaver’s method with triples generation based on bgw. |
| [MOTION](https://github.com/encryptogroup/MOTION) | Semi-honest | MOTION incorporates performance optimizations with better communication complexity and latency over Oblivious Transfer. *N* parties and up to *N-1* passive corruptions, performs hundreds of times faster than SCALE-MAMBA and tens of times faster than MP-SPDZ. |
| [MP-SPDZ](https://github.com/data61/MP-SPDZ) | Semi-honest or malicious, dishonest or honest majority | MP SPDZ benchmarks MPC protocols in a variety of security models that are provided by the package. MP SPDZ only offers garbled circuits with binary computation. |
| [MPyC](https://github.com/lschoe/mpyc) | Semi-honest, honest majority, t <= (m-1)/2 | MPyC is built on Python and features secure types, asynchronous evaluation of underlying protocols, and transparent communication between parties. Succeeds VIFF. Next steps include secure numpy arrays and transparent secret indexing. |
| [SCALE-MAMBA](https://github.com/KULeuven-COSIC/SCALE-MAMBA) | Malicious, dishonest majority | SCALE-MAMBA is a general MPC platform that is built on secret sharing. |
| [swanky](https://github.com/GaloisInc/swanky) | - | Swanky is a suite of rust libraries for MPC, including oblivious transfer, garbled circuits, and private set intersection. |
| [EzPC](https://github.com/mpc-msri/EzPC) | Semi-honest | EzPC is an end-to-end MPC-based library for Tensorflow-based secure machine learning, offering protocols, frameworks, compilers, and inferences. |
| [libTMCG](https://github.com/HeikoStamer/libtmcg) | Semi-honest | libTMCG is built on C++ and offers tools and protocols for MPC-based multi-party verifiable online card games |
| [SecretFlow-SPU](https://github.com/secretflow/spu) | - | SecretFlow-SPU is a provable MPC-powered machine learning framework that provides significant performance improvements over MP-SPDZ and TF Encrypted. |