| **Name** | **Party #** | **Honesty** | **Operations** | **Purpose** | **Docs/Updates** | **Techniques** | **Performance** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [TNO-MPC](https://github.com/TNO-MPC/) | Tested on 3PC, supports others | Semi-honest | **Secure:** integer-based arithmetic, bitwise operations, comparison operators, NumPy arrays, basic floating point arithmetic, all(), any(), sum(), min(), max(), sorted() | Secure FL-based machine learning | Comprehensive documentation, no paper, last updated 1 year ago | Shamir secret sharing, homomorphic encryption | Unknown |
| [FRESCO](https://github.com/aicis/fresco) | 2PC for TinyTables, >=2PC for SPDZ | Semi-honest for TinyTables, malicious for SPDZ | **Not Secure:**  Dummy boolean, dummy arithmetic  **Secure:**  TinyTables, SPDZ | Development of secure MPC applications with mixed-protocol | Mediocre documentation, no paper, last updated 1 year ago | **TinyTables:** Preprocesses function before the input is known by creating a small table of values for each and gate involved  **SPDZ:**  Additive secret sharing, Beaver triples, | Advertises high scalability |
| [MOTION](https://github.com/encryptogroup/MOTION) | >=2PC, performs best at <=16PC | Semi-honest ONLY, up to *n*-1 corrupted parties | **Secure:**  boolean, arithmetic, Yao | Development of secure MPC applications with mixed-protocol | Comprehensive documentation, comprehensive paper, last updated 1 year ago | Oblivious transfer, Beaver triples, square pairs, shared bits, fixed-key AES | 2x better than OT, 16x better than AES-128, 10-18x better than MP-SPDZ, 190-586x better than SCALE-MAMBA |
| [SCALE-MAMBA](https://github.com/KULeuven-COSIC/SCALE-MAMBA) | >=2PC | Up to malicious, dishonest majority parties | **Secure:**  inverse, random, prefix multiplication, OR, XOR, prefix operator, random shares, addition, mod, multiplication, →unary, truncation, oblivious truncation, absolute value, negative, division, =0, >0, <=0, >=0, =, <, >, <=. >=. Also supports floating point arithmetic | Industrial MPC with actively secure MPC protocol | Comprehensive documentation, no paper, last updated 1 year ago | BDOZ, SPDZ, TinyOT, HSS, FHE key | Unknown |
| [TinySMPC](https://github.com/kennysong/tinysmpc) | N/A | N/A | **Secure**:  addition, subtraction, multiplication, exponentiation, > | Education | Mediocre documentation, no paper, last updated 3 years ago | Additive secret sharing | Unknown |
| [SecretFlow-SPU](https://github.com/secretflow/spu) | Tested on 3PC, supports others | Semi-honest | **Secure:**  mixed-data-type multiplication fusion, mixed-visibility multiplication operands reorder, inverse sqrt transformation, select predicate reuse, max-pooling transformation, multiplication, truncation | Secure machine learning | Comprehensive documentation, comprehensive paper, updated within the last week | ABY3 threat model, SPDZ2k, Cheetah | 4.1x better than MP-SPDZ, 2.3x better than TF Encrypted |

# TNO-MPC

* Aims to move away from traditional single party central authorities and into private-preserving computation. TNO-MPC aims to blend elements of MPC and federated learning (FL) for more secure machine learning solutions.
* Based on Shamir secret sharing and homomorphic encryption
* Provides a general base of software for building various MPC solutions called MPyC
  + Includes basic machine learning models, namely logistic regression and SVMs
  + Can be easily simulated on 3PC, but likely work with other PC
* Secure learning features:
  + Data: Grid Search, Cross Validation
  + Training: Gradient descent
  + Models: ElasticNets, Lasso, Linear, Logistic, Ridge, SVM
* Integrated sklearn features:
  + Utilities: ParameterGrid, BaseCrossValidator, BaseKFold, KFold
  + Metrics: Accuracy, Precision, Recall, F1, R2, MSE
* Limitations:
  + No code is provided to securely apply the trained models

# FRESCO

* FRESCO aims to provide a more efficient solution for structuring secure computation
  + Can be thought of as a hub for connecting applications with protocols
  + Easy to write applications and specify which data to open and which to close
  + Provides a standard library of MPC software
* Allows for implementation of multiple runtimes such as VIFF and FairPlay at once
* Has support for TinyTables and SPDZ
* Advertises scalability to large computations
* Designed to reduce memory usage for large circuits (like VIFF)
  + Done by implicitly representing the structure of the circuit as code with rounds of evaluation
  + Several gates are made explicit and evaluated in parallel per round
    - Each gate is only allowed to send and receive once over the network
* FRESCO has been used to:
  + Compute statistical data from surveys without revealing individual responses
  + Benchmark performance of companies without revealing individual company data
  + Let banks credit rate potential customers without revealing customer data or credit rating functions of banks

# MOTION

* Generic open-source framework that allows mixed-protocol MPC
* Enables MPC over arbitrary messaging interfaces (no need for owning network sockets)
* Performs significantly better than precomputed correlated OT, MP-SPDZ, and SCALE-MAMBA
* Also supports and is efficient for privacy-preserving neural network inference
* Based on a different type of adversary, not active security:
  + Parties follow the protocol but try to infer additional information about the other parties’ inputs from the transcript
  + Advertises full-threshold security against any number of passive adversaries
* Supports multiple MPC protocols and conversion between them to benefit from each strengths
* Claims to be first MPC framework supporting >2PC for non-honest majority, but with best performance with <=16PC
* Works at core based on ABY protocol, but implements several novel design aspects
  + Communication serialization: Aims to address the issue of owning network sockets.
  + Provider-based use of MPC Primitives: Developer does not need to know MPC primitive-network interactions. Also enables the developer to easily replace the computation procedure partially or completely
  + Multiple layers of abstraction
  + HyCC integration: HyCC is the software that efficiently and securely combines MPC protocols for mixed-protocol MPC
  + Asynchronous gate evaluation: Allows evaluation of arbitrary circuits without information about their structure
  + Code vectorization: Vectorization of CPU instructions for improved efficiency. One of only three MPC frameworks written in a low-level programming language (others are ABY and MP-SPDZ)
  + SIMD: Utilizes vectors of bit instead of single bits, decreasing overhead by orders of magnitude
  + Interleaved setup and online phase: Claims to be first framework to offer both sequential and interleaved circuit evaluation, giving exactly required amount of correlated randomness
* Supports many MPC protocols
  + Arithmetic sharing
  + Boolean sharing with GMW
  + Yao sharing with BMR
  + Boolean→Yao
  + Yao→Boolean
  + Boolean→Arithmetic
  + Arithmetic→Yao
  + Arithmetic→Boolean
  + Yao→Arithmetic

# SCALE-MAMBA

* SCALE-MAMBA arose out of BDOZ, SPDZ, TinyOT, and HSS papers
  + Was implemented out of SPDZ system with similar code base
  + Close to productions system, and is a complete system
* Supports a wide variety of functions on a multi-party system
* SCALE consists of three sub-systems: offline phase, online phase, and compiler
  + Offline phase and online phase are fully integrated
  + System is almost secure out of the box
  + Supports a variety of secret sharing schemes, such as Full Threshold
* MAMBA is the language of the compiler, which takes in programs

# TinySMPC

* Intended for educational purposes
* Unique aspect is that it allows programmers to see encrypted values on their end
* Uses additive secret sharing for private shares and additionally claims to implement several other SMPC protocols for direct computation on private data
* Potentially could be made simulation-usable via several improvements

# 

# SecretFlowSPU

* Introduces the Secure Processing Unit (SPU) which claims to be a provable and measurable secure computation device
  + Core components are a frontend compiler and backend MPC runtime
    - Backend employes vectorization and streaming
  + Not bound to one specific framework and can support various libraries via PPHLO
    - PPHLO represents a tensor via <Shape, Data Type, Visibility>
  + Can use several different protocols, including 3PC ABY3, *n*PC SPDZ2k, and 2PC Cheetah
  + Employes the SPMD programming model, where all nodes receive the same PPHLO to execute
* Focuses on provable security which uses a secure runtime for XLA-like tensor operations
* Is compatible with existing ML programs, and does not require users to restructure the project
* Boasts performance improvements over MP-SPDZ and TF Encrypted