

CLAIMS:

1. A computer-implemented method for data processing, comprising:

obtaining a bytecode compiled from source code comprising one or more input parameters, the source code including an encoding function to encode the one or more input parameters, save the encoded one or more input parameters in a memory segment, and provide a memory location of the memory segment;

executing, according to the bytecode, the encoding function to encode the one or more input parameters to obtain the memory location of the memory segment storing the encoded one or more input parameters; and

providing the memory location to a function for retrieving and decoding the encoded one or more input parameters to obtain the one or more input parameters.

2. The method of claim 1, wherein:

at least one of the one or more input parameters comprises a pointer type variable.

3. The method of claim 2, wherein:

the pointer type variable comprises a multiple indirection type variable.

4. The method of any of claims 1-3, wherein:

the memory location of the memory segment comprises a pointer corresponding to a starting or ending address of the memory segment and a data length corresponding to the memory segment.

5. The method of any of claims 1-3, wherein:

the memory location of the memory segment comprises a first pointer corresponding to a starting address of the memory segment and a second pointer corresponding to an ending address of the memory segment.

6. The method of any preceding claim, wherein:

the method is performed by a virtual machine; and

obtaining the bytecode compiled from the source code comprising the one or more input parameters comprises: obtaining, from a computing device, the bytecode compiled by a

compiler associated with the computing device from the source code comprising the one or more input parameters.

7. The method of claim 6, wherein:

the virtual machine and the compiler have different word sizes.

8. The method of any of claims 6 and 7, wherein:

the virtual machine is based on a WebAssembly (WASM) format.

9. The method of any of claims 1-3 and 6-8, wherein the memory location comprises a pointer storing an address corresponding to the memory segment, and the address is shifted by compilation in the bytecode compiled from the source code.

10. The method of claim 9, further comprising:

executing the function, wherein executing the function comprises:

restoring the shifted address according to a storage starting location of the virtual machine and an address shift obtained by the virtual machine;

retrieving the encoded one or more input parameters according to the restored address; and

decoding the encoded one or more input parameters to retrieve the one or more input parameters.

11. The method of claim 10, wherein executing the function further comprises:

invoking another function based on the one or more input parameters to return one or more output parameters; and

encoding the one or more output parameters.

12. The method of claim 11, wherein invoking the another function based on the one or more input parameters to return the one or more output parameters comprises:

invoking the another function to return a state of a blockchain.

13. A system for data processing, comprising:

one or more processors; and

one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to perform the method of any of claims 1 to 12.

14. An apparatus for data processing, comprising a plurality of modules for performing the method of any of claims 1 to 12.

15. A non-transitory computer-readable medium having stored therein instructions that, when executed by a processor of a device, cause the device to perform the method of any of claims 1 to 12.