CLAIMS:

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

obtaining 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, wherein the memory location comprises an address corresponding to the memory segment, and the address is shifted by compilation in the bytecode compiled from the source code;

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;

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; and

executing the function by:

restoring the shifted address according to a storage starting location of a 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.

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 claim 1, 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 claim 1, 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 claim 1, wherein:

the method is performed by the 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 claim 6, wherein:

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

9. The method of claim 1, 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.

10. The method of claim 9, 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.

11. A system for data processing comprising one or more processors and one or more non-transitory computer-readable memories coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the system to perform operations comprising:

obtaining 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, wherein the memory location comprises an address corresponding to the memory segment, and the address is shifted by compilation in the bytecode compiled from the source code;

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;

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; and

executing the function by:

restoring the shifted address according to a storage starting location of a 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.

12. The system of claim 11, wherein:

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

13. The system of claim 11, wherein:

the system is implemented in the 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.

14. The system of claim 13, wherein:

the virtual machine and the compiler have different word sizes.

15. A non-transitory computer-readable storage medium for data processing, the storage medium coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the one or more processors to perform operations comprising:

obtaining 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, wherein the memory location comprises an address corresponding to the memory segment, and the address is shifted by compilation in the bytecode compiled from the source code;

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;

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; and

executing the function by:

restoring the shifted address according to a storage starting location of a 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.

16. The storage medium of claim 15, wherein:

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

17. The storage medium of claim 15, wherein:

the storage medium is implemented in the 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.

18. The storage medium of claim 17, wherein:

the virtual machine and the compiler have different word sizes.

19. The storage medium of claim 15, wherein:

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

20. The storage medium of claim 19, wherein:

the pointer type variable comprises a multiple indirection type variable.