CLAIMS

1. A device comprising:

a storage system including one or more computer-readable storage mediums, wherein the storage system stores:

a computer program comprised of a set of encoded computer-executable instructions; and

a genomic data set that includes a genomic differentiation object that was used to encode the set of encoded computer-executable instructions;

genomic regulation instructions (GRI) corresponding to the computer program, wherein the genomic regulation instructions were used to encode the set of encoded computer-executable instructions;

a processing system comprising:

a VDAX that is configured to:

receive encoded instructions to be executed from the set of encoded computer-executable instructions; and

decode the encoded instructions into decoded executable instruction based on a modified genomic differentiation object and sequences extracted from metadata associated with the encoded instructions, wherein the modified genomic differentiation object is modified from the genomic differentiation object using the GRI; and

a set of processing cores that are configured to receive the decoded executable instructions from the VDAX and to execute the decoded executable instructions.

2. The device of claim 1, wherein the VDAX is configured to:

modify the genomic differentiation object based on the GRI using a set of computational functions to obtain the modified genomic differentiation object; and

for each respective encoded instruction that is to be executed by the processing device:

obtain a genomic engagement factor (GEF) corresponding to the respective encoded computer-executable instruction, wherein the GEF is generated based on a sequence that is extracted from metadata corresponding to the respective encoded computer-executable instruction, the GRI, and the modified genomic data object;

decode the respective encoded instruction using the GEF corresponding to the respective encoded instruction to be executed to obtain a respective decoded executable instruction; and

output the respective decoded instruction to a processing core of the set of processing cores.

3. The device of claim 2, wherein decoding the respective encoded instruction using the GEF includes decrypting the respective encoded instruction using the GEF as a key to obtain the respective decoded instruction.

4. The device of claim 2, wherein decoding the respective encoded instruction using the GEF includes disambiguating the respective decoded instruction from the respective encoded instruction using the respective GEF, and wherein disambiguating the respective decoded instruction from the respective encoded instruction includes executing an XOR operation using the encoded instruction and the GEF as input to the XOR operation.

5. The device of claim 2, wherein each respective encoded instruction was encoded with a different respective GEF, such that each respective GEF is determined based on a respective sequence extracted from respective metadata that uniquely corresponds to the respective encoded instruction.

6. The device of claim 5, wherein the respective metadata that uniquely corresponds to the respective encoded instruction includes one or more of random-access memory (RAM) metadata, application metadata, operating system-application metadata, and filesystem metadata.

7. The device of claim 6, wherein the respective sequence that is used to generate the respective GEF that is used to decode the respective encoded instruction is a set of bits extracted from a bit representation of the respective metadata.

8. The device of claim 2, wherein the set of encoded computer-executable instructions includes a plurality of sub-sets of instructions, such that each subset of plurality of subsets is decoded with a respective GEF that is generated based on a respective sequence extracted from metadata that respectively corresponds to the respective subset of computer-executable instructions.

9. The device of claim 8, wherein the respective metadata from which the respective sequence was extracted includes one or more of random-access memory (RAM) metadata corresponding to the subset, application metadata corresponding to the subset, operating system-

~~application metadata corresponding to the subset, and filesystem metadata corresponding to the subset.~~

~~10. The device of claim 2, wherein obtaining the GEF corresponding to the respective encoded computer executable instruction includes sequence mapping the sequence into the modified genomic data object to obtain the GEF, and wherein sequence mapping the sequence into the modified genomic data object includes:~~

~~generating a sequence conversion vector (SCV) based on the sequence and the GRI;~~

~~mapping the modified genomic differentiation object based on the SCV to obtain a mapped modified genomic differentiation object; and~~

~~generating the GEF based on the mapped modified genomic differentiation object.~~

~~1~~1. A method for executing computer programs in a trusted execution environment of a device comprising:

retrieving a genomic differentiation object corresponding to a set of one or more computer programs including a computer program;

modifying the genomic differentiation object based on genomic regulation instructions (GRI) to obtain a modified genomic differentiation object;

executing a first executable instruction of the computer program, wherein executing the first executable instruction includes:

retrieving first encoded data that is input to the first executable instruction;

extracting a first sequence from first metadata associated with the first encoded data;

generating a first genomic engagement factor (GEF) based on the first sequence, the GRI and, and the modified genomic differentiation object;

decoding the first encoded data based on the first GEF to obtain first decoded data;

executing the first executable instruction using the first decoded data; and

executing a second executable instruction of the computer program, wherein executing the second executable instruction includes:

retrieving second encoded data that is input to the second executable instruction;

extracting a second sequence from second metadata associated with the second encoded data;

generating a second GEF based on the second sequence, the GRI and, and the modified genomic differentiation object;

decoding the second encoded data based on the second GEF to obtain second decoded data; and

executing the second executable instruction using the second decoded data.

~~1~~2. The method of claim ~~1~~1, wherein executing the first executable instruction further includes:

obtaining output data resulting from execution of the first executable instruction;

generating a third GEF based on a third sequence corresponding to the output data, the GRI, and the modified genomic differentiation object;

transforming the output data into third encoded data based on the third GEF, and

writing the third encoded data to memory.

3. The method of claim 2, further comprising writing the third encoded data to memory.

~~13.~~4. The method of claim ~~1~~2, wherein the third sequence is extracted from metadata corresponding to the output data, wherein the metadata includes one or more of random-access memory (RAM) metadata, application metadata, operating system-application metadata, and filesystem metadata, the method further comprising executing a third executable instruction of the computer program by:

retrieving the third encoded data that is input to the third executable instruction;

extracting the third sequence from a value representing the metadata corresponding to the output data;

generating the third GEF based on the third sequence, the GRI and, and the modified genomic differentiation object;

decoding the third encoded data based on the third GEF to obtain the output data; and

executing the first executable instruction using the output data resulting from execution of the first executable instruction.

5. The method of claim 4, further comprising executing a third executable instruction of the computer program by:

retrieving the third encoded data that is input to the third executable instruction;

extracting the third sequence from a value representing the metadata corresponding to the output data;

generating the third GEF based on the third sequence, the GRI and, and the modified genomic differentiation object;

decoding the third encoded data based on the third GEF to obtain the output data; and

executing the first executable instruction using the output data resulting from execution of the first executable instruction.

~~14.~~6. The method of claim ~~1~~1, wherein the first metadata from which the first sequence is extracted and the second metadata from which the second sequence is extracted includes one or more of random-access memory (RAM) metadata, application metadata, operating system-application metadata, and filesystem metadata.

7. The method of claim 1, wherein generating the first GEF includes:

generating a sequence conversion vector (SCV) based on the first sequence and the first GRI;

mapping the modified genomic differentiation object based on the SCV to obtain a mapped modified genomic differentiation object; and

generating the first GEF based on the mapped modified genomic differentiation object.

8. The method of claim 7, wherein generating the SCV includes combining the first sequence and the GRI.

9. The method of claim 8, wherein generating the SCV further includes applying one or more computational functions to the combination of the first sequence and the GRI to obtain the SCV.

~~15.~~10. The method of claim ~~1~~1, wherein decoding the first encoded data includes decrypting the first encoded data using the first GEF and decoding the second encoded data includes decrypting the second encoded data using the second GEF.

~~16.~~11. The method of claim ~~1~~1, wherein decoding the first encoded data includes disambiguating the first encoded data using the first GEF and disambiguating the second encoded data includes decrypting the second encoded data using the second GEF.

12. The method of claim 1, wherein the genomic differentiation object is an ZNA object.

13. The method of claim 12, wherein the ZNA object is an N x M binary matrix.

14. The method of claim 12, wherein a VDAX of the device generates and assigns the ZNA object to the computer program.

15. The method of claim 1, wherein the device is: a server computing device, a mobile computing device, a personal computing device, a vehicle, a gaming device, an internet of things device, a sensor device, a video recording device, a wearable computing device, a smart appliance, or a network router.

16. The method of claim 1, wherein the computer program is: a software application, a middleware application, a firmware application, or an operating system.