

CLAIMS

1. A method for protecting software from piracy, the method comprising:
receiving multiple binary files, each binary file comprising code and data;
modifying the multiple binary files such that the data of each binary file may be located at an arbitrary distance in memory from the code of that binary file;
encrypting the code of each binary file; and
receiving, from a computing device, a request for a decryption key, wherein the computing device includes a hardware enclave and wherein the encrypted code of each binary file but not the data of each binary file is stored in the hardware enclave.
2. The method of claim 1 further comprising: providing, to the computing device, the decryption key.
3. The method of claim 2 further comprising: authenticating, before providing the decryption key, a processor signature, wherein a processor of the computing device generated the processor signature and the request for the decryption key includes the processor signature.
4. The method of claim 3 further comprising: validating, before providing the decryption key, a hash, wherein the request for the decryption key includes the hash.
5. The method of claim 1, wherein modifying the multiple binary files includes identifying data references in the code of each binary file and modifying the data references.
6. The method of claim 5, wherein modifying the multiple binary files is done without access to source code or debug symbols for the multiple binary files.
7. A system for facilitating protecting a software program from piracy, comprising:
one or more processors;
memory in electronic communication with the one or more processors; and
instructions stored in the memory, the instructions being executable by the one or more processors to:
receive multiple binary files, each binary file comprising code and data;
modify the multiple binary files such that the data of each binary file may be located at an arbitrary distance in memory from the code of that binary file;
encrypt the code of each binary file but not the data of each binary file; and
receive, from a computing device, a request for a decryption key, wherein the computing device includes a hardware enclave, wherein the encrypted code of each binary file but not the data of each binary file is stored in the hardware

enclave, and wherein the hardware enclave includes instructions to mark the hardware enclave as non-readable before executing the code.

8. The system of claim 7, the instructions further being executable by the one or more processors to: provide, to the computing device, the decryption key.

9. The system of claim 8, the instructions further being executable by the one or more processors to: authenticate, before providing the decryption key, a processor signature, wherein a processor of the computing device generated the processor signature and the request for the decryption key includes the processor signature.

10. The system of claim 9, the instructions further being executable by the one or more processors to: validate, before providing the decryption key, a hash, wherein the request for the decryption key includes the hash and validating the hash comprises comparing the hash to a verified hash value.

11. A computer-readable medium comprising instructions that are executable by one or more processors to cause a computing system to:

receive a request to initiate an application, the application comprising one or more files with executable code and data, wherein the executable code is encrypted;

load the executable code into a hardware enclave on the computing system;

load the data into a location in memory on the computing system, wherein the location is outside the hardware enclave and is not a predetermined distance in the memory from the executable code;

send a request for a decryption key to an authentication server;

decrypt the executable code using the decryption key, wherein the decryption key is received by the computing system after sending the request for the decryption key; and

mark the hardware enclave as non-readable.

12. The computer-readable medium of claim 11, further comprising additional instructions that are executable by the one or more processors to cause the computing system to: load startup code into the hardware enclave.

13. The computer-readable medium of claim 12, further comprising additional instructions that are executable by the one or more processors to cause the computing system to: measure a hash of the startup code and the executable code.

14. The computer-readable medium of claim 13, further comprising additional instructions that are executable by the one or more processors to cause the computing system to: sign the hash with a signature of the one or more processors.

15. The computer-readable medium of claim 14, wherein the request for the decryption

key includes the signature and the hash.