CLAIMS

1. A method for providing data protection in an integrated circuit, the method comprising: detecting a change to software or firmware; determining whether a user is authentic in response to detecting the change to the software or the firmware; generating a false cipher-based message authentication code (CMAC) value that is not based on a Hardware Unique Key (HUK) associated with the integrated circuit in response to the user not being authentic; determining that the change to the software or the firmware was unauthorized based at least in part on the false CMAC value not matching an expected CMAC value; and corrupting an anti-replay counter (ARC) value, maintained in a one-time programmable memory of the integrated circuit and used by the integrated circuit to protect contents of a non-volatile memory, responsive to determining that the change to the software or the firmware was unauthorized.

2. The method of claim 1, wherein corrupting the ARC value comprises: updating an ARC value stored in the one-time programmable memory with a current ARC value maintained in volatile memory.

3. The method of claim 2, further comprising: determining a first number of bits of the one-time programmable memory storing the ARC value to be set; determining locations of a first set of bits of the one-time programmable memory comprising the first number of bits; and updating the ARC value stored in the one-time programmable memory by setting the first set of bits of the one-time programmable memory.

4. The method of claim 3, wherein determining the first number of bits of the one-time programmable memory to set comprises: determining the first number of bits of the one-time programmable memory to set based on a second number of bits already set in the one-time programmable memory.

5. The method of claim 3, wherein determining the first set of bits of the one-time programmable memory to be set comprises: randomly selecting the first set of bits of the one-time programmable memory from a second set of bits that have not been set in the one-time programmable memory.

6. The method of claim 1 wherein the Hardware Unique Key (HUK) comprises a bit string stored in a secure memory location of a computing device.

7. The method of claim 1 wherein the Hardware Unique Key (HUK) comprises a bit string stored in a one-time programmable memory of a computing device.

8. An integrated circuit providing data protection, the integrated circuit comprising: a one-time programmable memory; a processor communicably coupled to the one-time programmable memory, the processor configured to: detect a change to software or firmware; determine whether a user is authentic in response to detecting the change to the software or the firmware; generate a false cipher-based message authentication code (CMAC) value that is not based on a Hardware Unique Key (HUK) associated with the integrated circuit in response to the user not being authentic; determine that the change to the software or the firmware was unauthorized based at least in part on the false CMAC value not matching an expected CMAC value; and corrupt an anti-replay counter (ARC) value, maintained in the one-time programmable memory of the integrated circuit and used by the integrated circuit to protect contents of a non-volatile memory, responsive to determining that the change to the software or the firmware was unauthorized.

9. The integrated circuit of claim 8, wherein the processor being configured to corrupt the ARC value is further configured to: update a first ARC value stored in the one-time programmable memory with a second ARC value maintained in volatile memory of the integrated circuit.

10. The integrated circuit of claim 9, wherein the processor is further configured to: determine a first number of bits of the one-time programmable memory storing the ARC value to be set; determine locations of a first set of bits of the one-time programmable memory comprising the first number of bits; and update the ARC value stored in the one-time programmable memory by setting the first set of bits of the one-time programmable memory.

11. The integrated circuit of claim 10, wherein the processor being configured to determine the

first number of bits of the one-time programmable memory to set is further configured to: determine the first number of bits of the one-time programmable memory to set based on a second number of bits already set in the one-time programmable memory.

12. The integrated circuit of claim 10, wherein the processor being configured to determine which bits of the one-time programmable memory to set is further configured to: randomly select the first set of bits of the one-time programmable memory from a second set of bits that have not been set in the one-time programmable memory.

13. A non-transitory, computer-readable medium, having stored thereon computer-readable instructions for providing data protection in an integrated circuit, comprising instructions configured to cause a computer to: detect a change to software or firmware; determine whether a user is authentic in response to detecting the change to the software or the firmware; generate a false cipher-based message authentication code (CMAC) value that is not based on a Hardware Unique Key (HUK) associated with the integrated circuit in response to the user not being authentic; determine that the change to the software or the firmware was unauthorized based at least in art on the false CMAC value not matching an expected CMAC value; and corrupt an anti-replay counter (ARC) value, maintained in a one-time programmable memory of the integrated circuit and used by the integrated circuit to protect contents of a non-volatile memory, responsive to determining that the change to the software or the firmware was unauthorized.

14. The non-transitory, computer-readable medium of claim 13, wherein the instructions configured to cause the computer to corrupt the ARC value further comprise instructions configured to cause the computer to: update a first ARC value stored in the one-time programmable memory with a second ARC value maintained in volatile memory of the integrated circuit.

15. The non-transitory, computer-readable medium of claim 14, further comprising instructions configured to cause the computer to: determine a first number of bits of the one-time programmable memory storing the ARC value to be set; determine locations of a first set of bits of the one-time programmable memory comprising the first number of bits; and update the ARC

value stored in the one-time programmable memory by setting the first set of bits of the one-time programmable memory.

16. The non-transitory, computer-readable medium of claim 15, wherein the instructions configured to cause the computer to determine the first number of bits of the one-time programmable memory to set further comprise instructions configured to cause the computer to: determine the first number of bits of the one-time programmable memory to set based on a second number of bits already set in the one-time programmable memory.

17. The non-transitory, computer-readable medium of claim 16, wherein the instructions configured to cause the computer to determine which bits of the one-time programmable memory to set further comprise instructions configured to cause the computer to: randomly select the first set of bits of the one-time programmable memory from a second set of bits that have not been set in the one-time programmable memory.