~~CLAIMS~~

~~What is claimed is:~~

CLAIMS

1.- A method for providing data protection in an integrated circuit, the method comprising:

~~determining that an unauthorized update has been made~~ 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 ~~unauthorized update has been made to the software or the firmware.~~

~~2. The method of claim 1, wherein determining that the unauthorized update has been made to the software or the firmware associated with the integrated circuit comprises:~~

~~detecting a change to the software or the firmware;~~

~~determining whether at least one of a user or the software or the firmware is authentic in response to detecting the change to the software or the firmware; and~~

~~determining that the~~ change to the software or the firmware was unauthorized ~~in response the at least one of the user or the software not being authentic.~~

~~3. The method of claim 2, wherein the method comprises authenticating the user, and authenticating the user comprises:~~

~~determining whether the user has been authenticated; and~~

~~performing at least one authentication procedure to authenticate the user in response to the user not having been authenticated.~~

~~4. The method of claim 2, wherein the method comprises authenticating the software or the firmware, and authenticating the software or the firmware comprises:~~

~~determining a message authentication code (MAC) for the software or the firmware by applying a cipher-based message authentication code (CMAC) algorithm to the software or the firmware using a key derived from a Hardware Unique Key (HUK) associated with the integrated circuit; and~~

~~comparing the MAC with a previously determined MAC to determine whether the MAC matches the previously determined MAC~~.

~~5.~~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.

~~6.~~3. The method of claim ~~5~~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.

~~7.~~4. The method of claim ~~6~~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.

~~8.~~5. The method of claim ~~6~~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.

~~9.~~8. An integrated circuit providing data protection, the integrated circuit comprising:

~~means for determining that an unauthorized update has been made to~~ 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 ~~associated with the integrated circuit; and~~

~~means for 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 unauthorized update has been made to the software or the firmware.~~

~~10. The integrated circuit of claim 9, wherein the means for determining that the unauthorized update has been made to the software or the firmware comprises:~~

~~means for detecting a change to the software or the firmware;~~

~~means for determining;~~ determine whether ~~at least one of~~ a user ~~or the software or the firmware~~ is authentic in response to detecting the change to the software or the firmware; ~~and~~

~~means for determining that~~ 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 ~~in response the at least one of the user or the software or the firmware not being authentic.~~

~~11. The integrated circuit of claim 10, wherein the means for determining whether at least one of the user or the software or the firmware is authentic comprises:~~

~~means for determining whether the user has been authenticated; and~~

~~means for performing at least one authentication procedure to authenticate the user in response to the user not having been authenticated.~~

~~12. The integrated circuit of claim 10, wherein the means for determining whether at least one of the user or the software or the firmware is authentic comprises:~~

~~means for determining a message authentication code (MAC) for the software or the firmware by applying a cipher-based message authentication code (CMAC) algorithm to the software or the firmware using a key derived from a Hardware Unique Key (HUK) associated with the integrated circuit; and~~

~~means for comparing the MAC with a previously determined MAC to determine whether the MAC matches the previously determined MAC.~~

~~13. The integrated circuit of claim 9, wherein the means for corrupting the ARC~~based at least in part on the false CMAC value ~~further comprise:~~

~~means for updating a first ARC~~not matching an expected CMAC value ~~stored in the one-time programmable memory with a second ARC value maintained in volatile memory of the integrated circuit.~~

~~14. The integrated circuit of claim 13, further comprising:~~

~~means for determining a first number of bits of the one-time programmable memory to be set;~~

~~means for determining locations of a first set of bits of the one-time programmable memory comprising the first number of bits; and~~

~~means for updating the ARC value stored in the one-time programmable memory by setting the first set of bits of the one-time programmable memory.~~

~~15. The integrated circuit of claim 14, wherein the means for determining the first number of bits of the one-time programmable memory to set further comprise:~~

means for 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.

16. The integrated circuit of claim 14, wherein the means for determining which bits of the one-time programmable memory to set further comprise:

means for 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.

17. 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:

determine that an unauthorized update has been made to software or firmware associated with the integrated circuit; and

; 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 unauthorized update has been made to the software or the firmware.

18. The integrated circuit of claim 17, wherein the processor is configured to determine that the unauthorized update has been made to the software or the firmware by being configured to:

detect a change to the software or the firmware;

authenticate at least one of a user and the software or the firmware in response to detecting the change to the software or the firmware; and

determine that the change to the software or the firmware was unauthorized in response the at least one of the user and the software not being authenticated.

19.9. The integrated circuit of claim 18, wherein the processor is configured to authenticate the user by being configured to:

determine whether the user has been authenticated; and

~~perform at least one authentication procedure to authenticate the user in response to the user not having been authenticated.~~

~~20. The integrated circuit of claim 18, wherein the processor is configured to authenticate the software by being configured to:~~

~~determine a message authentication code (MAC) for the software by applying a cipher-based message authentication code (CMAC) algorithm to the software using a key derived from a Hardware Unique Key (HUK) associated with the integrated circuit; and~~

~~compare the MAC with a previously determined MAC to determine whether the MAC matches the previously determined MAC.~~

~~21. The integrated circuit of claim 17~~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.

~~22.~~10. The integrated circuit of claim ~~21~~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.

~~23.~~11. The integrated circuit of claim ~~22~~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.

~~24.~~12. The integrated circuit of claim ~~22~~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.

~~25.~~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:

~~determine that an unauthorized update has been made to software or firmware associated with the integrated circuit; and~~

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 ~~unauthorized update has been made~~change to the software or the firmware~~.~~ was unauthorized.

~~26.~~14. The non-transitory, computer-readable medium of claim ~~25, wherein the instructions configured to cause the computer to determine that the unauthorized update has been made to the software or the firmware further comprise instructions configured to cause the computer to:~~

~~detect a change to the software or the firmware;~~

~~authenticate at least one of a user and the software or the firmware in response to detecting the change to the software or the firmware; and~~

~~determine that the change to the software or the firmware was unauthorized in response the at least one of the user and the software or the firmware not being authenticated.~~

~~27. The non-transitory, computer-readable medium of claim 25~~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.

~~28.~~15. The non-transitory, computer-readable medium of claim ~~27~~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.

~~29.~~16. The non-transitory, computer-readable medium of claim ~~28~~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.

~~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.~~

~~30. The non-transitory, computer-readable medium of claim 29, 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.~~