CLAIMS:

1. A computer implemented method for performing cryptographic operations on a message A using a computing device having two or more processors, the method comprising the computing device:

retrieving the message A from a register, wherein the message A comprises contiguous ordered digits, each digit being characterized by a weight w and identified as A[w], where w is an integer from 0 to (n-1), and where n represents the maximum number of digits in the message;

performing cryptographic operations on the message by squaring the message A, wherein the squaring the message A comprises:

computing and summing, by a processor, non-squaring terms for the message A, wherein values for the computed and summed non-squaring terms are stored in a register R[ ], wherein register R[ ] comprises a (n x 2)-bit sized array;

in response to a determination that a value in R[2·w + 1] of the register R[ ] has been set, computing, by another processor, the squaring term for another digit represented by A[w] simultaneously as the step of computing and summing non-squaring terms for the message A is being carried out by the processor; and

computing, by the processor, squaring terms for the message A and adding, by the processor, the computed squared terms to the values stored in register R[ ] wherein the values stored in register R[ ] are multiplied by two before the values are added to the computed squared terms, to generate the

square of message A, whereby the square of message A is utilized in cryptographic operations.

2. The computer implemented method according to claim 1,

wherein the computing and the summing the non-squaring terms for the message A include:

initializing (MSD, LSD) = 0 and all terms in R[ ] = 0;

for each integer i with $0 \leq i < (n/2)$, performing by the processor the following operations:

for each integer j with $(i+1) \leq j < (n-i)$ do,

set (MSD, LSD) = A[i] x A[j] + MSD + R[j+i]; and

set R[j+i] = LSD;

set count = i;

when (count < 2·i) do,

set (MSD, LSD) = A[j] x A[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase count by 1;

set (MSD,LSD) = A[j] x A[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase j by 1;

set R[j+count] = R[j+count] + MSD;

returning register R with set values; and

wherein the computing the squaring terms for the message A and adding the computed squared terms to the terms stored in register R include:

initializing (MSD, LSD) = 0;

for each integer i with $0 \leq i < n$, performing by the processor the following operations:

set (MSD,LSD) = A[i] x A[i] + MSD;

set (MSD', LSD) = R[i·2]·2 + LSD;

set R[i·2] = LSD;

set (MSD,LSD) = R[i·2 + 1] ·2 + MSD' + MSD; and

set R[i·2 + 1] = LSD;

returning values stored in register R as the square of the message A,

where, MSD is a most significant digit of an integer and LSD is a least significant digit of the integer.

3. The computer implemented method according to claim 2 wherein the computing and the summing the non-squaring terms for the message A further comprises:

determining, by the processor, if a value of R[2·i - 1] in the register R has been set; and

in response to a determination that the value of R[2·i - 1] has been set, completing the remainder of the operations for the value of i using the processor and simultaneously, increasing the value of i by 1 wherein,

for each integer i whereby $i < (n/2)$, performing by the another processor simultaneously the following operations:

for each integer j with $(i+1) \leq j < (n-i)$ do,

set (MSD, LSD) = A[i] x A[j] + MSD + R[j+i]; and

set R[j+i] = LSD;

set count = i;

when (count < 2·i) do,

set (MSD, LSD) = A[j] x a[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase count by 1;

set (MSD,LSD) = A[j] x A[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase j by 1; and

returning register R with set values.

4. The computer implemented method according to claim 1 further comprising:

applying a Montgomery modular reduction for a modulus value of m to the square of the message A.

5. The computer implemented method according to claim 1 further comprising:

applying a Barett modular reduction for a modulus value of m to the square of the message A.

6. A system for performing cryptographic operations on a message A comprising:

two or more processing units; and

a non-transitory media readable by the two or more processing units, the non-transitory media storing instructions that when executed by a processing unit, cause the processing unit to:

retrieve the message A from a register, wherein the message A comprises contiguous ordered digits, each digit being characterized by a weight w and identified

as A[w], where w is an integer from 0 to (n-1), and where n represents the maximum number of digits in the message;

perform cryptographic operations on the message by squaring the message A, wherein the squaring the message A comprises:

instructions for directing the processing unit to:

compute and sum non-squaring terms for the message A, wherein values for the computed and summed non-squaring terms are stored in a register R[ ], wherein register R[ ] comprises a (n x 2)-bit sized array;

in response to a determination that a value in R[2·w + 1] of the register R[ ] has been set, causing another processing unit to execute instructions stored in the non-transitory media to compute the squaring term for another digit represented by A[w] simultaneously as the step of computing and summing non-squaring terms for the message A is being carried out by the processing unit; and

compute squaring terms for the message A; and

add the computed squared terms to the values stored in register R[ ], wherein the values stored in register R[ ] are multiplied by two before the values are added to the computed squared terms, to generate the square of message A, whereby the square of message A is utilized in cryptographic operations.

7. The system according to claim 6,

wherein instructions for directing the processing unit to compute and sum the non-squaring terms for the message A include:

instructions for directing the processing unit to

initialize (MSD, LSD) = 0 and all terms in R[ ] = 0;

for each integer i with 0 ≤ i < (n/2), perform the following operations:

for each integer j with (i+1) ≤ j < (n-i) do,

set (MSD, LSD) = A[i] x A[j] + MSD + R[j+i]; and

set R[j+i] = LSD;

set count = i;

when (count < 2·i) do,

set (MSD, LSD) = A[j] x A[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase count by 1;

set (MSD,LSD) = A[j] x A[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase j by 1;

set R[j+count] = R[j+count] + MSD;

return register R with set values; and

wherein the computing the squaring terms for the message A and adding the computed squared terms to the terms stored in register R include:

instructions for directing the processing unit to:

initialize (MSD, LSD) = 0;

for each integer i with 0 ≤ i < n, perform the following operations:

set (MSD,LSD) = A[i] x A[i] + MSD;

set (MSD', LSD) = R[i·2]·2 + LSD;

set R[i·2] = LSD;

set (MSD,LSD) = R[i·2 + 1] ·2 + MSD' + MSD; and

set R[i·2 + 1] = LSD;

return values stored in register R as the square of the message A,

where, MSD is a most significant digit of an integer and LSD is a least significant digit of the integer.

8. The system according to claim 7 wherein the instructions for directing the processing unit to compute and sum the non-squaring terms for the message A further comprises: instructions for directing the processing unit to:

determine if a value of R[2·i - 1] in the register R has been set; and

in response to a determination that the value of R[2·i - 1] has been set, complete the remainder of the operations for the value of i using the processing unit and simultaneously increasing the value of i by 1;

whereby in response to the determination that the value of R[2·i - 1] has been set, the another processing unit is configured to execute instructions stored in the non-transitory media to:

for each integer i whereby i < (n/2), perform the following operations:

for each integer j with (i+1) ≤ j < (n-i) do,

set (MSD, LSD) = A[i] x A[j] + MSD + R[j+i]; and

set R[j+i] = LSD;

set count = i;

when (count < 2·i) do,

set (MSD, LSD) = A[j] x a[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase count by 1;

set (MSD,LSD) = A[j] x A[count] + MSD + R[j+count];

set R[j+count] = LSD;

increase j by 1; and

return register R with set values.

9. The system according to claim 6 further comprising:

instructions for directing the processing unit to:

apply a Montgomery modular reduction for a modulus value of m to the square of the message A.

10. The system according to claim 6 further comprising:

instructions for directing the processing unit to:

apply a Barett modular reduction for a modulus value of m to the square of the message A.