

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
7 May 2009 (07.05.2009)

PCT

(10) International Publication Number
WO 2009/055906 A1

- (51) International Patent Classification:
G06F 21/24 (2006.01) **H04L 9/28** (2006.01)
- (21) International Application Number:
PCT/CA2008/001887
- (22) International Filing Date: 29 October 2008 (29.10.2008)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/984,851 2 November 2007 (02.11.2007) US
- (71) Applicant (for all designated States except US): **CERTICOM CORP.** [CA/CA]; 4th Floor, 5520 Explorer Drive, Mississauga, Ontario L4W 5L1 (CA).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **LAMBERT, Robert J.** [CA/CA]; 63 Holm Street, Cambridge, Ontario N3C 3N3 (CA).
- (74) Agents: **SLANEY, Brett J.** et al.; Box 25, Commerce Court West, 199 Bay Street, Toronto, Ontario M5L 1A9 (CA).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- with international search report

[Continued on next page]

(54) Title: SIGNED MONTGOMERY ARITHMETIC

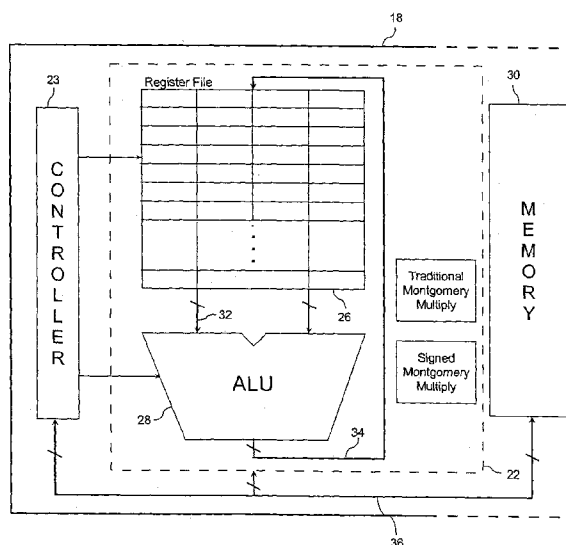


Figure 2

(57) Abstract: A system and method configured for applying Montgomery style reduction directly to negative quantities as well as positive values, producing the new form which does not require conditional operations to move values into the positive range. The low-order components of the resulting product, or partially completed product, can be reduced either by the addition of multiples of the modulus, as is usual in the standard Montgomery multiplication which accepts positive values, or by subtracting multiples of the modulus, which of course depends on the actual computation. Signed versions of the Montgomery values in a Montgomery computation are used to avoid the conditional addition and subtraction that can leak information, for example, using a two's complement representation.

WO 2009/055906 A1



-
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

SIGNED MONTGOMERY ARITHMETIC**TECHNICAL FIELD**

[0001] The following relates generally to Montgomery arithmetic and in particular provides Montgomery arithmetic utilizing signed quantities.

BACKGROUND

[0002] In cryptography, e.g. public key cryptography, operations such as multiplication or exponentiation of integers in some group Z_n may be required, where modular arithmetic is used to operate on the integers. For example, to multiply two numbers modulo some n , the classical approach is to first perform the multiplication and then calculate the remainder. Although the classical approach is simple for basic operations such as in multi-precision calculations and does not require precomputation, the step of calculating the remainder is considered slow. The calculation of the remainder is referred to as reduction in modular arithmetic.

[0003] Modular reduction is often employed in cryptographic applications. Of the well known methods for modular reduction, the one most commonly used is the method of Montgomery modular reduction, referred to as Montgomery reduction in short. One way to avoid the expensive reduction in the classical approach is to use such Montgomery reduction, and thus the computation of modular reduction is often computed this way. Montgomery reduction benefits from the fact that steps of multiplication and shifting are generally faster than division on most computing machines. Montgomery reduction relies on performing certain precomputations and, by doing so, many calculations can be done faster. Also, as opposed to classical methods of reduction-from-above such as Euclidean division, Montgomery reduction reduces from below, that is, the method proceeds by clearing the least-significant portions of the unreduced quantity, leaving the remainder in the upper portion.

[0004] In Montgomery reduction, calculations with respect to a modulus n are carried out with the aid of an auxiliary number R called the Montgomery radix or base. When the modulus is a prime number, a good choice of radix is a power of two, typically chosen as the first convenient power of two larger than the modulus. In the following, the exponent of the power is denoted by L so that $R = 2^L$. The Montgomery reduction of a number a with radix R and prime modulus n is the quantity given by $aR^{-1} \bmod n$. The Montgomery multiplication of

two numbers is the Montgomery reduction of their product, written as $a \otimes b = abR^{-1} \bmod n$. Calculations are carried out on numbers in their Montgomery form. The Montgomery form of a number a is defined as $\hat{a} = aR \bmod n$. Conversion to Montgomery form may be carried out via Montgomery multiplication where $\hat{a} = a \otimes R^2 = aR \bmod n$. Conversion from Montgomery form back to regular form may be carried out by Montgomery reduction: $\hat{a}R^{-1} \bmod n = a \bmod n$, or the Montgomery multiplication: $\hat{a} \otimes 1 = aRR^{-1} = a \bmod n$.

[0005] The Montgomery exponentiation of a number is carried out via the usual square-and-multiply method, substituting Montgomery multiplication for the usual multiplication. As can be appreciated, efficiency may be increased by pre-computing certain fixed values to be used in the calculations. Such values include $\mu = (-n)^{-1} \bmod 2^w$, for some w typically being the bit size of a word (or block) of the value (or perhaps the entire value) being operated on and $R^2 \bmod n$.

[0006] In a given cryptographic system, a computational engine may be used for calculating the Montgomery product of two numbers, sometimes referred to as a Montgomery engine or Montgomery machine. The engine may be implemented in a hardware or software module and operates on a set of parameters to produce a result. For example, the engine may be used to produce the result $a \otimes b$ by inputting a and b . The Montgomery engine can be configured to also convert to and from Montgomery form. For converting to Montgomery form, the engine accepts a and R^2 as inputs and produces an output $\hat{a}$. Conversely, for converting back to normal form, the engine accepts $\hat{a}$ and 1 as inputs and outputs a . The engine may also be configured to calculate the Montgomery reduction of a number. In this case, the engine accepts a and 1 as inputs and produces $aR^{-1} \bmod n$ as an output. To initialize the Montgomery engine, the engine is loaded with a modulus p and a radix R .

[0007] The use of Montgomery reduction to implement Montgomery multiplication is well known. There are many algorithms that can be used to perform the Montgomery multiplication. In one example, the Montgomery multiplication of two m -bit integers a and b in base 2^w , reduced mod an m -bit integer n , where $a = (a_{m-1} \dots a_1 a_0)$, $b = (b_{m-1} \dots b_1 b_0)$, $n = (n_{m-1} \dots n_1 n_0)$, and $0 \leq a, b < n$, produces an output $abR^{-1} \bmod n$. In this example,

1 $R = 2^L$, and $\mu = -n^{-1} \bmod 2^w$ as above. In one exemplary algorithm, Montgomery
 2 multiplication may proceed as follows:

3 [0008] 1. $A \leftarrow 0$, where Z is the result and $Z = (z_m z_{m-1} \dots z_1 z_0)$.

4 [0009] 2. For i from 0 to $(m-1)$ do the following:

5 [0010] 2.1 $u_i \leftarrow (z_0 + a_i b_0) \mu \bmod 2^w$; and

6 [0011] 2.2 $Z \leftarrow (Z + a_i b + u_i n) / 2^w$.

7 [0012] 3. If $Z \geq n$ then $Z \leftarrow Z - n$.

8 [0013] 4. Return (Z).

9 [0014] In the above algorithm, it can be seen that a final comparison against the modulus
 10 and a conditional subtraction is performed, if the result is greater than or equal to the
 11 modulus. It will be appreciated that in performing an EC addition, a conditional addition
 12 may be performed. A side-effect of such a conditional subtraction is that information can be
 13 leaked to a side-channel analysis or attack, which is known to generally involve a process of
 14 exploiting some implementation aspect of a cryptographic algorithm such as sequential
 15 computational operations.

16 [0015] If Montgomery multiplication is to be used in elliptic curve computations, then the
 17 required operations may mix additions and subtractions alongside Montgomery
 18 multiplications and squarings. An example of such an elliptic curve operation is the EC point
 19 doubling formulae, which is defined in the Guide to Elliptic Curve Cryptography (Hankerson,
 20 Menezes, Vanstone) on page 91, Algorithm 3.21. A side-effect of these mixed additions and
 21 subtractions is a non-uniformity of operation to again reduce the results into the range
 22 required for Montgomery multiplication, which can possibly involve conditional additions or
 23 subtractions. For example, adding quantities can require subtraction of the modulus to reduce
 24 the value. Also, the subtractions of quantities can result in negative values. Typically, the
 25 modulus is added to such negative quantities to bring the modular result into the positive
 26 range. All of these conditional additions/subtractions can potentially leak information on the
 27 operands. Even if the operation is always performed, but the result is not always used (so

called 'dummy' operations when not used), information can still potentially leak if the use or non-use can be detected.

[0016] It is therefore an object of the following to inhibit the above-mentioned side-channel attacks during Montgomery operations.

SUMMARY

[0017] It has been recognized that modifying Montgomery arithmetic to utilize signed quantities, allows for the computation of Montgomery arithmetic, in particular Montgomery multiplication, in particular for performing Montgomery multiplication mixed with modular additions and subtractions, to be implemented in a more uniform way and thus inhibit information leakage.

[0018] In one aspect, there is provided a method for performing Montgomery arithmetic in a cryptographic operation comprising representing operands in a Montgomery arithmetic operation as signed values; and operating on the signed values to produce a result for use in the cryptographic operation.

[0019] In another aspect, there is provided a computer readable medium for performing Montgomery arithmetic in a cryptographic operation comprising computer readable instructions for representing operands in a Montgomery arithmetic operation as signed values; and operating on the signed values to produce a result for use in the cryptographic operation.

[0020] In yet another aspect, the arithmetic operation is Montgomery multiplication.

[0021] In yet another aspect, the signed values are obtained by representing said operands in two's complement form.

[0022] In yet another aspect, there is provided a Montgomery machine comprising an arithmetic logic unit configured to operate on signed versions of Montgomery operands; a register file; and a controller, said controller being configured to load said signed versions of said Montgomery operands into said register and operate said arithmetic logic unit on said signed versions to produce a result.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] An embodiment of the invention will now be described by way of example only with reference to the appended drawings wherein:

[0024] Figure 1 is a schematic block diagram of a cryptographic communication system.

[0025] Figure 2 is a schematic block diagram of any one or both of cryptographic modules shown in Figure 1.

DETAILED DESCRIPTION OF THE DRAWINGS

[0026] Referring now to Figure 1, a cryptographic system 10 generally comprises a first correspondent 12 that communicates with a second correspondent 14 over a communication channel 16. Each correspondent includes a cryptographic module 18, which either has its own memory 20 for storing inputs, outputs and intermediate portions of cryptographic operations, or has access to an external memory 20 that is part of the correspondent (12, 14). It can be seen that in the example shown in Figure 1, the first correspondent 12 includes a memory 20 external to the cryptographic module 18 and the second correspondent 14 includes memory 20 that is internal to the cryptographic module 18 to illustrate that the capability to store data can be provided in any suitable arrangement. It will also be appreciated that the memory 20 can be external to and accessible (e.g. via a network connection etc.) to the correspondent 12, 14 (not shown) if necessary or desired.

[0027] The cryptographic module 18 is configured to perform cryptographic operations such as encryption/decryption, signing and modular arithmetic, etc. In this example, the cryptographic module 18 is configured for performing elliptic curve cryptographic (ECC) operations, and includes a block Montgomery machine 22, further detail of which is shown in Figure 2. It will be appreciated that the cryptographic module 18 and any component thereof may be implemented as an apparatus in either hardware or software (computer readable instructions embodied in/on a computer readable medium).

[0028] As can be seen in Figure 2, the cryptographic module 18 includes a controller 24, the block Montgomery machine 22, and a memory 30. The machine 22 generally comprises a register file 26 and an arithmetic logic unit (ALU) 28.

1 **[0029]** The memory 30 may also be included in the machine 22 or be external to the
2 machine 22 as illustrated. It will be appreciated that the memory 30 may comprise all or part
3 of the memory 20 (shown in Figure 1) or may be provided as a separate component in the
4 cryptographic module 18 as shown. The memory 30 may include random access memory
5 (RAM), read only memory (ROM) and/or any other type of suitable memory structure.

6 **[0030]** The register file 26 is a group of general purpose registers, which may be used as
7 intermediate storage for cryptographic operations performed for and by the cryptographic
8 module 18. The register file 26 communicates with the ALU 28 via data input buses 32. The
9 ALU 28 comprises integer arithmetic circuitry. A data output or result bus 34 is provided
10 from the ALU 28 to the register file 26 for writing results of computations performed in the
11 ALU 28 to the register file 26.

12 **[0031]** Computational operations of the ALU 28 are controlled via programmed
13 instructions residing in or accessible to the controller 24. A memory bus 36 is also provided
14 to enable the controller 24 and the machine 22 to utilize memory 30 when performing and
15 outputting results of the cryptographic operations. Typically, the machine 22 is utilized in an
16 existing host computer system and the controller 23 receives control signals from the host
17 system and communicates data to the register file 26 and ALU 28.

18 **[0032]** The general purpose registers that make up the register file 26 are made up of a
19 certain number of blocks, each having a fixed length, e.g. 128 bits. As is well known in the
20 art, the blocks are used to store data (operands, results etc.) during Montgomery
21 computations.

22 **[0033]** It has been recognized that by providing and using a version of Montgomery
23 arithmetic utilizing signed quantities, the computations can be performed in a more uniform
24 way so as to not leak information to a side channel analysis. For example, using signed
25 quantities allows for not only addition, but also subtraction to merely increase the bound of
26 the magnitude of the quantities to be input to the Montgomery multiplication operation.
27 Given bounds on initial quantities, addition or subtraction increases this bound by a single bit,
28 assuming that reduction is not applied, which is important as the Montgomery algorithm is
29 limited in the range of values it can handle by the Montgomery radix R that has been

implemented. The operands will thus become larger as they are added or subtracted (again if reduction does not take place).

[0034] As will be discussed and shown below, the parameters of the Montgomery multiplication can be tailored to suit the bounds on the input operands, resulting in more uniform computations and thus not requiring the leakage prone expedient of conditional or dummy subtraction.

[0035] Firstly, the following illustrates how the requirement for positive inputs in traditional Montgomery multiplication can leak information about the operands.

[0036] The computation being performed in the following examples is $((\hat{a} - \hat{b}) \otimes \hat{c} \bmod n$, where the '^' indicates Montgomery form, $\otimes$ indicates a Montgomery multiplication operation, and n is the modulus. For this computation, the inputs to the Montgomery multiplier would be $(\hat{a} - \hat{b})$ and $\hat{c}$. It will be appreciated that this computation is for illustrative purposes only and that the following principles can be equally applied to other computations. In the following example, for ease of explanation, the modulus $n = 2^{30} - 1$. As outlined above, the Montgomery multiplication of two values a and b is obtained by performing the computation: $a \otimes b = abR^{-1} \bmod n$, where R is the radix, which is typically chosen to be a power of 2, larger than n .

[0037] In this example, R may be chosen to be: $R = 2^{32}$. To simplify the following explanation, hexadecimal notation will be used. As such, $n = 0x3FFFFFFF$ and $R = 0x100000000$, '0x' being the typical computer language representation where the leading zero is used to recognize the value as a number and the 'x' is used to signify hexadecimal.

[0038] In this example: $\hat{a} = 0x2953F462$, $\hat{b} = 0x33AA8952$, and $\hat{c} = 0x1E349902$.

[0039] Given the above values, the result obtained by computing the first part of the computation is: $\hat{a} - \hat{b} = -0xA5694F0$, i.e. a negative value. To input this result to a traditional Montgomery multiplication operation, this value would require an addition to move the input $(\hat{a} - \hat{b})$ into the positive range. Often this is done by adding the modulus n to the negative value, namely where: $\hat{a} - \hat{b} + n = 0x35A96B0F$. It can be seen that in order to condition the inputs for use in traditional Montgomery multiplication, a conditional operation

is performed to determine if the value is negative and, if so, the conditional addition shown above is performed. This conditional addition can be problematic in that the presence of the conditional operation can leak information about the operands. For example, in a side-channel analysis, whenever the conditional addition is performed, the attacker may be able to ascertain that one of the inputs is a negative value, thus leaking information that can be used in an attack.

[0040] Once the conditional addition has been performed, the standard Montgomery multiplication of the two inputs (where $(\hat{a} - \hat{b})$ has been shifted to be positive) can proceed, since the inputs are both positive. The full multiplication (i.e. before reduction) is then:

[0041] $(\hat{a} - \hat{b} + n) * \hat{c} = 0x35A96B0F * 0x1E349902 = 0x654E1082D5ACD1E.$

[0042] The product from the above computation requires the use of a cancelling multiple u , which in this example would be $0xAD5ACD1E$ (see step 2.1 above where u_i represents a digit from the cancelling multiple u).

[0043] The cancelling multiple $u = (u_{m-1} \dots u_1 u_0)$ is then used as follows:

[0044] $(\hat{a} - \hat{b} + n) * \hat{c} + 0xAD5ACD1E * n = 0x31AB944F00000000$ to zero the bottom bits for reduction.

[0045] The value $0x31AB944F00000000$, when shifted down, produces the result:

[0046] $(\hat{a} - \hat{b} + n) \otimes \hat{c} = 0x31AB944F.$

[0047] It is our observation that the cancellation of the low-order components of negative multiples is also a method of reduction, now in absolute value. In other words, it has been observed that Montgomery style reduction can be applied directly to negative quantities as well as positive values, producing the new form which does not require conditional operations to move values into the positive range. The low-order components of the resulting product, or partially completed product, can be reduced either by the addition of multiples of the modulus, as is usual in the standard Montgomery multiplication which accepts positive values, or by subtracting multiples of the modulus, which of course depends on the actual computation.

[0048] From the above observations, it has been recognized that signed versions of the Montgomery values in a Montgomery computation can be used to avoid the conditional addition and subtraction that can leak information. The above example, namely Montgomery multiplication will be used to illustrate how the signed version obtains the same result without the conditional operations although it will be appreciated that the signed Montgomery values can be used in any Montgomery operation performed by the Montgomery machine 22.

[0049] As there are many ways to represent negative integer quantities, e.g. one's compliment, two's complement etc., there are many ways to represent the Montgomery values as signed values. Since the most ubiquitous of the techniques is the two's compliment approach, the following example will illustrate the effectiveness of using signed Montgomery values using a two's complement representation.

[0050] As computed above, the value $\hat{a} - \hat{b} = -0xA5694F0$ is a negative value. If the two's compliment of this value is computed, the value would be: $\hat{a} - \hat{b} = 0xF5A96B10$, where the sign bits have extended the value to 32 bits.

[0051] To illustrate the conversion, the decimal representation can be examined:

[0052] $(\hat{a} - \hat{b})_{Decimal} = 693367906 - 866814290 = -173446384$, which in hexadecimal would be $-A5694F0$ as before. To take the two's complement, first consider the binary representation of value $A5694F0$, which is: 0000 1010 0101 0110 1001 0100 1111 0000, with the four leading zeros indicating it is a negative value.

[0053] To take the two's complement, each bit is inverted and a one (1) added to the inverted value, namely: 1111 0101 1010 1001 0110 1011 0000 1111 + 1 =

[0054] 1111 0101 1010 1001 0110 1011 0001 0000.

[0055] When converting this two's complement value in binary back into hexadecimal, the result is $F5A96B10$ where the F indicates that it is a signed value.

[0056] The two's complement or signed version of $(\hat{a} - \hat{b})$ can be directly input into a signed-capable version of a Montgomery multiplication routine. A signed-capable

Montgomery multiplication is a version wherein the multiplications consider quantities to be signed. That is, the values a , b , Z (not the individual digits) are considered to be signed quantities. The signed capable version does not need to perform the conditional subtraction shown above in the context of the traditional Montgomery multiplication. As a result, the signed-capable Montgomery multiplication operation computes:

[0057] $(\hat{a} - \hat{b}) \otimes \hat{c} = 0xF5A96B10 \otimes 0x1E349902 = 0x31AB944F$, which as shown above in the traditional Montgomery operation, is the correct result.

[0058] The signed-capable Montgomery operation would be implemented by including two's complement architecture in the cryptographic unit 18 so that the controller 23 performs the arithmetic with a two's complement signed representation and then perform a normal Montgomery multiplication on the positive values, which does not require a conditional subtraction at the end. As shown in Figure 2, the machine 22 can store a traditional Montgomery multiply routine but more importantly, has a signed version of Montgomery multiply that does not include the conditional subtraction. The operation would be performed by obtaining the signed versions of the operands, initiate the signed Montgomery multiply routine and use this routine to operate on the signed operands. Again, if the signed versions are in two's complement, the cryptographic unit 18 would include architecture to represent values in the two's complement form.

[0059] As a more realistic example, another computation that could be performed is $(\hat{a} - \hat{b}) \otimes \hat{c} \bmod p$, where $\otimes$ is the Montgomery multiplication, and $p = 2^{224} * (2^{32} - 1) + 2^{192} + 2^{96} - 1$ (i.e. a prime used for the NIST ECC standard).

[0060] Then, if:

[0061] $\hat{a} =$

0xFDF45D5F6C24293B5DB473EF2344D12DA153C6B6FD9DDE10E9E3401F45EAD3A6;

[0062] $\hat{b} = 0xFF2BEC9B6C491FE9BEF159874BCBC8EA3D5F386A750B52D8B781339F66C5765E$;
and

[0063] $\hat{c} = 0x60AC2031CA11CC7B0C7232B22CB9FC9DC011C42C70F5432353F8739D3AC91FAE$

1 [0064] then:

2 [0065] $\hat{a} - \hat{b} = -$

3 0x1378F3C0024F6AE613CE5982886F7BC9C0B71B3776D74C7CD9DF38020DAA2B8;

4 [0066] which would require an addition to move the value to the positive range, similar to
 5 the example above. If this value is instead represented in two's complement, (where the
 6 FF..FF denotes sign extension to the top of the representation), $\hat{a} - \hat{b}$ would be represented
 7 as:

8 [0067] $\hat{a} - \hat{b} = 0xFF...FFEC870C3FFDB09519EC31A67D779084363F48E4C88928B3832620C7FDF255D48$

9 [0068] This result could be directly input into the signed-capable version of Montgomery
 10 multiply mentioned above, i.e. a version lacking any final conditional subtraction.

11 [0069] This results in:

12 [0070] $(\hat{a} - \hat{b}) \otimes \hat{c} = 0x5AEBE0E982B09295A55C725B0AE23F4E415F5EE87CDBAD67CB3F1752026201E2;$

13 which is the correct result.

14 [0071] It can therefore be seen that Montgomery style reduction can be applied directly to
 15 negative quantities as well as positive values, producing the new form which does not require
 16 conditional operations to move values into the positive range. The low-order components of
 17 the resulting product, or partially completed product, can be reduced either by the addition of
 18 multiples of the modulus, as is usual in the standard Montgomery multiplication which
 19 accepts positive values, or by subtracting multiples of the modulus, which of course depends
 20 on the actual computation.

21 [0072] From the above observations, it has been recognized that signed versions of the
 22 Montgomery values in a Montgomery computation can be used to avoid the conditional
 23 addition and subtraction that can leak information, for example, using a two's complement
 24 representation.

25 [0073] Although the invention has been described with reference to certain specific
 26 embodiments, various modifications thereof will be apparent to those skilled in the art

- 1 without departing from the spirit and scope of the invention as outlined in the claims
- 2 appended hereto.

Claims:

1. A method for performing Montgomery arithmetic in a cryptographic operation comprising representing operands in a Montgomery arithmetic operation as signed values; and operating on said signed values to produce a result for use in said cryptographic operation.
2. The method according to claim 1 wherein said arithmetic operation is Montgomery multiplication.
3. The method according to claim 1 wherein said signed values are obtained by representing said operands in two's complement form.
4. The method according to claim 1 wherein said signed values are obtained by representing said operands in one's complement form.
5. The method according to claim 1 comprising executing a signed-capable Montgomery operation stored in a Montgomery machine to instruct said Montgomery machine to operate using said signed values.
6. A computer readable medium for performing Montgomery arithmetic in a cryptographic operation comprising computer readable instructions for performing the method according to any one of claims 1 to 5.
7. A Montgomery machine comprising an arithmetic logic unit configured to operate on signed versions of Montgomery operands; a register file; and a controller, said controller being configured to load said signed versions of said Montgomery operands into said register and operate said arithmetic logic unit on said signed versions to produce a result.
8. The machine according to claim 7 wherein said arithmetic operation is Montgomery multiplication.

9. The machine according to claim 7 wherein said signed values are obtained by representing said operands in two's complement form.
10. The machine according to claim 7 wherein said signed values are obtained by representing said operands in one's complement form.
11. The machine according to claim 7, further comprising a memory storing a signed capable Montgomery operation, said controller being further configured for executing said signed-capable Montgomery operation to instruct said controller to operate using said signed values.
12. The machine according to claim 11, further comprising a traditional Montgomery operation stored in said memory, said controller being further configured for selecting either said signed-capable Montgomery operation or said traditional Montgomery operation according to an instruction.

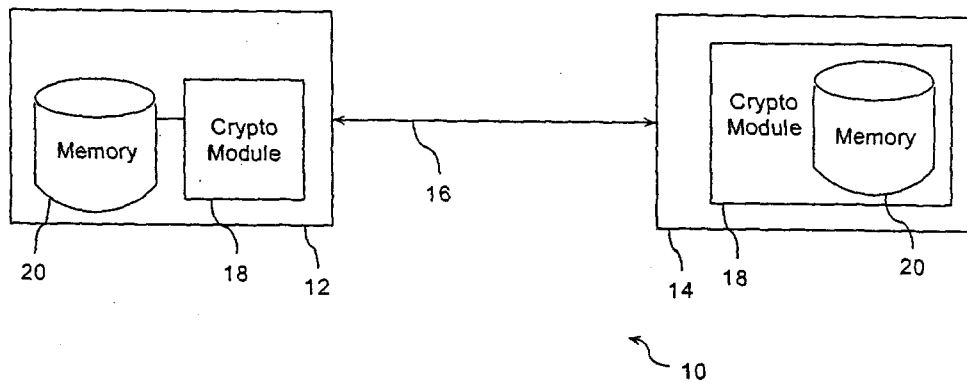


Figure 1

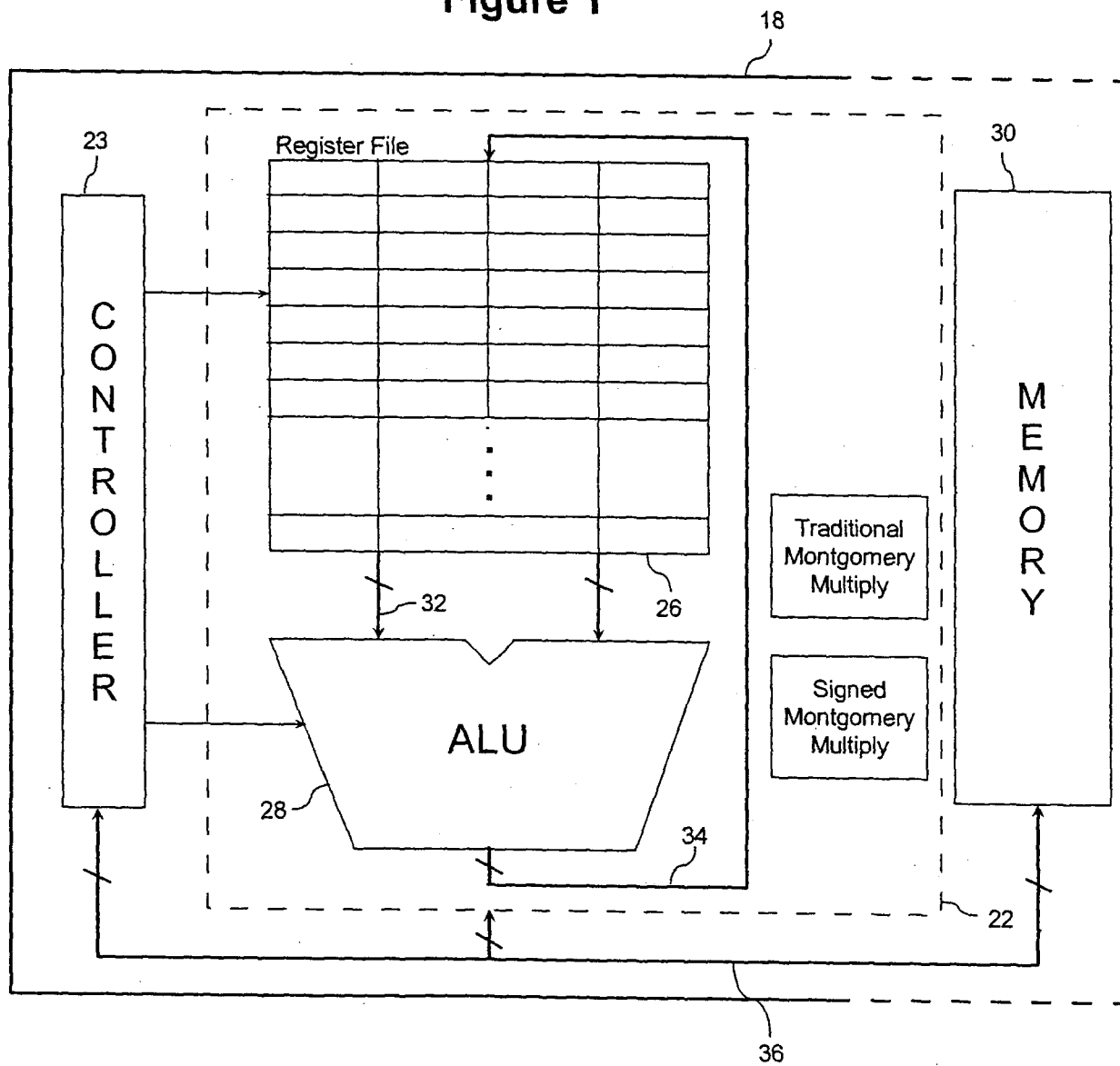


Figure 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2008/001887

A. CLASSIFICATION OF SUBJECT MATTER

IPC: **G06F 21/24** (2006.01) , **H04L 9/28** (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F (2006.01) , **H04L** (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)

Databases: Canadian Patents Database, WEST, Delphion, Google Patents, IEEE XPLORE

Keywords: Montgomery, arithmetic, multiplication, signed, one's, two's, complement, negative, 1's, 2's, elliptic, crypto*, algorithm

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/0198093 A1 (Son) 08 September 2005 (08-09-2005) * par [0005] - [0010], [0069] - [0079], [0119]; Figs 3, 8A - 8C *	1 - 12
X	Jin-Hua Hong; Cheng-Wen Wu, "Radix-4 modular multiplication and exponentiation algorithms for the RSA public-key cryptosystem," Design Automation Conference, 2000. Proceedings of the ASP-DAC 2000. Asia and South Pacific , pp.565 - 570, 2000. * Abstract; Sections 2.2 and 3.2 *	1 - 12
X	G.M. de Dormale; P. Bulens; J.-J. Quisquater, "An improved Montgomery modular inversion targeted for efficient implementation on FPGA," Field-Programmable Technology, 2004. Proceedings. 2004 IEEE International Conference on, pp. 441 - 444, 6-8 Dec. 2004. * Abstract; Section B: Modified Montgomery Modular Inverse Algorithm *	1 - 12
A	Peter L. Montgomery, "Modular Multiplication Without Trial Division", Mathematics of Computation, vol. 44, no. 170, pp. 519 - 521, Apr, 1985. * Whole Document *	

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

12 February 2009 (12-02-2009)

Date of mailing of the international search report

25 February 2009 (25-02-2009)

Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 001-819-953-2476

Authorized officer

Raghid Shreih 819- 994-2694

INTERNATIONAL SEARCH REPORTInternational application No.
PCT/CA2008/001887

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2004/0179681 A1 (Lee et al.) 16 September 2004 (16-09-2004) * Whole Document *	
A	US 7,035,889 B1 (Carlson et al.) 25 April 2006 (25-04-2006) * Whole Document *	
A	US 6,381,699 B2 (Kocher et al.) 30 April 2002 (30-04-2002) * Whole Document *	
A	US 6,480,606 B1 (Kurumatani) 12 November 2002 (12-11-2002) * Whole Document *	
A	US 7,174,015 B1 (Koc et al.) 06 February 2007 (06-02-2007) * Whole Document *	

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2008/001887

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US 2005198093A1	08-09-2005	CN 1702613A	30-11-2005
		DE102005010764A1	29-09-2005
		FR 2867579A1	16-09-2005
		FR 2867579B1	18-04-2008
		JP 2005250481A	15-09-2005
		KR 20050088506A	07-09-2005
US 2004179681A1	16-09-2004	CN 1530824A	22-09-2004
		EP 1457875A2	15-09-2004
		EP 1457875A3	28-06-2006
		JP 2004280103A	07-10-2004
		KR 20040081581A	22-09-2004
		US 2008065713A1	13-03-2008
US 7035889B1	25-04-2006	None	
US 6381699B2	30-04-2002	AT 325478T	15-06-2006
		AT 385089T	15-02-2008
		AU 2557399A	26-07-1999
		AU 5203899A	20-12-1999
		AU 6381699A	10-01-2000
		CA 2316227A1	15-07-1999
		CA 2333095A1	29-12-1999
		CA 2333095C	10-05-2005
		DE 69834431D1	08-06-2006
		DE 69834431T2	19-04-2007
		DE 69938045D1	13-03-2008
		DE 69938045T2	15-01-2009
		EP 1050133A1	08-11-2000
		EP 1050133A4	04-05-2005
		EP 1050133B1	03-05-2006
		EP 1084543A1	21-03-2001
		EP 1084543A4	17-11-2004
		EP 1084543B1	23-01-2008
		EP 1090480A2	11-04-2001
		EP 1090480A4	15-06-2005
		EP 1691501A1	16-08-2006
		EP 1926241A2	28-05-2008
		EP 1933496A2	18-06-2008
		IL 139935A	19-06-2005
		IL 139935D0	10-02-2002
		JP 2002519722T	02-07-2002
		US 6278783B1	21-08-2001
		US 6304658B1	16-10-2001
		US 6327661B1	04-12-2001
		US 2001002486A1	31-05-2001
		US 2001053220A1	20-12-2001
		US 2002124178A1	05-09-2002
		US 2003028771A1	06-02-2003
		US 2006045264A1	02-03-2006
		US 2008022146A1	24-01-2008
		US 2008059826A1	06-03-2008
		US 2008104400A1	01-05-2008
		WO 9935782A1	15-07-1999
		WO 9963696A1	09-12-1999
		WO 9967919A2	29-12-1999
		WO 9967919A3	23-03-2000

INTERNATIONAL SEARCH REPORTInternational application No.
PCT/CA2008/001887

US	6480606B1	12-11-2002	DE	69924912D1	02-06-2005
			DE	69924912T2	30-03-2006
			EP	0952697A2	27-10-1999
			EP	0952697A3	14-05-2003
			EP	0952697B1	27-04-2005
			JP	11242434A	07-09-1999

US	7174015B1	06-02-2007	None
----	-----------	------------	------