Title: METHOD AND SYSTEM FOR ATOMICITY FOR ELLIPTIC CURVE CRYPTO SYSTEMS

FIG. 2A

<table>
<thead>
<tr>
<th>HPAt elliptic curve doubling</th>
<th>Input: P = (R_x, R_y, R_z) in Jacobian coordinates</th>
<th>Output: 2P = (A, A') in Jacobian coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( R_x )</td>
<td>( A )</td>
</tr>
<tr>
<td>( \Gamma_1 )</td>
<td>( R_x + A' )</td>
<td>( R_y + A' )</td>
</tr>
<tr>
<td>( \Gamma_2 )</td>
<td>( R_x + R_y + A' )</td>
<td>( -R_z - A' )</td>
</tr>
<tr>
<td>( \Gamma_3 )</td>
<td>( R_x + A' )</td>
<td>( R_y - A' )</td>
</tr>
</tbody>
</table>

Abstract: A method and system are provided for atomicity for elliptic curve cryptosystems (ECC systems). The method includes a side channel atomic scalar multiplication algorithm using mixed coordinates. The algorithm including repeating a sequence of field operations for each elliptic curve addition or doubling operation to provide an atomic block, wherein an atomic block appears equivalent by side-channel analysis. The mixed coordinates are chosen based on a ratio of I/M where I and M are the time required to execute an inversion and a multiplication in the ground field respectively. If the I/M ratio is less than 60, a mixture of affine and Jacobian coordinates are used during scalar multiplication. If the I/M ratio is 60 or more, a mixture of Chudnovsky-Jacobian and Jacobian coordinates are used during scalar multiplication. The method is optimized for elliptic curves over \( F_p \) defined by an equation of the form \( y^2 = x^3 + ax + b \), where \( a, b \in F_p \), having \( a = -3 \).
before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments
METHOD AND SYSTEM FOR ATOMICITY FOR ELLIPTIC CURVE CRYPTOSYSTEMS

This invention relates to the field of encryption. In particular, it relates to atomicity for cryptosystems based on elliptic curves over finite fields (ECC-systems).

An elliptic curve is a plane curve defined by an equation of the form

\[ y^2 = x^3 + ax + b. \]

The set of points on such a curve (i.e., all solutions of the equation together with a point at infinity) can be shown to form an abelian group with the point at infinity as the identity element. If the coordinates \(x\) and \(y\) are chosen from a finite field, the solutions form a finite abelian group. The discrete logarithm problem on such elliptic curve groups is believed to be more difficult than the corresponding problem in the multiplicative group of non-zero elements of the underlying finite field. Thus keys in elliptic curve cryptography can be chosen to be much shorter than RSA algorithm keys for a comparable level of security.

Due to the nature of the algorithms used in ECC-system, these systems are vulnerable to so-called side channel attacks which are based on analysis of the power consumption in the system. The most commonly used method is simple power analysis (SPA) in which an attacker needs only a single power trace from an execution of the scalar multiplication algorithm in the ECC-system to deduce the secret key. This is possible as the scalar multiplication algorithm processes the bits in the secret key one by one and consumes different amounts of power depending on whether the value of a given bit is 0 or 1.

An attacker analyzing the power consumption in an ECC-system will, therefore, see a trace as shown in Figure 1A. From the trace 110, the attacker can deduce that the bits of the secret key processed were 0001.

There exist various countermeasures against SPA, but all of these reduce the performance of the system to some extent. This is a major problem as performance is usually a major issue in cryptographic applications.
Among the known countermeasures, the method known as side channel atomicity results in the least performance penalty. Side channel atomicity was originally described in B. Chevallier-Mames, M. Ciet and M. Joye, "Low-Cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity" in IEEE Transactions on Computers, volume 53, 2004 (referred to herein as Chevallier-Mames et al). The basic idea of the countermeasure is to rewrite the scalar multiplication algorithm to make it appear as if it consisted of a sequence of identical blocks ("atoms") with indistinguishable power traces. With side channel atomicity enabled an attacker will see a trace 120 as in Figure 1B. No information about the secret key can be deduced from this trace 120.

There are, however, two serious problems with the known solution of side channel atomicity.

1. The solution does not apply to high performance ECC-systems which make use of mixed coordinate representations of the curves being used. The solution only applies to situations where a fixed coordinate representation is used. As the use of mixed coordinate representations is necessary in highly optimized ECC-systems, this makes the solution of side channel atomicity less attractive.

2. The solution does not make use of the optimizations available for the curves recommended by NIST (see National Institute for Standards and Technology, "FIPS PUB 186-2", http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf). This means that the solution introduces a higher performance penalty than necessary for the NIST curves. As these curves are described in open standards and are selected by the NSA for the Suite B algorithms (see National Security Agency, "Fact Sheet NSA Suite B Cryptography", http://www.nsa.gov/ia/industry/crypto_suite_b.cfm), this is a major drawback for the solution of side channel atomicity.

It is an aim of the present invention to provide a solution to atomicity of ECC-systems which, as well from incorporating all the advantages from the side channel atomic solution, it must support the use of mixed coordinates. A further aim is to support the use of optimizations available for the NIST curves.
According to a first aspect of the present invention there is provided a method of providing
atomicity for cryptosystems based on elliptic curves, comprising: a side channel atomic
scalar multiplication algorithm using mixed coordinates, including: repeating a sequence of
field operations for each elliptic curve addition or doubling operation to provide an atomic
block, wherein an atomic block appears equivalent by side-channel analysis; choosing the
coordinates based on a ratio of I/M where I and M are the time required to execute an
inversion and a multiplication in the ground field respectively.

The side channel atomic scalar multiplication algorithm is preferably optimized for elliptic
curves over $\mathbb{F}_p$, defined by an equation of the form $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_p$, having $a = -3$.

If the I/M ratio is less than 60, a mixture of affine and Jacobian coordinates are preferably
used during scalar multiplication. If the I/M ratio is 60 or more, a mixture of Chudnovsky-
Jacobian and Jacobian coordinates are preferably used during scalar multiplication.

The sequence of field operations may start with a multiplication operation and may be a
multiplication, an addition, a negation, and an addition.

According to a second aspect of the present invention there is provided an elliptic curve
cryptography system (ECC-system) providing atomicity, comprising: means for encryption
including means for applying a side channel atomic scalar multiplication using mixed
coordinates, including: means for repeating a sequence of field operations for each elliptic
curve addition or doubling operation to provide an atomic block, wherein an atomic block
appears equivalent by side-channel analysis; and means for choosing the coordinates based
on a ratio of I/M where I and M are the time required to execute an inversion and a
multiplication in the ground field respectively; and means for transferring encrypted data,
wherein the encrypted data is protected from side-channel analysis.

According to a third aspect of the present invention there is provided a computer program
product stored on a computer readable storage medium, comprising computer readable
program code means for performing the steps of: a side channel atomic scalar multiplication
algorithm using mixed coordinates, including: repeating a sequence of field operations for each elliptic curve addition or doubling operation to provide an atomic block, wherein an atomic block appears equivalent by side-channel analysis; and choosing the coordinates based on a ratio of I/M where I and M are the time required to execute an inversion and a multiplication in the ground field respectively.

Embodiments of the present invention will now be described, by way of examples only, with reference to the accompanying drawings in which:

Figure 1A is a curve showing a schematic power trace from an unprotected ECC-system as known in the art;

Figure 1B is a curve showing a schematic power trace from an ECC-system with side channel atomicity enabled as known in the art;

Figures 2A to 2C are tables showing elliptic curve doubling in Jacobian coordinates, elliptic curve addition in mixed affine/Jacobian coordinates, and elliptic curve addition in mixed Chudnovsky-Jacobian/Jacobian coordinates, all in accordance with the present invention;

Figures 3A and 3B are matrices showing scalar multiplication on the NIST elliptic curves assuming a high I/M ratio in Figure 3A, and assuming a low I/M ratio in Figure 3B, both in accordance with the present invention;

Figures 4A and 4B are schematic flow diagrams of the steps of doubling and addition in accordance with the present invention;

Figure 5 is a schematic flow diagram of the steps a method in accordance with the present invention; and

Figure 6 is a block diagram of a computer system in which the present invention may be implemented.

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. The most time-consuming operation performed in an ECC-system is the so-called scalar multiplication. When implementing an ECC-system, two important choices must be made. The first is which method to use for scalar multiplication. The second is which coordinate representation to use for the elliptic curve being used. The optimal coordinate representation depends on the specific
computational environment (processor power, memory, available software, etc.). The efficiency of the system depends heavily on these choices.

A scalar multiplication algorithm in an ECC-system is, basically, a sequence of elliptic curve additions (ECADD) and doublings (ECDBL).

The described method is constituted by atomic specifications of ECADD and ECDBL which take mixed coordinates and curve type into account. A detailed introduction to scalar multiplication in ECC-systems can be found in L. Elmegaard-Fessel, “Efficient Scalar Multiplication and Security against Power Analysis in Cryptosystems based on the NIST Elliptic Curves Over Prime Fields”, 2006, available online at http://eprint.iacr.org/2006/313.pdf (referred to herein as Elmegaard-Fessel).

Known methods of countermeasures against side channel analysis use scalar multiplication algorithms with uniform behaviour or alter the scalar multiplication scheme to ensure a fixed pattern of ECDBL and ECADD. This means that the scalar multiplication is made atomic with the “atoms” as ECADD and ECDBL.

There is a lot to gain by moving the atomicity approach to the level of field operations. For example, as described in Chevallier-Mames et al. This approach is adapted in the present method to apply to mixed coordinate representations and to apply to the NIST curves.

A process is defined as a sequence of instructions operating on the elements of $\mathbb{F}_p$, e.g. addition or multiplication in $\mathbb{F}_p$. Sequences $I_i$ and $I_j$ of instructions are said to be side channel equivalent and can be written $I_i \sim I_j$, if $I_i$ and $I_j$ cannot be distinguished through observation of their individual power consumption traces.

Definition of side channel atomicity. Let $\Pi = \{\Pi_0, \ldots, \Pi_n\}$ be a set of processes. A common side channel atomic block $\Gamma$ for $\Pi$ is a sequence of instructions such that for all $i \in \{0, \ldots, n\}$ there exists a set of sequences of instructions $\{\gamma_{i1}, \ldots, \gamma_{in}\}$ with $\gamma_{ij} \sim \Gamma$ for all $j = 1, \ldots, si$ and
$$\Pi_t = \gamma_{i1} \parallel \gamma_{i2} \parallel \ldots \parallel \gamma_{is}$$

The instruction sequences $\gamma_j$ are called side channel atomic blocks.

A common side channel atomic block $\Gamma$ always exists for any given $\{\Pi_0, \ldots, \Pi_n\}$. To see this, it should be noticed that dummy field operations can be added to the individual processes $\Pi_0, \ldots, \Pi_n$ in order to make them side channel equivalent. A more difficult task is to find a $\Gamma$ which is minimal with respect to the ordering which results from first considering the number of field multiplications, second the number of additions and third the number of negations.

Example 1. To illustrate the idea of side channel atomicity, consider the double and add algorithm in the setting of a general multiplicatively written group $G$:

Algorithm 1: Square and multiply.
-----------------------------------------------
Input: $x \in G$ and $k = (k_1, \ldots, k_0)_2$
Output: $x^k \in G$.
1. $R \leftarrow x$; $i \leftarrow l - 2$;
2. while $i \geq 0$ do
3.  $R \leftarrow R^2$;
4.  if $k_i = 1$ then
5.    $R \leftarrow R \times x$;
6.  end if
7.  $i \leftarrow i - 1$;
8. end while
9. return $R$
-----------------------------------------------

Algorithm 1 is obviously vulnerable to SPA, as the multiplication in line 5 is executed if, and only if, $k_i = 1$. To secure the algorithm, the aim is to identify a common side channel atomic block $\Gamma$ such that Algorithm 1 can be written as a sequence of repetitions of $\Gamma$. The
naïve choice for $\Gamma$ is a block consisting of a squaring and a multiplication. This results in the square and multiply always algorithm shown in Algorithm 2. This algorithm requires $2(l-1)$ group multiplications.

Algorithm 2: Square and multiply always.

-----------------------------------------------

Input: $g \in G$ and $k = (k_{l-1} \ldots k_0)_2$

Output: $g^k \in G$.

1. $R_0 \leftarrow g$; $i \leftarrow l - 2$;
2. while $i \geq 0$ do
3.  $R_0 \leftarrow R_0^2$;
4.  $R_{i+1} \leftarrow R_{i+1} \cdot g$;
5.  $i \leftarrow i - 1$;
6. end while
7. return $R_0$

-----------------------------------------------

Instead, assuming that a squaring can be performed by executing a multiplication and that the instructions $R \leftarrow R \cdot R$ and $R \leftarrow R \cdot x$ are side channel equivalent, one can let $\Gamma$ consist of:

1) A multiplication (which might be a squaring).
2) An XOR of two bits.
3) An integer substitution.

This results in the SPA resistant method shown in Algorithm 3.

Algorithm 3: SPA resistant square and multiply.

-----------------------------------------------

Input: $x \in G$ and $k = (k_{l-1} \ldots k_0)_2$

Output: $x^k \in G$.

1. $R_0 \leftarrow x$; $R_1 \leftarrow x$; $i \leftarrow l - 2$; $s \leftarrow 0$;
2. while $i \geq 0$ do
3. \( R_0 \leftarrow R_0 \cdot R_s \);
4. \( s \leftarrow s \oplus k_i; \) //XOR of two bits;
5. \( i \leftarrow i \leftarrow \neg(s); \)
6. end while
7. return \( R_0 \)

Algorithm 3 requires only \( 3(l - 1)/2 \) group multiplications on average, the same as the unprotected version (under the assumption that \( M = S \)).

In general, finding a common side channel atomic block \( \Gamma \) for \( \Pi = \{\Pi_0, \ldots, \Pi_n\} \) works as follows: Assume the existence of five disjoint classes of side channel equivalent operations:

1) Modular multiplication of two registers \( R_i \) and \( R_j \) where possibly \( i = j \).
2) Modular addition and subtraction of two registers.
3) Arithmetical and logical operations performed by the arithmetic logic unit (ALU) of the CPU.
4) Equality testing of two CPU registers.
5) Loading/storing of values.

If the hardware does not correspond to these assumptions, software implementation of one or more instructions can be used to emulate a suitable environment.

One first identifies a common side channel atomic block \( \Gamma \) for \( \Pi \). Next, each process \( \Pi_i \) is written as:

\[ \Pi_i = \gamma_{i1} \parallel \gamma_{i2} \parallel \cdots \parallel \gamma_{in} \]

with \( \gamma_{i0} \sim \Gamma \) for all \( i \). The naïve way of constructing an SPA resistant algorithm for executing \( \Pi \) requires a time of:

\[ (n + 1) \cdot \max (s_i) \cdot t(\Gamma), \text{ where } 0 \leq i \leq n. \]

where \( t(\Gamma) \) is the number of field multiplications in \( \Gamma \). The goal is to find instead a \( \Gamma \) satisfying:

\[ t(\Pi_i) = s_i \cdot t(\Gamma), \text{ } i = 0, \ldots, n. \]
Using the notion of side channel atomicity, a version of an algorithm for width-w NAF scalar multiplication can be constructed with mixed coordinates which is secure against SPA.

The choice of coordinate representation used in an ECC-system depends on the ratio I/M where I and M are the time required to execute an inversion and a multiplication in the ground field respectively. When using the curves recommended by NIST there are two cases, as follows.

The I/M ratio is “low”, for example, under 60. In this situation one uses a mixture of affine and Jacobian coordinates during scalar multiplication. More specifically, precomputations are done in affine representation while intermediate doublings are done in Jacobian representation. This implies ECADD in mixed affine/Jacobian coordinates and ECDBL in fixed Jacobian coordinates.

The I/M ratio is “high”, for example, 60 or over. In this situation one uses a mixture of Chudnovsky-Jacobian and Jacobian coordinates during scalar multiplication. More specifically, precomputations are done in Chudnovsky-Jacobian representation while intermediate doublings are done in Jacobian representation. This implies ECADD in mixed Chudnovsky-Jacobian/Jacobian coordinates and ECDBL in fixed Jacobian coordinates.

Both situations require ECDBL in fixed Jacobian representation. This ECDBL should be atomic and take advantage of the optimizations available for the NIST curves having the coefficient a equivalent to -3 (see Elmegaard-Fessel, Section 2.2). It is known that ECDBL in Jacobian coordinates on the NIST curves can be carried out using four multiplications and four squarings in the ground field. Figure 2A shows the specification 210 of the ECDBL of the described method. The version of ECDBL in accordance with the described method does not introduce additional multiplications or squarings.

An * in Figures 2A, 2B and 2C denotes an operation of the correct type which does not alter the values of any register depended upon by the algorithm.
In the case of a “low” I/M ratio, addition takes place in mixed affine/Jacobian coordinates. An ECADD on the NIST curves using these coordinate representations can be carried out using eight multiplications and three squarings in the ground field. Figure 2B shows the specification 220 of the described method’s ECADD in affine/Jacobian coordinates on the NIST curves. The specification does not introduce any additional multiplications or squarings.

In case of a “high” I/M ratio, addition takes place in mixed Chudnovsky-Jacobian/Jacobian representation. An ECADD on the NIST curves using these coordinate representations can be carried out using 11 multiplications and 3 squarings in the ground field. Figure 2C shows the specification 230 of described method’s ECADD in Chudnovsky-Jacobian/Jacobian coordinates on the NIST curves. The specification does not introduce any additional multiplications or squarings.

Notice that both the described method’s ECDBL and ECADD operations are constructed from repetitions of the following sequence: multiplication, addition, negation, addition (all performed in the ground field). It is assumed to be possible to use all possible permutations of M,A,N,A, i.e. MAAN, MANA, MNAA, NAAM, NAMA, NMAA, AANM, AAMN, AMAN, ANAM, AMNA, ANMA. However, several of these may prove to be inefficient.

Notice that the requirements of both atomic ECDBL and atomic ECADD are unchanged from the original formulas, i.e. a requirement of 8M and 11M for atomic ECDBL and ECADD respectively. This is because no dummy field multiplications are used, so only a negligible overhead is introduced by splitting the operations into side channel atomic blocks.

The secure version of the NAF_w (width-w non-adjacent form) scalar multiplication algorithm assuming ‘low’ I/M ratio is as follows, where δ is defined as δ(k_i) = \begin{cases} 1 & k_i \neq 0 \\ 0 & k_i = 0 \end{cases} and A=[a_{ij}] is a 19x10 matrix.

Algorithm 4: Side channel atomic NAF_w scalar multiplication.
Input: \( P \in E(F_p), \quad w > 1 \) and \( k = (d_1...d_0)_{NAF_w} \)

Output: \([k]P \in E(F_p)\),

1. Compute odd multiples \( \{3P, ..., [2^{w-1}]P\} \).
2. \((R_1, R_2, R_3) \leftarrow [d_0]P; [R_3] = 1.\)
3. \(i \leftarrow l - 1; s \leftarrow l; m \leftarrow 0;\)
4. while \( i \geq 0 \) do
5. \((R_{10}, R_{11}) \leftarrow [d_1]P; [0]P \) is never used and can be set to any affine point.
6. \(m \leftarrow -s \cdot (m + 1);\)
7. \(s \leftarrow \delta(d_i) \cdot \left\lfloor \frac{m}{18} \right\rfloor + \delta(d_i) \cdot \left\lfloor \frac{m}{7} \right\rfloor\)
8. \(R_{1m0} \leftarrow R_{1m1} \times R_{1m2}\)
9. \(R_{1m5} \leftarrow R_{1m4} + R_{1m5}\)
10. \(R_{1m6} \leftarrow -R_{1m6}\)
11. \(R_{1m7} \leftarrow R_{1m6} + R_{1m9};\)
12. \(i \leftarrow i - s\)
13. end while
14. return \((R_1, R_2, R_3)\)

A full version of the scalar multiplication algorithm on the NIST elliptic curves assuming
“low” I/M ratio is specified by the matrix shown in Figure 3A. Scalar multiplication on the
NIST elliptic curves assuming “high” I/M ratio is specified by the matrix shown in Figure 3B.

The described method is easily implemented in any optimized ECC-system which uses the
NIST curves (e.g. systems which support the algorithms in Suite B). In practice, the
implementation is done by keeping a suitable matrix \( A = \{a_{mn}\} \) (either the one in Figure 3A or
the one in Figure 3B) in memory. The i’th row of A is used to specify the following
operations in the ground field:

\[
\begin{align*}
R_{ai0} & \leftarrow R_{ai1} \times R_{ai2} \\
R_{ai3} & \leftarrow R_{ai4} + R_{ai5} \\
R_{ai6} & \leftarrow -R_{ai6}
\end{align*}
\]
\[ R_{a17} \leftarrow R_{a18} + R_{a19} \]

Each row of A corresponds to an atomic block which is side channel equivalent to \( \Gamma \). The first 8 rows determine the structure of a doubling and the last 11 rows determine the structure of an addition. As the iterations of the main loop are side channel equivalent, the algorithm will expose a structure of:

\[ \ldots \Gamma \Gamma \Gamma \Gamma \ldots \]

which exposes no information about the scalar.

The scalar multiplication resulting from using the described method is atomic as an attacker only sees a trace consisting of repetitions of power traces from the sequence “M A N A” (multiplication, addition, negation, addition). Additionally, it applies to real-life high performance ECC-systems which use the NIST elliptic curves and mixed coordinate representations.

Referring to Figures 4A and 4B, schematic flow diagrams 410, 420 show the atomic steps in doubling and addition sequences in accordance with the described atomicity method. Figure 4A shows the 8 atomic steps 412 applied to a point P 414 on an elliptic curve to double the point to obtain \([2]P \) 416. Each atomic step 412 is made up of the operations M, A, N, A.

Figure 4B shows the 11 or 14 atomic steps 422 applied to the points Q, R 424 on an elliptic curve to add the points to obtain \( Q \oplus R \) 426. Again, each atomic step 422 is made up of the operations M, A, N, A.

Referring to Figure 5, a schematic flow diagram 500 shows the scalar multiplication 510 of a point P on an elliptic curve, by the scalar k, using the atomicity matrix M of the described method.

The multiplication 510 uses the operations of doubling 520 and addition 530. A point \( P' \) 521 can be doubled 520 using the atomicity matrix M to result in \([2]P' \) 522. The points Q, R 531 can be added 530 using the atomicity matrix M to result in \( Q \oplus R \) 532.
The results of the doublings 520 and additions 530 are used in the multiplication 510 to obtain the scalar multiplication result \([k]P\) 511.

A cryptographic system is provided which includes a means for encryption including the means for applying the side channel atomic scalar multiplication as described. The cryptographic system including means for transferring encrypted data, wherein the encrypted data is protected from side-channel analysis.

Referring to Figure 6, an exemplary system for implementing the described method of ECC-system atomicity includes a data processing system 600 suitable for storing and/or executing program code including at least one processor 601 coupled directly or indirectly to memory elements through a bus system 603. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

The memory elements may include system memory 602 in the form of read only memory (ROM) 604 and random access memory (RAM) 605. A basic input/output system (BIOS) 606 may be stored in ROM 604. System software 607 may be stored in RAM 605 including operating system software 608. Software applications 610 may also be stored in RAM 605.

The system 600 may also include a primary storage means 611 such as a magnetic hard disk drive and secondary storage means 612 such as a magnetic disc drive and an optical disc drive. The drives and their associated computer-readable media provide non-volatile storage of computer-executable instructions, data structures, program modules and other data for the system 600. Software applications may be stored on the primary and secondary storage means 611, 612 as well as the system memory 602.

The computing system 600 may operate in a networked environment using logical connections to one or more remote computers via a network adapter 616.

Input/output devices 613 can be coupled to the system either directly or through intervening
I/O controllers. A user may enter commands and information into the system 600 through input devices such as a keyboard, pointing device, or other input devices (for example, microphone, joy stick, game pad, satellite dish, scanner, or the like). Output devices may include speakers, printers, etc. A display device 614 is also connected to system bus 603 via an interface, such as video adapter 615.

The invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In a preferred embodiment, the invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

The invention can take the form of a computer program product accessible from a computer-readable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer usable or computer readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus or device.

The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk read only memory (CD-ROM), compact disk read/write (CD-R/W), and DVD.

Improvements and modifications can be made to the foregoing without departing from the scope of the present invention.
CLAIMS

1. A method of providing atomicity for cryptosystems based on elliptic curves, comprising:
   a side channel atomic scalar multiplication algorithm (510) using mixed coordinates, including:
   repeating a sequence of field operations (412, 422) for each elliptic curve addition or doubling operation to provide an atomic block, wherein an atomic block appears equivalent by side-channel analysis;
   choosing the coordinates based on a ratio of I/M where I and M are the time required to execute an inversion and a multiplication in the ground field respectively.

2. A method as claimed in claim 1, wherein the side channel atomic scalar multiplication algorithm (510) is optimized for elliptic curves over $\mathbb{F}_p$ defined by an equation of the form $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_p$, having $a=-3$.

3. A method as claimed in claim 1 or claim 2, wherein if the I/M ratio is less than 60, a mixture of affine and Jacobian coordinates are used during scalar multiplication (220).

4. A method as claimed in claim 1 or claim 2, wherein if the I/M ratio is 60 or more, a mixture of Chudnovsky-Jacobian and Jacobian coordinates are used during scalar multiplication (230).

5. A method as claimed in any one of the preceding claims, wherein the sequence of field operations (412, 422) starts with a multiplication operation.

6. A method as claimed in claim 5, wherein the sequence of field operations (412, 422) is a multiplication, an addition, a negation, and an addition.

7. An elliptic curve cryptography system (ECC-system) providing atomicity, comprising:
means for encryption including means for applying a side channel atomic scalar
multiplication (510) using mixed coordinates, including:
means for repeating a sequence of field operations (412, 422) for each elliptic curve
addition or doubling operation to provide an atomic block, wherein an atomic block appears
equivalent by side-channel analysis; and
means for choosing the coordinates based on a ratio of I/M where I and M are the
time required to execute an inversion and a multiplication in the ground field respectively;
and
means for transferring encrypted data, wherein the encrypted data is protected from
side-channel analysis.

8. An ECC-system as claimed in claim 7, wherein the means for encryption is
optimized for elliptic curves over $F_p$ defined by an equation of the form $y^2 = x^3 + ax + b$, where
$a, b \in F_p$, having $a=-3$.

9. An ECC-system as claimed in claim 7 or claim 8, wherein if the I/M ratio is less than
60, a mixture of affine and Jacobian coordinates are used during scalar multiplication (220).

10. An ECC-system as claimed in claim 7 or claim 8, wherein if the I/M ratio is 60 or
more, a mixture of Chudnovsky-Jacobian and Jacobian coordinates are used during scalar
multiplication (230).

11. An ECC-system as claimed in any one of claims 7 to 11, wherein the means for
repeating a sequence of field operations (412, 422), includes a sequence starting with a
multiplication operation.

12. An ECC-system as claimed in claim 11, wherein the means for repeating a sequence
of field operations (412, 422), includes a sequence of a multiplication, an addition, a
negation, and an addition.

13. A computer program product stored on a computer readable storage medium,
comprising computer readable program code means for performing the steps of:
a side channel atomic scalar multiplication algorithm (510) using mixed coordinates, including:

repeating a sequence of field operations (412, 422) for each elliptic curve addition or doubling operation to provide an atomic block, wherein an atomic block appears equivalent by side-channel analysis; and

choosing the coordinates based on a ratio of I/M where I and M are the time required to execute an inversion and a multiplication in the ground field respectively.
**FIG. 2A**

**HPAS elliptic curve doubling.**

Input: \( P = (R_1, R_2, R_3) \) in Jacobian coordinates.

Output: \( 2P = (R_1', R_2', R_3') \) in Jacobian coordinates.

<table>
<thead>
<tr>
<th>( \Gamma )</th>
<th>( M )</th>
<th>( A_1 )</th>
<th>( N )</th>
<th>( A_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Gamma_1 )</td>
<td>( R_1 - R_3 \times R_3 )</td>
<td>( {2} )</td>
<td>( R_2 - R_1 + R_3 )</td>
<td>( {X_1 + Z_1} )</td>
</tr>
<tr>
<td>( \Gamma_2 )</td>
<td>( R_2 - R_2 \times R_2 )</td>
<td>( {X_1 + Z_2} )</td>
<td>( R_3 - R_1 + R_3 )</td>
<td>( {X_2 + Z_1} )</td>
</tr>
<tr>
<td>( \Gamma_3 )</td>
<td>( R_3 - R_3 \times R_3 )</td>
<td>( {X_2 + Z_2} )</td>
<td>( R_4 - R_1 + R_3 )</td>
<td>( {X_2 + Z_2} )</td>
</tr>
<tr>
<td>( \Gamma_4 )</td>
<td>( R_4 - R_2 \times R_2 )</td>
<td>( {X_2 + Z_2} )</td>
<td>( R_5 - R_1 + R_4 )</td>
<td>( {X_2 + Z_2} )</td>
</tr>
<tr>
<td>( \Gamma_5 )</td>
<td>( R_5 - R_3 \times R_3 )</td>
<td>( {X_2 + Z_2} )</td>
<td>( R_6 - R_1 + R_5 )</td>
<td>( {X_2 + Z_2} )</td>
</tr>
<tr>
<td>( \Gamma_6 )</td>
<td>( R_6 - R_4 \times R_4 )</td>
<td>( {X_2 + Z_2} )</td>
<td>( R_7 - R_1 + R_6 )</td>
<td>( {X_2 + Z_2} )</td>
</tr>
<tr>
<td>( \Gamma_7 )</td>
<td>( R_7 - R_3 \times R_3 )</td>
<td>( {X_2 + Z_2} )</td>
<td>( R_8 - R_1 + R_7 )</td>
<td>( {X_2 + Z_2} )</td>
</tr>
<tr>
<td>( \Gamma_8 )</td>
<td>( R_8 - R_4 \times R_4 )</td>
<td>( {X_2 + Z_2} )</td>
<td>( R_9 - R_1 + R_8 )</td>
<td>( {X_2 + Z_2} )</td>
</tr>
</tbody>
</table>
**HPAS elliptic curve addition.**

**Input:** \( P = (R_1, R_2, R_3) \) in Jacobian coordinates and \( Q = (R_6, R_11) \) in affine coordinates.

**Output:** \( P \oplus Q = (R_1, R_3, R_5) \) in Jacobian coordinates.

<table>
<thead>
<tr>
<th>( M )</th>
<th>( A )</th>
<th>( N )</th>
<th>( A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Gamma_1 )</td>
<td>( R_3 \leftarrow R_3 \times R_2 ) [27]</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>( \Gamma_2 )</td>
<td>( R_3 \leftarrow R_3 \times R_4 ) [4]</td>
<td>*</td>
<td>( R_3 \leftarrow R_3 ) [28]</td>
</tr>
<tr>
<td>( \Gamma_3 )</td>
<td>( R_5 \leftarrow R_5 \times R_3 ) [9 \times 25]</td>
<td>( R_5 \leftarrow R_5 + R_6 ) [2]</td>
<td>( R_5 \leftarrow R_5 ) [3]</td>
</tr>
<tr>
<td>( \Gamma_4 )</td>
<td>( R_6 \leftarrow R_6 \times R_4 ) [3]</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>( \Gamma_5 )</td>
<td>( R_3 \leftarrow R_3 \times R_8 ) [2 \times 25]</td>
<td>*</td>
<td>( R_3 \leftarrow R_3 ) [24]</td>
</tr>
<tr>
<td>( \Gamma_6 )</td>
<td>( R_6 \leftarrow R_6 \times R_7 ) [25]</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>( \Gamma_7 )</td>
<td>( R_5 \leftarrow R_5 \times R_6 ) [1 \times [2 \times 25]]</td>
<td>( R_5 \leftarrow R_5 + R_6 ) [2 \times 25]</td>
<td>( R_5 \leftarrow R_5 ) [3]</td>
</tr>
<tr>
<td>( \Gamma_8 )</td>
<td>( R_3 \leftarrow R_3 \times R_4 ) [25]</td>
<td>( R_3 \leftarrow R_3 + R_6 ) [2 \times 25]</td>
<td>*</td>
</tr>
<tr>
<td>( \Gamma_9 )</td>
<td>( R_3 \leftarrow R_3 \times R_4 ) [25]</td>
<td>( R_3 \leftarrow R_3 + R_5 ) [25]</td>
<td>( R_3 \leftarrow R_4 ) [2]</td>
</tr>
<tr>
<td>( \Gamma_{10} )</td>
<td>( R_3 \leftarrow R_3 \times R_4 ) [25]</td>
<td>( R_3 \leftarrow R_3 + R_5 ) [25]</td>
<td>*</td>
</tr>
<tr>
<td>( \Gamma_{11} )</td>
<td>( R_3 \leftarrow R_3 \times R_3 ) [-CE^2]</td>
<td>( R_3 \leftarrow R_3 + R_6 ) [25]</td>
<td>*</td>
</tr>
</tbody>
</table>
HPAS elliptic curve addition.

Input: $P = (R_1, R_2, R_3)$ in Jacobian coordinates
and $Q = (R_{10}, R_{11}, R_{12}, R_{13}, R_{14})$ in
Chudnovsky-Jacobian coordinates.

Output: $P \oplus Q = (R_1, R_2, R_3)$ in Jacobian coordinates.

<table>
<thead>
<tr>
<th>$\Gamma$</th>
<th>M</th>
<th>A</th>
<th>N</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma_1$</td>
<td>$R_4 \leftarrow R_5 \times R_6$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_2$</td>
<td>$R_6 \leftarrow R_{10} \times R_4$</td>
<td>$\ast$</td>
<td>$R_6 \leftarrow -R_6$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_3$</td>
<td>$R_6 \leftarrow R_{13} \times R_1$</td>
<td>$R_6 \leftarrow R_6 + R_5$</td>
<td>$R_6 \leftarrow -R_6$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_4$</td>
<td>$R_7 \leftarrow R_{11} \times R_3$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_5$</td>
<td>$R_7 \leftarrow R_7 \times R_4$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_6$</td>
<td>$R_8 \leftarrow R_8 \times R_3$</td>
<td>$\ast$</td>
<td>$R_8 \leftarrow -R_8$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_7$</td>
<td>$[E]_1$</td>
<td>$[E]_1$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_8$</td>
<td>$R_9 \leftarrow R_9 \times R_6$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_9$</td>
<td>$[E^2]$</td>
<td>$[E^2]$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_{10}$</td>
<td>$R_9 \leftarrow R_9 \times R_6$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_{11}$</td>
<td>$-[E^3]$</td>
<td>$-[E^3]$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_{12}$</td>
<td>$R_1 \leftarrow R_1 \times R_9$</td>
<td>$R_1 \leftarrow R_1 + R_9$</td>
<td>$R_1 \leftarrow -R_1$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_{13}$</td>
<td>$[F]_2$</td>
<td>$[F]_2$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_{14}$</td>
<td>$-[F(X_3-AB^2)]$</td>
<td>$[X_3]$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
<tr>
<td>$\Gamma_{15}$</td>
<td>$R_6 \leftarrow R_6 \times R_6$</td>
<td>$\ast$</td>
<td>$\ast$</td>
<td>$\ast$</td>
</tr>
</tbody>
</table>
FIG. 4A

410

411

P

412

MANA

8 steps

MANA

416

[2]P

FIG. 4B

420

424

QR

422

MANA

11 or 14 steps

MANA

426

Q ⊕ R

SUBSTITUTE SHEET (RULE 26)
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

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

INV. 606F 77/72

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbol(s))

606F

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

Electronic database consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>LARS ELMEGAARD-FESSEL: &quot;Efficient Scalar Multiplication and Security against Power Analysis in Cryptosystems based on the NIST Elliptic Curves Over Prime Fields&quot; THESIS., 1 August 2006 (2006-08-01), XP007906551 cited in the application Chapter 4</td>
<td>1-4, 7-10, 13 5, 6, 11, 12</td>
</tr>
</tbody>
</table>

X Further documents are listed in the continuation of box C. See patent family annex.

Special categories of cited documents:

*A* document defining the general state of the art which is not considered to be of particular relevance

*E* earlier document but published on or after the international filing date

*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)

*O* document referring to an oral disclosure, usage, exhibition or other means

*P* document published prior to the international filing date but later than the priority date claimed

*Y* document member of the same patent family

Date of the actual completion of the international search: 5 December 2008

Date of mailing of the international search report: 15/12/2008

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Fax (+31-70) 340-3016

Authorized officer

Prins, Leendert

Form PCT/IBA510 (second sheet) (April 2005)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>IZU T ET AL: &quot;IMPROVED ELLIPTIC CURVE MULTIPLICATION METHODS RESISTANT AGAINST SIDE CHANNEL ATTACKS&quot; PROGRESS IN CRYPTOLOGY–INDOCRYPT. INTERNATIONAL CONFERENCE INCRYPTOLOGY IN INDIA. PROCEEDINGS, XX, XX, 16 December 2002 (2002-12-16), pages 296-313, XP001160564 Section 2, Section 4.2</td>
<td>1-4, 7-10,13</td>
</tr>
</tbody>
</table>