Non Singular Elliptic Curves Analyze and Web Pages Authentication

Nicolae CONSTANTINESCU, Mirel COŞULSCHI

Faculty of Mathematics and Computer Science,
Department of Computer Science,
University of Craiova, Romania
{nikyc, mirelc}@central.ucv.ro

Abstract. Starting with the description of an elliptic curve E it is created a set of restrictions which helps to realize an implementation in a real system of the theories concerning the infeasibility of the ECDL problem. It is studied the non-singular elliptic curves case problem, where the infeasibility of the attack is increased by the particularity of the specific field. Also, there are presented the attack discussion of elliptic curve discrete logarithm and the way to improve the attack consistency.
Keywords: elliptic curves cryptography, web authentication
Math. Subjects Classification 2000: 11G07; 94A62

1 Introduction

Side-channel attacks rely on power consumption [8] and timing [7] as indicators for complexity of computations required for point multiplication on implementations of cryptosystems based on elliptic curves. The computational issue here is obtaining a result of the form:

$$R = [e]P$$

where R is the result, $[e]$ is a integer value, always secret, be it ephemeral or long-term used, also e is considered to be the key, while P is a point on the elliptic curve used with the cryptosystem.

The class of elliptic curves used in cryptographic applications requires that they have a prime-order sub-group. The order of this sub-group is denoted by P. The points of p order are used only.

Following measures can be employed to randomise computations and make power analysis a harder task for the attacker:

- If the cryptosystem uses projective coordinates for point representation, then the input point can be transformed in a random equivalent representation. (projective coordinates is the most often choice because of efficiency reasons [1],[5].

- The product $[e]P$ can be expressed using a random point Q as

$$R = [e](P - Q) + [e]Q$$

- The product $[e]P$ can be expressed using a random integer n as

$$R = [e + np]P$$

or

$$R = [e - n]P + [n]P$$

The inserted randomness considered above is meant to reduce the risk of differential attacks, which use correlations from multiple computations. However if direct interpretation of readings is possible above measures can become futile. Simple side-channel analysis works with implementations that use a straightforward approach to point multiplication. This is because the bits of the integer e leave quite an easy trace in the power consumption usage, which can be interpreted, along with the fact that P is publicly available this leads to the disclosure of the secret integer e if a double-and-add algorithm is used. Algorithms like m-ary or sliding window method may obscure the bits of e but according to [9] this may still reveal information.

Considering above issues, point representation seems the best solution at hand. Liardet and Smart [9], and Joye and Quisquater [6] proposed a set of curves and special representations that can be used in such manner that the same formula can be used for both addition and multiplication operations. Their solution however is not cost-effective and induces performance drawbacks.

In [10] the author proposes a point multiplication algorithm, which is improved by Okeya and Sakuray in [12] and then is used in [13] for suitable curves over odd characteristic fields, where the usual group operations are replaced by certain special operations working with triplets of points with y-coordinates omitted. This method makes the retrieval of bits from e much harder.

Details of the above methods are not discussed here, however they all have the disadvantage of unusability with NIST and SECG recommended curves from [11] and [2]. This leads to major drawbacks in what concerns interoperability.

This paper proposes a method by which limitations of specific usage of curves is removed by using a uniform pattern both for addition and multiplication. Because of this feature the method can help interoperability of systems.

The method uses more addition operations than the standard 2^w-ary point multiplication algorithm. Despite this, dummy additions used in other algorithms to achieve a fixed pattern of point doubling and addition [4] are avoided, reasons for this are given in next section.

Randomly choosing the integer e avoids failure of the considered method. Failure occurs in cases where addition involves actually a point doubling or the point at infinity. These situations are potentially clearly visible through side-channel analysis.

According to [13] methods of protection presented in the start of this section can be combined with the use of projective coordinates. We also recommend integrating the presented method with randomization techniques above mentioned.

2 Elliptic curves operations strength

Security analysis often uses a model in order to test side-channel information leakage. While it is practically impossible to analyse all possible information leakage, often this model is aimed at specific aspects, configurations or implementations.

Before presenting the method, information leakage is considered at a lower level. Subsection 2.1 discusses special cases of point operations that should be avoided. Subsection 2.2 discusses the importance of using randomized projective coordinates and certain extended point representations. Also this subsection questions the insertion of dummy point additions to achieve uniform behavior.

2.1 Point operations

Side-channel analysis is the operation which involves gathering of information concerning timing of computations and power consumption of such operations. Values that these indicators provide may be interpreted to obtain a certain order of operations involved in a cryptosystem. To conceal this order a careful devised algorithm should use point doubling and addition operations in order to create a uniform pattern, which should be independent of the specific multiplier, used in considered operations. Of course there are exceptions, and these situations should be treated in a special manner at the time of their occurrence. These are presented in the following statements:

- Point doubling $[2]A$ requires conditional statements for the case that A is the point at infinity or that A is a point of order two. If these cases are avoided, then, expressed in field operations, point doubling runs as a fixed routine.
- Point addition $A + B$ requires conditional statements for the case that one of the points is the point at infinity, or that A coincides with B, or that one point is the inverse of the other. For other cases, it too can be implemented as a fixed routine.

Details of the sequences of field operations used for point doubling and point addition depend on the underlying field (odd characteristic vs. characteristic 2) and the choice of point representations (e.g. either affine coordinates or one of multiple styles of projective coordinates, according to [3]), so implementations may vary widely. The essential observation is that the respective algorithm always behaves the same as long as the above special cases are avoided.

2.2 Finite field operations

An important observation is that when an attacker analyses the side-channel information he does not have immediate access to the factors involved in a operation. However it is prudent to say that not all operations look the same, and this is the basic idea for side-channel attacks. Inserting randomization techniques into one's protocol or any other cryptosystem based on elliptic curves is a good idea. This is combined with the usefulness of projective coordinates [4],[13]. Take for example Jacobian projective coordinates, which are triplets of the form (X, Y, Z) with $Z = 0$, they represent affine points $(X/Z^2, Y/Z^3)$; then for any field element $\epsilon \neq 0$,$(\epsilon^2 X, \epsilon^3 Y, \epsilon Z)$ is a representation of the same point on the curve. Randomization makes it difficult for an attacker to guess the values obtained by using a randomly chosen ϵ.

Point doubling or point addition using projective coordinates results in a point represented with a Z-coordinate that is the product of the Z-coordinate(s) of the input point(s) and a short polynomial involving one or more other coordinates of the input points; thus the output point is again in a randomized representation.

Randomization makes it difficult for an attacker to guess or imply that a certain operation involving known points is taking place at a certain point in time. Still the attacker may observe the same operation reoccurring if the same field operation is executed several times throughout the computation. Even if the attacker cannot obtain the factors involved in the operation we still want to mask this as this in some cases may be considered an important information leakage.

Point multiplication, $R = [e]P$, is performed in stages by a great part of existing algorithms, these stages are as follows:

Precompute stage: First, independently of the specific multiplier e, certain small multiples of P are computed and stored in a table.

Evaluation stage: Second, the product $[e]P$ is evaluated as follows: A variable A is initialized to one of the table values; then, many times, A either is doubled or a table value is added to A, replacing the previous value of A. Finally, A contains the result $[e]P$.

3 Implementation

In order to implements in calculation systems arithmetic in F_p are used, where p is a prime number, large enough to meet certain conditions required by the present problem. The main problems under consideration refer to calculation in F_p: addition and multiplication. The latter is also the most difficult to solve. In order to create an efficient algorithm in [14–16] we present methods which start from a special p form i.e. $p = b^t - a$, where a has a sufficiently low value. The algorithm is based on multiplication subroutine, followed by reduction subroutine such as

Algorithm 1
1 $q_0 \leftarrow \lfloor x/b^t \rfloor, r_0 \leftarrow x - q_0 b^t, r \leftarrow r_0, i \leftarrow 0$
2 *while* $q_i > 0$ *do*
– $q_{i+1} \leftarrow \lfloor q_i a/b^t \rfloor$, $r_{i+1} \leftarrow q_i a - q_{i+1} b^t$
– $i \leftarrow i + 1$, $r \leftarrow r + r_i$
3 *while* $r \geq p$ *do* $r \leftarrow r - p$

In this way the reduction function uses only shift operations, addition and multiplication by a.
For the calculation of certain parameters found in the systems implemented in practice, RNSA (Residue Number System Arithmetic) is used. This concept is a rather old one and is based on CRT (Chinese Remainder Theorem). Therefore, starting from the integer p, as defined above, we choose p_i prime numbers, so that

$$\prod_{i=1}^{t} p_i > p^2 \tag{1}$$

We will represent an element x *modulo* p as a vector $(x_1, \ldots, x_t)$, where $x \equiv x_i \ (\ mod\ p_i)$. With this representation there can be made fast implementations on calculating machines which use 32 or 64 bit-word one of this ways in which such interpretation can be used is in trapdoor functions, applications of this kind being found in the algorithms of Public-Key systems.
Another efficient method of implementing modulo a large prime p arithmetic consists in using Montgomery representation [17] . Let be b the base in which the system works. R and t will be defined so that $R = b^t > p$ will made. We conclude from this that which element $x \in F_p$ is represented by $xR(mod\ p)$. The reduction operation required by the multiplication process is based on the result provided by Lemma 1

Lemma 1 *Let be* $0 \leq y \leq pR$, $u = -yp^{-1}(mod\ R)$ *and*

$$x = \frac{(y + up)}{R}$$

then x *is an integer such that* $x < 2p$ *and* $x \equiv yR^{-1}(mod\ p)$

Also, the algorithm to compute the Montgomery reduction is:

Algorithm 2
1 $u \leftarrow -yp^{-1}(mod\ R)$
2 $x \leftarrow (y + up)/R$
3 *if* $x \geq p$ *then* $x \leftarrow x - p$
4 *return* x

In case of $y = (y_{2t-1}, \ldots, y_1, y_0)_{mod\ b} = y_{2t-1}b^{2t-1} + \ldots + y_1 b + y_0$ then we can compute $yR^{-1}(mod\ p)$ in the following way:

Algorithm 3

1 *for* $i = 0$ *to* $t - 1$
- $u \leftarrow y_i p' \ (mod\ b)$
- $y \leftarrow y + upb^i$

2 $z \leftarrow y/R$
3 *if* $z \geq p$ *then* $z \leftarrow z - p$
4 *return* z

These calculi are made in case of $p' = -p^{-1}(mod\ b)$. In order to find this one it is necessary to compute $x^{-1}(mod\ 2^w)$.

Algorithm 4

1 $y \leftarrow 1$
2 *for* $i = 2$ *to* w
- *if* $2^{i-1} < xy(mod\ 2^i)$ *then* $y \leftarrow y + 2^{i-1}$

3 *return* y

Another important aspect which must be taken into consideration is to solve quadratic equation in modulo p finite fields. These are necessary for calculating an $y - coordinate$ of a point on the elliptic curve. It is found by starting from the $x - coordinate$. The equation to be solved is of the type: $x^2 \equiv a(mod\ p)$. In order to taste that such an equation has a solution we will calculate Legendre symbol $\left(\frac{a}{p}\right)$, whose value will be 1 in case a is a square modulo p or the value will be 0 in case $a \equiv 0(mod\ p)$. If we are in none of the above cases Legendre symbol will be -1. The algorithm is presented below

Algorithm 5

1 *if* $a \equiv 0\ (mod\ p)$ *the return 0*
2 $x \leftarrow a$, $y \leftarrow p$, $L \leftarrow 1$
3 $x \leftarrow x\ (mod\ y)$
4 *if* $x > y/2$ *then*
- $x \leftarrow y - x$
- *if* $y \equiv 3(mod\ 4)$ *the* $L \leftarrow -L$

5 *while* $x \equiv 0\ (mod\ 4)$ *do* $x \leftarrow x/4$
6 *if* $x \equiv 0\ (mod\ 2)$ *then*
- $x \leftarrow x/2$
- *if* $y \equiv \pm 3\ (mod\ 8)$ *then* $L \leftarrow\ -L$

7 $x = 1$ *then return* L
8 $x \equiv 3\ (mod\ 4)$ *and* $y \equiv 3\ (mod\ 4)$ *then* $L \leftarrow\ -L$
9 $temp \leftarrow x$
10 $x \leftarrow y$
11 $y \leftarrow temp$
12 *go to* 3

During the computational process, the machine representation is made in base 2, so that, in order to optimize the algorithms all necessary arithmetic must be translated in the finite fields F_{2^n}. Therefore, let be a quadratic equation

$$x^2 + \beta = 0 \tag{2}$$

in F_{2^n}, where its double square will be $x_0 = \beta^{2^{n-1}}$. A nontrivial quadratic equation $x^2 + x + \beta = 0$ will have, in F_{2^n}, a solution of the type $x_0 = \tau(\beta)$, where

$$\tau(\beta) = \sum_{j=0}^{(n-1)/2} \beta^{2^{2j}} \tag{3}$$

Let be the matrix $T = (T_{ij})$.

$$\alpha^{1+2i} = \sum_{j=0}^{n-1} T_{ij}\alpha^{2^j},\ 0 \le i \le n-1 \tag{4}$$

where $(\alpha, \alpha^2, \alpha^{2^2},\ ..., \alpha^{2^{n-1}})$ is a normal base in F_{2^n} over F_2 and $\alpha \in F_2$. $Tr_{q|2}(\alpha_i\alpha_j) = 1$ iff $i = j$, $Tr_{q|2}(z)$ is the trace of $z \in F_q$ over F_2, with $q = 2^n$.

4 Optimization in elliptic curves arithmetic

As the elliptic curves theory was founded a long time ago there is a large variety of interpretations and also ways to solve them. Let be an integral of type

$$\int \frac{dx}{\sqrt{4x^3 - h_2x - h_3}} \tag{5}$$

The inverse function of such an integral is called elliptic function. Let be two constants α_1 and α_2, a function and a double periodic function over R then Weierstrass function will be of the type

$$(\gamma')^2 = 4\gamma^3 - \alpha_1\gamma - \alpha_2 \tag{6}$$

This pair (γ, γ') will define a point on the curve

$$y^2 = 4x^3 - \alpha_1 x - \alpha_2 \tag{7}$$

making an elliptic curve.

Definition 1 *Let be $p > 3$ a prime integer. The elliptic curve $y^2 = x^3 + \alpha_1 x + \alpha_2$, defined over Z_p is the set of solutions $(x, y) \in Z_p \times Z_p$ to the congruence*

$$y^2 \equiv x^3 + \alpha_1 x + \alpha_2 \,(mod\, p) \tag{8}$$

where $\alpha_1, \alpha_2 \in Z_p$ are constants such that $4\alpha_1^3 + 27\alpha_2^2 \not\equiv 0 \,(mod\, p)$ together with a special point O called the point at infinity.

As already described in section 2 the main problems are to define the addition of two points in such a field and to make multiplications by a given integer of a point on the elliptic curve. The problem of adding two points, be them A_1 and A_2 is divided between $x_1 = x_2$ and $y_1 = y_2$ on the one hand and the other cases on the other hand.

Lemma 2 *Let E denote an elliptic curve given by*

$$E : Y_2 + \alpha_1 XY + \alpha_3 Y = X^3 + \alpha_2 X^2 + \alpha_4 X + \alpha_6 \tag{9}$$

and let $A_1 = (x_1, y_1)$ and $A_2 = (x_2, y_2)$ two points on the curve. Then

$$-A_1 = (x_1, -y_1 - \alpha_1 x_1 - \alpha_3) \tag{10}$$

Set

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1},\ \gamma = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1} \tag{11}$$

where x_1, x_2 satisfy the condition $x_1 \neq x_2$ and, from this point we will have

$$\lambda = \frac{3x_1^2 + 2\alpha_2 x_1 + \alpha_4 - \alpha_1 y_1}{2y_1 + \alpha_1 x_1 + \alpha_3} \tag{12}$$

,

$$\gamma = \frac{-x_1^3 + \alpha_4 x_1 + 2\alpha_6 - \alpha_3 y_1}{2y_1 + \alpha_1 x_1 + \alpha_3}. \tag{13}$$

In case of equality between x_1 and x_2 and the points $A_2 \neq -A_1$ the addition of these two points will be the point A_3 with the following coordinates:

$$x_3 = \lambda^2 + \alpha_1 \lambda - \alpha_2 - x_1 - x_2,\ y_3 = -(\lambda + \alpha_1) x_3 - \gamma - \alpha_3 \tag{14}$$

Thus we will have

1. $x_2 = x_1$ and $y_2 = y_1$. Then $A_1 + A_2 = O$
2. Otherwise $A_1 + A_2 = B$, $B(x_3, y_3)$ where

$$x_3 = \lambda^2 - x_1 - x_2, y_3 = \lambda(x_1 - x_3) - y_1 \tag{15}$$

and

$$\lambda = \begin{cases} (y_2 - y_1)(x_2 - x_1)^{-1}, A_1 \neq A_2; \\ (3x_1^2 + a)(2y_1)^{-1}, A_1 = A_2; \end{cases} \tag{16}$$

5 WEB attack study

This section concentrates on extending the attack of Girault and Misarsky's multiplicative attack on WEB RSA signatures with affine redundancy to a level where we have the size of the message equal to one third of the RSA modulus n. A multiplicative attack is an attack in which the redundancy function of a message can be expressed as a multiplicative combination of the redundancy

functions of other messages. With respect to this we search for four messages, m_1, m_2, m_3, m_4, which are at least one third of the size of the modulus n, and verify the following equation

$$\begin{aligned} &R(m_1) \cdot R(m_2) = \\ &R(m_3) \cdot R(m_4) \pmod{n} \end{aligned} \tag{17}$$

Message m_1, is the message whose signature will be forged, this can be done by computing

$$R(m_1)^d = \frac{R(m_3)^d \cdot R(m_4)^d}{R(m_2)^d} \pmod{n}$$

From (17) we obtain:

$$\begin{aligned} &(\omega \cdot m_1 + a) \cdot (\omega \cdot m_2 + a) = \\ &(\omega \cdot m_3 + a) \cdot (\omega \cdot m_4 + a) \pmod{n} \end{aligned}$$

Denoting $P = a/\omega \bmod n$, we obtain:

$$\begin{aligned} &(P + m_1) \cdot (P + m_2) = \\ &(P + m_3) \cdot (P + m_4) \pmod{n} \end{aligned}$$

For the following substitutions

$$\begin{aligned} t &= m_3 \\ y &= m_2 - m_3 \\ x &= m_1 - m_3 \\ z &= m_4 - m_1 - m_2 + m_3 \end{aligned} \tag{18}$$

the following equation holds

$$\begin{aligned} &((P + t) + x) \cdot ((P + t) + y) = \\ &(P + t) \cdot ((P + t) + x + y + z) \pmod{n} \end{aligned}$$

which simplifies into

$$x \cdot y = (P + t) \cdot z \pmod{n} \tag{19}$$

Next we need to determine the values x, y, z and t with respect to 19. First, we obtain two integers z and u such that

$$\begin{aligned} &P \cdot z = u \pmod{n} \\ &\text{with } \begin{cases} -n^{\frac{1}{2}} < z < n^{\frac{1}{3}} \\ 0 < u < 2 \cdot n^{\frac{2}{3}} \end{cases} \end{aligned}$$

One solution is suggested by [18]. Finding a good approximation to the fraction $\frac{P}{n}$ can be done efficiently by developing it in continued fractions. This implies using the extended Euclidean algorithm to P and n. A solution is found such that $|z| < Z$ and $0 < u < U$ if $Z \cdot U > n$, which is the case here with $Z = n^{\frac{1}{3}}$ and $U = 2 \cdot n^{\frac{2}{3}}$.

We then select an integer y such that

$$n^{\frac{1}{3}} \leq y \leq 2 \cdot n^{\frac{1}{3}}$$

and $gcd(y, z) = 1$. We find the non-negative integer $t < y$ such that:

$$t \cdot z = -u \pmod{n}$$

which is possible since $gcd(y, z) = 1$. Then we take

$$x = \frac{u + t \cdot z}{y} \leq 4n^{\frac{1}{3}}$$

and obtain

$$P \cdot z = u = x \cdot y - t \cdot z \pmod{n}$$

which gives equation (19), with x, y, z and t being all smaller than $4 \cdot n^{\frac{1}{3}}$. From x, y, z, t we derive, using (18), four messages m_1, m_2, m_3 and m_4, each of size one third the size of n:

$$\begin{aligned} m_1 &= x + t \\ m_2 &= y + t \\ m_3 &= t \\ m_4 &= x + y + z + t \end{aligned} \tag{20}$$

Since $-n^{1/3} < z < n^{1/3}$ and $y \geq n^{1/3}$, we have $y + z > 0$, which gives using $u \geq 0$

$$x + t = \frac{u + t \cdot (y + z)}{y} \geq 0$$

which shows that the four integers m1, m2, m3 and m3 are non-negative, and we have

$$R(m_1) \cdot R(m_2) = R(m_3) \cdot R(m_4) \pmod{n}$$

The complexity of our attack is polynomial in the size of n.

6 Existence of selective forgery

The attack discussed in the previous section is existential which means that the attacker needs to find the four messages required for forgery; if the messages m_2, m_3, m_4 do not exist then the attack is not possible. This section deals with the possibility of a selective forgery attack, but in this case the attack no longer runs in polynomial time. Let m_3 be the message whose signature must be forged. Letting x, y, z and t as in (18), we compute two integers z and u such that

$$(P + t) \cdot z = u \pmod{n}$$
$$\text{with } \begin{cases} -n^{\frac{1}{2}} < z < n^{\frac{1}{3}} \\ 0 < u < 2 \cdot n^{\frac{2}{3}} \end{cases}$$

We then factor u, and try to write u as the product $x \cdot y$ of two integers of roughly the same size, so that eventually we have four integers x, y, z, t of size roughly one third of the size of the modulus, with:

$$x \cdot y = (P + t) \cdot z \pmod{n}$$

which gives again

$$R(m_1) \cdot R(m_2) = R(m_3) \cdot R(m_4) \pmod{n}$$

The signature of m_3 can now be forged using the signatures of m_1, m_2 and m_4. For a 512-bit modulus the selective forgery attack is truly practical. For a 1024-bit modulus the attack is more demanding but was still implemented with success.

References

[1] **I. F. Blake, G. Seroussi, N. P. Smart**:- Elliptic Curves in Cryptography, vol. 265 of London Mathematical Society, Lecture Note Series, Cambridge University Press, 1999

[2] *******:- Certicom Research. Standards for efficient cryptography SEC 2: Recommended elliptic curve cryptography domain parameters, Version 1.0, 2000, Available from http://www.secg.org/

[3] **H. Cohen, T. Ono, A. Miyaji**:- Efficient elliptic curve exponentiation using mixed coordinates. In: K. Ohta, D. Pei (Eds.), Advances in Cryptology ASIACRYPT 98, vol. 1514 of Lecture Notes in Computer Science, 1998, 51-65

[4] **J.-S. Coron**:- Resistance against differential power analysis for elliptic curve cryptosystems. In: C. K. Koc, C. Paar (Eds.), Cryptographic Hardware and Embedded Systems CHES 99, vol. 1717 of Lecture Notes in Computer Science, 1999, 292-302

[5] *******:- Institute of Electrical and Electronics Engineers (IEEE), IEEE standard specifications for public-key cryptography. IEEE Std 1363-2000, 2000

[6] **M. Joye, J. -J. Quisquater**:- Hessian elliptic curves and side-channel attacks. In: C. K. Koc, D. Naccache, C. Paar (Eds.), Cryptographic Hardware and Embedded Systems CHES 2001 [Pre-]Proceedings, 2001, 412-420

[7] **P. C. Kocher**:- Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: N. Koblitz, (Ed.), Advances in Cryptology CRYPTO 96, vol. 1109 of Lecture Notes in Computer Science, 1996, 104-113

[8] **P. C. Kocher, J. Jaffe, B. Jun**:- Differential power analysis. In: M. Wiener (Ed.), Advances in Cryptology CRYPTO 99 , vol. 1666 of Lecture Notes in Computer Science, 1999, 388-397

[9] **A. Miyaji, T. Ono, H. Cohen**:- Efficient elliptic curve exponentiation. In: Y. Han, T. Okamoto, S. Qing (Eds.), International Conference on Information and Communications Security ICICS 97 , vol. 1334 of Lecture Notes in Computer Science, 1997, 282-290

[10] **P. L. Montgomery**:- Speeding the Pollard and elliptic curve methods of factorization, Mathematics of Computation 48, 1987, 243-264

[11] ***:- National Institute of Standards and Technology (NIST), Digital Signature Standard (DSS), FIPS PUB 186-2, 2000

[12] **K. Okeya, H. Kurumatani, K. Sakurai**:- Elliptic curves with the Montgomery-form and their cryptographic applications. In: H. Imai, Y. Zheng (Eds.), Public Key Cryptography PKC 2000, vol. 1751 of Lecture Notes in Computer Science, 2000, 238-257

[13] **K. Okeya, K. Sakurai**:- Power analysis breaks elliptic curve cryptosystems even secure against the timing attack. In: B. K. Roy, E. Okamoto (Eds.), Progress in Cryptology INDOCRYPT 2000, vol. 1977, Lecture Notes in Computer Science, 2000, 178-190

[14] **R. Crandall**:- Method and apparatus for public key exchange in a cryptographic system, U. S. Patent Number 5159632

[15] **R. Lercier, F. Morain**:- Counting points in elliptic curves over F_{p^n} using Couveignes algorithm, Rapport de Recherche LIX/RR/95/09.1995

[16] **J.H. van Lint**:- Introduction to Coding Theory, Springer-Verlag, 1982

[17] **P.L. Montgomery**:- Modular multiplication without trial division, Math. Comp., Vol 44, 1985, 519-521

[18] **M. Girault, P. Toffin, B. Vallee**:- Computation of approximation L-th roots modulo n and application to cryptography, Proceedings of Crypto '88, LNCS vol. 403, Springer-Verlag, 1988, 100-117.