A cryptography method employing a new mathematical paradigm for public keys schemes

Vinicius Gadis Ribeiro*
FACENSA, UNILASALLE and UNIRITTER Professor
Porto Alegre, Rio Grande do Sul, Brasil
and
Raul Fernando Weber
UFRGS Professor
weber@inf.ufrgs.br
Porto Alegre, Rio Grande do Sul, Brasil

ABSTRACT

Public Key Schemes usually generate two pairs of keys that bear some mathematical relation -- usually, a relation of the field known as Number Theory. Identifying a key from another that is given is a problem of great difficulty. This paper presents a public key scheme in which such key identification is also a great difficulty problem -- however, the mathematical problem does not fall under Number Theory; rather, it is one of Differential Equations. The proposed scheme is based on the difficulty of solving differential equations, the rules of Lie groups being the best solution. Even using Lie groups, a problem of great difficulty still remains. The first few examples use Maple 5.0, a symbolic processing program, and are available in an Internet site.

Keywords: Computer Security, Cryptography, Public Key Cryptography, Lie Groups, Symbolic Processing.

1. INTRODUCTION

In 1976, Whitfield Diffie and Martin E. Hellman of Stanford University (California) published *New Directions in Cryptography*, which first presented relevant concepts that enable the use of public key schemes in cryptosystems. Among these concepts, one stands out: the use of different keys, one called public because it can be widely distributed, and another one called private, which must be kept secret [21]. One of them is used to encrypt the message, and the other to decrypt it. Though calculated from a common origin, calculating these keys from one of them is prohibitive. In other words, having one of them -- the public one, for instance -- one cannot calculate the other -- the private one -- in a time such that the information they hide is still valuable. In fact, the time needed to effect such calculation is estimated as thousand of years [2][17][22].

The use of this concept solved the problem of key distribution, since public keys can be presented anywhere and circulate freely.

The requirements to employ Public Key cryptography are related to the property of the pair of keys that each user generates [21][18]:

- if $C = E(M, K_e)$, then $M = D(C, K_d)$, for every M.
- it is computationally impossible to calculate K_d, based on K_e.
- it is computationally possible to calculate the pair of keys K_e and K_d, while meeting the requirements above.

where M is the Message to transmit; C, the operation to encrypt, and D, the decrypting operation.

This paper presents a public key scheme that is not based on number theory, but rather on differential equations -- thus setting a new paradigm. The new scheme is based on Lie groups -- more specifically, translation symmetry on the complex plane -- instead of discrete Galois groups [19]. In other words, the attacker's job is focused on the difficulty of finding out the root of the function that corresponds to the Private Key of one of the scheme participants -- i.e., solving algebraic or differential equations -- by applying the corresponding restrictive conditions, instead of running primality tests[3][12][14]18], solving discrete logarithms [1][21], or defining the scalar in elliptic curves [1][8]. Even if the attacker tries to solve this problem with Lie groups, the number of symbolic operations required makes the procedure ineffective due to their processing time[24]. Besides that, even the direct search of private keys by scanning -- equivalent to trial and error processes on Number Theory -- becomes considerably more costly under the new paradigm, since the searched set is a continuous power, as opposed to the set of prime numbers, for instance, which is numerable.

The major difficult in solving the inverse problem, e.g. determining the complex shifts which performs the enciphering process lies on the need a contour problem whose solutions are the enciphered messages. Indeed, without knowing the differential equation and the corresponding set of boundary conditions, the task of finding the complex shifts is unfeasible. Even the adversary know the differential equations and the boundary conditions, it is a very hard problem to obtain the Lie group admitted by the problem. This occurs because the determining equations [15] -- which constitutes a set of auxiliary partial differential equations that must to be solved in order to obtain the respective infinitesimal generators --, are often much complex than the original one. Formal features related to this process can be found in Chari [23] and Olver [16].

Here is a summary of how the paper is organized: section two presents the new paradigm, presenting continuous rather than discrete groups; section three describes the encrypting algorithm, focusing on a ciphering method based on translational Lie symmetries. Section four presents results of the application of the method to real situations. Section five brings the paper to a close by presenting the main advantages of employing the new paradigm and the corresponding conclusions based on the results obtained.

2. A NEW PARADIGM FOR PUBLIC KEY SCHEMES

The property that the new paradigm is based upon is the commutability of the differential operators involved in the ciphering and deciphering processes. This property is also valid on Galois groups, though applied restrictively. Whereas commutability arises naturally -- in the

algorithms based on Galois groups -- from the fact that the elements of the groups used are integers, this property, in algorithms that use Lie groups, is imposed to a certain class of differential operators -- which makes the process more flexible, regarding private key choice. In this case, the original message can be represented by continuous functions, or else by differential operators. This work represents the message by a linear combination of infinitely derivable functions, whose numerical coefficients contain the ASCII codes of the characters in the original message.

By employing the new paradigm, the user has great freedom to choose private keys, i.e., there is no need to generate long mantissa prime numbers.

However, some restrictions must be observed, regarding the form of the function associated to the original message:

1 - parts cannot be repeated in building the function that corresponds to the message;

2 - linear combinations of functions already inserted in building the function that corresponds to the message cannot be included;

3 - it is advisable to employ whole coefficients as the multiplying factors of the employed functions; and

4 - the use of independent terms in the arguments of exponential functions is not allowed.

It is important to notice that, in schemes based on Galois groups, participants must necessarily produce their Private Keys from a Public Key, given the numerical restrictions involved in the corresponding inverse process. Under the proposed paradigm, although it is also possible to produce Private Keys from a Public Key, it is not necessary to do so, since the set of Private Keys is a continuous power. This way, participants can arbitrate their Private Keys independently -- which are both a further safety element and also a considerable advantage from an operational point of view.

The next section presents a symbolic scheme based on the new paradigm. The proposed scheme called Rafaella, uses translation symmetries on a complex plane -- a specific case of the use of continuous groups in differential equations.

3. MODELING THE RAFAELLA SCHEME

The process can be formally described in a similar way to the models of Brawley and Gao [10], where

- the space of Private Key A is the set of complex numbers z = a + ib, where a and b are real non-zero numbers.
- the space of messages F is the set of continuous and infinitely derivable functions on the complex plane.
- the space of Public Keys K is the set of continuous and infinitely derivable functions on the complex plane.
- for every $z \in A$, the ciphering operator is given by the expression $Cf(x) = f(x+z)$.
- for every $z \in A$, the private keys are z_1 and z_2.
- for every $z \in A$, the ciphering operator is written as $Df(x) = f(x-z)$, which corresponds to the inverse ciphering operator (C^{-1}).
- the C and D operators are defined, respectively, as $C = e^{z\frac{\partial}{\partial x}}$, and $D = e^{-z\frac{\partial}{\partial x}}$.

However, unlike the Brawly and Gao models, the proposed scheme is not based on the use of one-way functions, i.e., functions whose inversion process is inefficient. The Rafaella scheme is based on the use of Lie symmetries, i.e., the intrinsic difficulty of the inversion process is not due to calculating f^{-1}, but only to the difficulty of identifying the cipher operator itself.

4. CONSIDERATIONS ABOUT PUBLIC KEYS

Under the proposed scheme, the Public Key -- consisting of a continuous function or a differential operator -- does not necessarily generate private keys. For example, let one consider operator A[1] -- used as a Public Key -- and defined as

$$A = \left(a_1 \frac{\partial}{\partial_x} + a_2 \frac{\partial}{\partial_y} \right). \quad (1)$$

The exponential of this operator, when applied to an arbitrary function, produces the following transformation

$$\left[e^{A} \right] f(x,y) = f(x + a_1, y + a_2). \quad (2)$$

This transformation corresponds to a translation on the real line or on the complex plane, depending on the nature of coefficients a_1 and a_2. Under the proposed scheme, coefficient a_1 and a_2 have real and imaginary parts, which determines a translation operation on the complex plane. Equation (2), therefore, defines the origin of the private keys, i.e., the process of generating private keys from a public key. Because the exponentiation of operator A produces an infinite series of powers of this operator, the process corresponding to the evaluation of the left-hand side of (2) is extremely costly, whereas the same effect obtained through direct translation -- by the use of complex numbers, corresponding to the right-hand side of the same equation -- is a trivial operation.

From an operational point of view, the equivalence of plugging differential operators on the function corresponding to the message and translating it verifies directly through the use o f Taylor series in assessing the offset functions:

$$F(x+\Delta x) = F(x) + \Delta x \left. \frac{dF}{dx} \right|_x + \frac{(\Delta x)^2}{2!} \left. \frac{d^2F}{dx^2} \right|_x +$$

$$\ldots + \frac{(\Delta x)^k}{k!} \left. \frac{d^kF}{dx^k} \right|_x + \ldots = \sum_{k=0}^{\infty} \frac{(\Delta x)^k}{k!} \left. \frac{d^kF}{dx^k} \right|_x \quad (3)$$

This operation corresponds to applying the exponential of a differential operator over the function corresponding to the message, since

$$\sum_{k=0}^{\infty} \frac{(\Delta x)^k}{k!} \left. \frac{d^k f}{dx^k} \right|_x = \left[e^{\Delta x \frac{d}{dx}} \right] f(x), \quad (4)$$

This way, it is possible to offset a function by applying the exponential of a differential operator of first degree with constant coefficients, i.e.,

$$\left[e^{a\frac{d}{dx}} \right] f(x) = f(x + a), \quad (5)$$

where $a = \Delta x$ is a real or complex constant.

Although the enciphering scheme being a straightforward procedure, the inverse problem constitutes a very expensive task. At first glance, the standard procedures for

[1] Traditionally, capital letters are used to represent operators -- or matrices.

calculating complex roots, for instance, Newton-Bairstow, steepest descent, Lanczos and conjugate gradients [7], can be employed in order to determine the shifts in the complex plane which generates the encrypted messages. However, because the high truncation errors arising from the application of these schemes, this task becomes prohibitive from the numerical point of view. In addition, the methods based on Lie symmetries are not suitable for finding the complex shifts, because the starting point of these procedures is the differential equation satisfied by the messages, which is never known beforehand. Besides, even knowing the differential equation, the corresponding boundary conditions must be also a priori known in order to carry out the attack based on any procedure based on Lie groups [6][15][16].

5. DESCRIPTION OF THE SCHEME

From a given message, the encrypting process consists of ten basic steps:

1. Converting the original message to ASCII code, making up the coefficients of the function corresponding to the message -- m_0;
2. Choosing a real, continuous and "n" times derivable function, with a number of parts equal to the number of resulting numerical coefficients -- and all parts must be distinct from one another -- followed by the addition of the Public Key of the participant receiving the message. Production of authentication arguments, which contain the product of whole powers of the Public Keys for each participant.
3. The sender's offsetting on the complex plane, with real and imaginary components;
4. Sending the ciphered message to the authorized receiver;
5. The receiver's offsetting with real and imaginary components, and generation of an auxiliary argument consisting of a continuous function;
6. Sending the mapped message, and the auxiliary argument, to the sender;
7. Application, by the sender, of the inverse change, and verification of authenticity of receiver's auxiliary argument by the generation of a new argument -- called verification argument (the new argument is used to distort the ciphered message in case the receiver's authenticity is not verified);
8. Sending, to the receiver, the mapped message and the auxiliary argument generated by the sender;
9. Application of the inverse change by the receiver, verifying authenticity of auxiliary argument of sender and extracting receiver's public key;
10. Recovering original characters -- which are the coefficients of the original function.

The previous codification of the message consists of determining the ASCII codes of each character involved, which will make up the coefficients of function m_0.

Choosing a continuous function consists in determining a one-variable f(x) function from the composition of the basic functions available in conventional programming languages: sin(x), cos(x), atn(x), ln(x), exp(x) and polynomials.

Function f(x) can be obtained through the linear combination of two or more functions on the list, compositions between functions, or both. For example, function

$$c_1 e^x + c_2 \operatorname{sen} x + c_3 x^2 + c_4 x^6 \tan x + e^{x^2} \quad (6)$$

-- where c1, c2, c3 and c4 are numerical coefficients that represent the ASCII codes of the original message -- contains compositions and linear combinations applied alternatively.

The application of the symmetry of offsetting containing the real and imaginary parts consists of the following change of variable

$$x \to x + a + ib. \quad (7)$$

This change maps function f(x) into function f(x + a + ib).

The application of inverse symmetry consists of changing the variable that works counter wise, i.e.,

$$x \to x - a - ib, \quad (8)$$

which maps f(x) into f(x - a - ib).

It must be noted that the private keys of each participant of the scheme are the real and imaginary components of the offsetting applied to function f. This way, each participant arbitrates a complex number, which is employed exclusively to **cipher** and **decipher the function**[2]. The Public Key of the **proposed scheme can consist**, on its turn, of a **differential operator**, built from the private keys. For example, private keys a = 190 + 65i, and b = 135 - 102i generate differential operator A, defined as

$$25650\left(\frac{\partial^2}{\partial x^2} f(x,y)\right) - 10605\left(\frac{\partial}{\partial y \partial x} f(x,y)\right) - 6630\left(\frac{\partial^2}{\partial y^2} f(x,y)\right) \quad (9)$$

Having got hold of this Public Key, the process of reconstitution of both private keys is extremely costly. It must be noticed that operator A, in the given example, was obtained from a rather simple factored form.

Choosing a key in the form of a differential operator gives the attacker counter-information, inducing the conclusion that it is an operator present in the very differential equation satisfied by f0 -- which is not necessarily true.

6. CIPHERING AND DECIPHERING PROCESSES

It is important to point out that the use of Lie symmetries can be effected through the application of differential operators with constant coefficients on functions, which makes up the encrypting process proposed here. The inverse process, i.e., inverting the differential operator, and its later application on the operated function, makes up the attacker's message decrypting process.

From a participant's point of view, the original message M must be codified in a way such that it can be represented by a function f. The operation of encrypting the message is equivalent to applying the operator that transforms function f so that

$$Cf = g, \quad (10)$$

where g is a function corresponding to the encrypted message -- whereas the decrypting operation will be noted by determining f, so that

$$f = C^{-1} g. \quad (11)$$

The traditional way to discover function f consists, therefore, of obtaining the solution of equation Cf = g, by inverting operator C. However, the process of inverting

[2] Ciphering a function is to map a real variable f(x) function into a complex variable function, which corresponds to a specific case of translational Lie symmetry.

operator C requires high processing time, so the classical procedure should be avoided.

The alternate procedure consists of employing certain differential operators to carry out the codification and decodification of the message. Consider A, the differential operator that effects translation on the complex plane, chosen from Alice's private key, and B, the differential operator that effects the translation on the complex plane, chosen from Bob's private key, like Alice. The message to codify must be expressed in the form of a function, f. Alice proceeds as follows:

$$g = Af \rightarrow g_1 = Bg \rightarrow g_2 = A^{-1}g_1 \rightarrow g_3 = B^{-1}g_2 = f .$$

It can be shown as follows:

$$g = Af \rightarrow g_1 = Bg = BAf$$
$$g_2 = A^{-1}g_1 = A^{-1}BAf$$
$$g_3 = B^{-1}g_2 = B^{-1}A^{-1}BAf$$

Assuming A and B to commute, it can be said that

$$g_3 = B^{-1}B\,A^{-1}A\,f \text{ , and}$$
$$A^{-1}A = B^{-1}B = I .$$

Therefore,

$$g_3 = I\,If = If = f .$$

So, operators A and B are required to commute with each other, i.e., AB=BA.

7. THE AUTHENTICATION PROCESS

Under the scheme proposed, the authentication process is carried out by the calculation of the auxiliary arguments; the first and the second are continuous functions, and the third is a complex number. The first, called *authentication argument*, is a continuous function made up of products between powers of the public keys of both participants. The receiver must plug its private key on this function in order to produce a new function. The sender then uses this new function to verify that the receiver is authentic, with the following test of authenticity:

The sender plugs its private key in the function obtained, and verifies the resulting complex number, called *verification argument*. In case the verification argument results null, sender's authenticity is verified; otherwise, the resulting complex number is multiplied by the derivative of the ciphered message at its current state, in order to change its content. This way, only the authorized receiver will be able, at the end of the process, to retrieve the original message. The authorized sender authentication process is effected likewise. However, the respective verification argument is not used to distort the message by the receiver, but only to check sender's authenticity.

The process of authentication is described quantitatively as follows: from authentication arguments -- *aa* and *ab*, respectively defined as

$$aa = cpa^{m} * cpb^{n} \text{ and}$$
$$ab = cpa^{p} * cpb^{q}$$

where n, m, p and q are integers --, the auxiliary arguments defined as follows are produced

$$ga = aa(cb) + aa(x) \text{ and}$$
$$gb = ab(cb) + ab(x).$$

Verification of participants' authenticity is effected by calculating the verification arguments of the sender and receiver, defined as

$$va = ga(ca) \text{ and}$$
$$vb = gb(cb).$$

Authentication of each participant is verified if its respective verification argument returns null.

In the specific case of the receiver, the verification argument is also used to distort the ciphered message in its current state. This operation consists of applying the following differential operator over the message:

$$V = I + vb\frac{\partial}{\partial_x} . \quad (12)$$

The application of the operator aims at changing the message in case an attacker attempts to personalize one of the participants. In such case, argument vb does not result null, and so operator V does in fact change the message -- if vb is null, V simply becomes the identity operator, thus preserving the message.

8. CONSIDERATIONS ABOUT THE INVERSION PROBLEM

The main difficulties found on the various processes of differential equation resolution are related to the large amount of memory, and the long processing time required to obtain numeric solutions [5][7][9][11]. As for analytical methods, algebraic manipulation of expressions involves two basic difficulties:

1 - the exponential growth of the function string[3]; and

2 - the application of some operators over analytical expressions (such as iterated integrals).

The first difficulty refers to the inefficiency of expression simplification algorithms. In any of the symbolic computation software applications (such as *MAPLE, Mathematica, SimbMath, Derive* and others), the simplification commands are limited to expanding expressions, often generating functions with a greater number of characters than the original expression to be simplified. At present, there are no efficient commercial systems to recognize patterns and regroup algebraic expressions.

The second difficulty is due to the fact that applying certain operators requires solving inverse problems.

From a mathematical viewpoint, the great difficulty found in solving differential equations is the application of inverse operators. For example, suppose that the function

$$f(x,y) = x.\cos(y) + e^{-x^2-3.x.y+2.x-y} \quad (13)$$

has the following differential operator applied to it

$$\frac{\partial^3}{\partial x^3} + 2\frac{\partial^2}{\partial x^2} + \frac{\partial}{\partial x} . \quad (14)$$

Therefore results the expression

$$-5(-2x-3y+2)exp(-x^2-3xy+2x-y)+(-2x-3y+2)^3exp(-x^2-3xy+2x-y) -2x\ cos(y)+2(-3x-1)^2exp(-x^2-3xy+2x-y)+cos(y). \quad (15)$$

It is clearly noticeable that a single application of the operator produces a considerable increase in the size of the resulting expression. Moreover, in most conventional analytical methods, operations of this kind are performed recursively, thus producing excessively long expressions. In order to illustrate the argument, let us say that spectral methods based on symbolic operation require, in average, 1,000 applications of differential operators similar to the one shown above, producing final expressions with 5,000 to 150,000 more characters than the original expression.

On the other hand, manipulation rules of the differential operators obtained through Lie groups for the resolution of partial differential equations significantly reduce the number of symbolic operations required to obtain the final expression, since their **direct** application is dispensed with. The solution to avoid recursive application of differential operators over analytical expressions is based on the use of what is called **Lie symmetries** of the specific

[3] Or expression length

solutions of differential equations.

There are basically two arguments that justify the method as viable: the first one is the high number of symbolic operations required of an attacker to restore the original message from the ciphered message. The second argument refers to the difficulties found in the execution of this process. In the absence of private keys, the attacker must proceed as follows in order to decode the message:

I) infer the form of the differential operator present in the differential equations satisfied by $f_0(x)$.

II) find the system of determining equations used to obtain the variable coefficients present in the infinitesimal generators of the symmetry groups;

III) solve the system obtained by using partial differential equation mapping and solving libraries;

IV) find a unique solution of the differential equation in order to start the mapping process;

V) map the unique solution by using the exponential of a linear combination of the infinitesimal generators obtained;

VI) infer and apply the initial conditions of contour -- or plotting -- that verify the solution as unique;

VII) recover the original message, by applying the ASC command on the ASCII codes of the function obtained.

The construction of the determining equations requires, in average, 250 symbolic operations for partial two-dimensional equations of second degree with variable coefficients. This is because it is necessary to calculate prolongation[4] of Second degree of the infinitesimal generators of the symmetry group and then apply the criterion of infinitesimal variation [15][16]. It is important to point out that the number of symbolic operations required to obtain the determining equations **grows exponentially** with the growing order of the prolongation used -- which is the same order of the differential equation to solve.

The number of symbolic variations required to solve the system of determining equations essentially depends on the coupling degree of the system obtained. In general, the number of symbolic operations required to solve a system of determining equations produced by partial two-dimensional equations of Second degree with variable coefficients is around 500, given an average number of ten determining equations. What happens is that the number of symbolic operations grows with the square of the number of equations, which on its turn grows linearly with the degree of the differential equation to solve.

A unique solution for the differential equation can be obtained in two ways: by using commands that perform direct resolution of the equation in special forms, or by finding the determining equations for the given special form. In the first case, the solution can be obtained immediately, with one single command line. In case direct resolution is not possible, step I) must be taken again, as well as all stages of the course set for the special form of the differential equation.

The solution can also be mapped in three ways: by the use of operator exponential manipulation rules [16], through the use of expansions of these operators in Taylor series, or by solving auxiliary differential equations. In case there are rules available for the manipulation of the exponential of the infinitesimal generators obtained, the number of resulting operations is rather reduced -- often being of the same order as the number of infinitesimal generators[5]. The number of operations required to obtain the expansions of the Taylor series depends solely on the convergence radius of the series, being proportional to the number of terms used. As for the resolution of auxiliary equations -- like the calculation to find the determining equations for the special form set --, that also requires running the algorithm fully.

The application of the initial conditions of contour basically consists of determining arbitrary functions contained in the mapped solution, through the resolution of algebraic or differential equations of first order -- thus having the same problem as in the previous step.

The other steps are essentially numerical and do not require the use of symbolic operations, but only a large number of floating point operations.

Even though the exponential growth of the number of symbolic operations required with the increase of the order of the differential equation is enough to justify the method as viable, there are additional difficulties in running the process presented above, i.e., an attacker's attempt to break down the ciphering. First of all, the very differential equation to be solved is not supplied as a public key -- so that it becomes necessary to infer its structure from a trial and error process. Besides, the initial conditions of contour and plotting are not supplied either. Under the extremely unlikely hypothesis of the attacker finding the equation general solution, it is still necessary to choose between several families of surfaces that represent it, a surface that follows the **contour conditions implicitly applied**. This implies running a new trial and error process, similar to the one applied to determine the form of the differential equation -- therefore, extremely costly. Moreover, there is not any systematic procedure to guide the trial and error process in either case presented. Actually, the solution of auxiliary problems that arise while performing the steps of the algorithm often require the recursive reapplication if the first five steps of the proposed algorithm. This looping procedure significantly increases the number of symbolic operations required, since each recursive application exponentially folds this number of operations.

9. CONSIDERATIONS AND CONCLUSIONS

The use of translation symmetries on the complex plane has three fundamental advantages over algorithms based on the direct formulation of differential equations: the first is that the expressions obtained are simples -- requiring less memory to store --; the second is the high processing speed -- allowing more efficacy in ciphering and deciphering, thanks to the extremely low processing time required to run the corresponding algebraic operations -- and the third and main advantage is the enormous difficulty of solving the associated inverse problem -- i.e., the deciphering process. If the attacker does not have the corresponding private keys, restoring the message becomes a computationally impracticable trial and error process, **even when Lie groups are used to solve the differential equations produced**. Besides, if the user tries to find the real and imaginary components of the private key directly, it will necessarily require a sweeping process on the complex plane without previously knowing the very limits of the corresponding search region. It should be

[4] Prolongation is the extended domain composed of independent variable, unknown functions and their derivatives.

[5] That is, the dimension of the corresponding group of Lie.

noticed that, in methods based on groups of Galois, sweeping would be limited to a subset of integers consisting of prime numbers -- which is a numerable set. The set of complex numbers, on its turn, is a continuous power, so that the scanning process would cover a region of the complex plane with an infinity of elements for each scanning subinterval. Therefore, it suffices to choose any two long integers to represent the real and imaginary parts of the complex number that makes up the private key in order to ensure the privacy of the scheme proposed.

The average processing time[6] required in 40 rounds was two minutes, and the number of characters required to store the ciphered message was approximately four times greater than the original file. In every simulation run, there was not, up to now, any code breaks via algorithms based on Lie algebra. The deciphering attempts consist of using libraries (*liesymm*, *pdetools* e *detools*) of the Maple 5.0 system, which aims at calculating the coefficients of the infinitesimal generators of the group of symmetry of differential equations obtained from translated functions. The main difficulties of the deciphering process are the absence of a differential equation to start the process, as well as the need to use recursive processes that involve the synthesis of determining equation [15][16], and their resolution through the use of available symmetries[13].

Some examples are available in http://www.sinpro-rs.org.br/vinicius.gadis.ribeiro, in download area.

The conclusions presented can be summarized as follows:

- There is a new paradigm that meets the Public Key requisites.
- It is beyond the traditional domain of schemes that use number theory, due to the power of the set of Private Keys involved.

The main advantages of the proposed scheme are the simplicity of the expressions obtained, the high processing speed and the difficulty to solve the inverse problem.

To conduct the experiments, it has been used Maple symbolic processor. One open question to work is floating point problems, when using some programming language like C. Performance is another interest point, when using a programming language. Another one is the data expansion that occurs during translations.

10. REFERENCES

[1] Alfred J.Menezes; Paul C. van Oorschot; Scott A. Vanstone. **Handbook of Applied Cryptography**. Boca Raton: CRC Press, 1996. 786 p.

[2]Bruce Schneier. **Applied Cryptography – Protocols, Algorithms and Source code in C**. 2nd ed. New York: John Wiley, 1994. 624 p.

[3] David Bressoud. **Factorization and primality test**. New York: Springer Verlag, 1989. 246 p.

[4] Donald Greenspan; Vicenzo Casulli. **Numerical Analysis for Applied Mathematics, Science and Engineering**. Redwood City, CA: Addison Wesley, 1988. 346 p.

[6]George Bluman & Sukeiuki Kummei. **Symmetries and differential equations**. New York: Springer-Verlag, 1989.

[5] G. Dattolli et al. Exponential operators, operational rules and evolution problems. **Il Nuovo Cimento**, Bologna, v. 113 B, n. 6, p 699-710, 1998.

[7] Gilbert Strang. **Introduction to Applied Mathematics**. Wellesley: Wesley-Cambridge, 1986. 760 p.

[8] Ian Blake; Gadiel Seroussi; Nigel Smart. **Elliptic Curves in Cryptography**. London: Cambridge Press, 2000. 210 p.

[9] Jame M.Ortega; William G Poole Jr. **Numerical Methods for Differential Equations**. Massachusetts: Pitman, 1981. 334 p.

[10] Joel Brawley; Shuhong Gao. **Mathematical Models in Public-Key Cryptology**. 1999. Available in: *<http://citeseer.nj.nec.com/ /brawley99mathematical.html>*. Access in: Apr 25th, 2001.

[11] Junuthula Narasimha Reddy. **Applied Functional Analysis and Variational Methods in Engineering**. New York: McGraw-Hill, 1986. 550 p.

[12] Leonard M. Adleman; Kevin S. McCurley. Open Problems in Numeric Theoretic Complexity, II. In ANTS-1.[199?] **Proceedings...** Available in: *<http://citeseer.nj.nec.com/168265.html>*. Access in: Feb 12nd, 2001.

[13] M. Bozicevic A Property of Lie Group Orbits. **Canadian Mathematical Bulletin**, Ottawa, v. 43, n. 1, p. 47-50, 2000.

[14] M. J. Daley. **Algorithmic Primality Testing**. 1997. Available in: *<http://citeseer.nj.nec.com/270620.html>*. Access in: Mar 13rd, 2001.

[15] Nail Ibragimov. **CRC Handbook of Lie Group Analisys of Differential Equations**. 2 nd ed. Boca Ratón: CRC, 1995. v. 2.

[16] Peter Olver. **Applications of Lie Groups to Differential Equations**. 2 nd ed.New York: Springer, 2000. 513 p.

[17] Randall K Nichols. **ICSA guide to Cryptography**. New York: McGraw-Hill, 1999. 840 p.

[18] Ron Rivest; Adi Shamir; Len A. Adleman. A method for obtaining digital signatures and public-key cryptosys tems. **Communications of the ACM**, New York, v. 21, n. 2, p. 120-126, 1978. Available in: *<http://citeseer.nj.nec.com/ rivest78method.html>*. Access in: Jun 8th, 2000.

[19] Severino Collier Coutinho. **Mathematics of Ciphers - Number Theory and RSA Cryptography**. Natick, MA: A. K. Peteres, 1998. 190 p.

[20] Theodoulus Garefalakis. **Primality Testing, Integer Factorization and Discrete logarithms**. 1998. Available in: *<http://citeseer.nj.nec.com/garefalakis98primality. html>*.Access in: Mar 13rd, 2001.

[21] Whitfield Diffie; Martin E. Hellman. New directions in cryptography. **IEEE Trans. Inform. Theory**, New York, v. IT-22, n. 6, p. 644-654, 1976. Available in: *<http://citeseer.nj.nec.com/diffie76new.html>*. Access in Apr 25th, 2001.

[22] William Stallings. **Cryptography and network security**: principles and practice. 2nd ed. Upper Saddle River: Prentice-Hall, 1998. 574 p.

[23] Vyjayanthi Chari & Andrew Pressley. **A Guide to Quantum Groups**. Cambridge: Cambridge University Press, 1994. 660 p.

[24] Vinicius G. Ribeiro & Raul Fernando Weber. Problemas Computacionais para esquemas de chave pública In: IV Workshop de Segurança de Sistemas - Wseg. Gramado: II/UFRGS, 2004. p 101-112.

[6] Experiments were run on a Pentium MMX 233 GHz with 128MB RAM.

• The first author would like to thank the sponsorship given by UNILASALLE, ULBRA and FACENSA.