INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2018, VOL. 64, NO. 2, PP. 185-188
Manuscript received January 30, 2017; revised March, 2018. DOI: 10.24425/119368

A first Catalgorithm?

Jean-François Geneste

***Abstract*—We propose building a new PKC in a ring structure, the classification of rings being an open problem. The difficulty of the scheme is based on retrieving the eigenvalues of endomorphism on a finite type module over a non-commutative ring. It is resistant to a chosen cipher text attack. Working in the fraction ring of a non-commutative ring makes our scheme a zero-knowledge proof of knowledge, result indistinguishable, in the Naor-Yung model. Finally, a dramatic improvement in security is obtained through the drawing with uniform probability of the working ring at high frequency.**

***Keywords*—Non commutative rings, Zero-knowledge proofs, finite type module**

I. INTRODUCTION

DURING the Catacrypt'2014 session some important issues were discussed. Among them a clear call for new PKC primitives was made. The fact is that the PKC's at our disposal today mainly are the Diffie-Hellmann scheme for key distribution which, in some way, has been extended into schemes based on elliptic curves and the RSA. It is a fact that all these algorithms are based on group theory for which we basically know two things:

1- The classification of finite groups is a problem which has been definitely solved by mathematicians
2- Many papers of group theory have not been published by governmental organizations so that the crypto community might not be aware about some possible attacks on cryptosystems working in groups (1).

This therefore pleads for proposing new PKC's dealing with other structures than groups. There are several candidates, but we decided to deal with rings as will be explained later on. In this paper we shall describe the algorithm (a PKC), we shall show its advantages and drawbacks and shall justify our choices. In the particular case of the fraction ring of noncommutative rings, we shall show that we can transform the algorithm into a one round zero-knowledge proof of knowledge. Finally, we shall explain how such a point of view can be used to propose a new way of making cryptosystems and in particular we shall insist on the fact that our very algorithm cannot be attacked by a chosen ciphertext attack. We shall also explain why quantum computers, whatever their computation power, cannot break it.

Jean-François Geneste is with the Skolkovo Institute of technology, Moscow, Russia (e-mail: J.geneste@skoltech.ru).

II. DEALING WITH RINGS

The first reason why we propose dealing with rings as a will to escape PKC's running in groups is because the problem of the classification of rings still is an open problem (2). Therefore, the risk of a broken general PKC running in rings is expected to be much lower than for a one running in a group. The only drawback is that in a ring we potentially have a richer structure than in a group and therefore we could expect, in the meanwhile, to be able to perform more computations in a ring than in a group and have something weaker in the end. But this is not what will happen as we shall see in the following.

The second reason why we choose rings is that we aim at choosing it noncommutative because in such a structure even a linear one variable equation displays no efficient algorithm to be solved. Typically, what could be any "intelligent" algorithm to solve the general following equation in a "random" noncommutative ring?

$$ax + xb = c \tag{0.1}$$

There also is an even simpler equation which is not evident to solve either and which is the following.

$$axb = c \tag{0.2}$$

If a and/or b is/are not invertible. In general, there seems to be no escape to trial and error. Now, it can obviously happen that in a particular ring such an instance of the problem is easy to solve. We shall discuss this point later on.

If we want to design a cryptosystem, we must find a way to hide information in a public key. The way we propose to solve this problem is the following. Let us consider a noncommutative finite left ring R. Let us consider a finite type module on R which we call M. Let us begin with the assumption that $\dim_R M = 2$ and let us consider the set of linear applications on M which we call $L(M)$. Let us choose at random two non-zero different elements in R which we call λ_2 and λ_3. Let us then draw at random (i.e. with uniform probability), two non-zero vectors $x, y \in M$ so that they make a basis of M. Let us write these vectors **in column** into a 2×2 matrix which we call P^{-1} and let us compute

$$F_0 = P^{-1}\begin{pmatrix}\lambda_2 & 0\\ 0 & \lambda_3\end{pmatrix}P \qquad (0.3)$$

Clearly, F_0 is a diagonalizable endomorphism of M. Nevertheless, to compute, given F_0, its eigenvalues, at least means solving a system of 2 equations with 2 unknowns. As asserted above, we are not even able to solve a sole equation with one unknown in general in a noncommutative ring. Therefore, we can expect this as being a difficult problem. To be more precise, if $F_0 = \begin{pmatrix}a & b\\ c & d\end{pmatrix}$ we need to solve the following system of equations.

$$\begin{cases} xa + yb = \lambda x \\ xc + yd = \lambda y \end{cases} \qquad (0.4)$$

Where x, y and λ are unknown.

The fact is that if we consider the same problem in a commutative ring, a way to solve it, is to compute the roots of the minimal polynomial. But, if, for example, the ring is *Z/nZ*, then solving the problem is equivalent to factoring. This remark therefore implies that in the general case, we face a (much) harder problem than the one of factoring.
But, at this very point, we only have a clue of the difficult problem we want to deal with in order to get security. We still are lacking an enciphering algorithm.

III. THE BASIC IDEA

Since finding the eigenvalues of an endomorphism is a difficult problem, then finding the eigenvectors also is a difficult problem and we can think about scrambling the message with random vectors. For that, we need the message to be a vector. So, let us consider M now as a module over R of dimension 3. Let us call the message to send $x \in M - \{0_M\}$. Let us build, as before, a matrix which will be our public key

$$F_0 = \begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & a & b\\ 0 & c & d\end{pmatrix} \qquad (0.5)$$

Where $\begin{pmatrix}a & b\\ c & d\end{pmatrix} = P^{-1}\begin{pmatrix}\lambda_2 & 0\\ 0 & \lambda_3\end{pmatrix}P$ and where $\lambda_1 \in R - \{0_R\}$ is chosen at random and different from λ_2 and λ_3.
Let us then choose at random, as before, two non-zero vectors $y, z \in M$ so that $\{x, y, z\}$ is a basis of M. Let us build a 3×3 matrix K^{-1} which is made of the vectors x, y, z **in column** and which is therefore invertible and let us compute

$$F = K^{-1}F_0K \qquad (0.6)$$

Now, just please notice that λ_1 is known, so that we know at least this eigenvalue of F and therefore the clear message is collinear to any eigenvector associated to λ_1. However, such knowledge is not enough to decipher anything[1]. We must therefore add some information. The first idea we had was simply to transfer, together with F, an additional vector which would have been $x + y + z$. But it is straightforward to show that this gives far too much information to an attacker and allows in the end to find the eigenvalues of F. A corrective idea is to play with the ring structure as shown above and multiply vector x by some "not so random" numbers in the ring R. This is what we are going to propose but, before, let us tackle some discussion.

Indeed, at Catacrypt'2014, it was said that we are pretty confident in the hash functions designed by the crypto community. We therefore should target, when possible, to include hash functions in PKC's. In the context where we are, such a fact is paramount in the sense that it is going to reinforce the hard problem we are dealing with and linked with the ring structures by adding operations which are not part of the ring structure itself. Typically, on purely theoretical grounds, since the classification of rings is an open problem, then the classification of rings which are "augmented" with hash functions has not commenced! So, let us consider 2 different one-way hash functions, say h_1 and h_2 both starting in M and providing an image in R. Now, we propose the cipher text being

$$\left(F, h_1(y).x.h_2(z) + y + z\right) \qquad (0.7)$$

Where the dots mean the external multiplication in M.
Deciphering works as follows. Upon reception of the cipher text from Bob, Alice decomposes $h_1(y).x.h_2(z) + y + z$ on the Eigen basis of F which she knows because she knows the

[1] In most cases depending on how far the application from *R* into itself and which associates to x the value λx is from a bijection.

eigenvalues. This gives her trivially y and z, hence $h_1(y)$ and $h_2(z)$. It remains to invert these 2 numbers and get x trivially.

For such an algorithm to work "well", we need the ring has "many" invertible elements so that the probability to get invertible elements for $h_1(y)$ and $h_2(z)$ is high otherwise we would be unable to decipher.

IV. Security

We shall only give heuristic proofs, but the reader should be aware that the general scheme presented here resists a chosen cipher text attack in the Naor Young model (3). Roughly speaking, let us assume an attacker would feed Alice with chosen cipher text in the form (g,v) where g is a 3×3 matrix and v is a vector of M. There are 2 cases. Either g is not similar to F_0 (that means there does not exist any matrix K such that $g = K^{-1}F_0K$). In such a case, Alice does not provide any deciphering of the message. Just please notice that the checking can easily be made in polynomial time. Now, if g is similar to F_0, Alice is going to give the decipherment of v. To some extent, this means that $v = h_1(y)xh_2(z)+y+z$, and, given our assumptions, the best which can occur for the attacker is to get a new matrix g' similar to g under the form

$$g' = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & a & b \\ 0 & c & d \end{pmatrix} \qquad (0.8)$$

And the attacker is also given x. Please notice that we were very kind with the attacker since in reality, he should not get the first line with two zeros in it[2]. The form of g' allows knowing $y+z$. Therefore the attacker has the equation $x = (h_1(y))^{-1}(v-y-z)(h_2(z))^{-1}$ and by the fact that y and z are eigenvectors, x cannot be recovered. Moreover, even if the attacker was able to know $h_1(y)$ and $h_2(z)$, then because of the security of the hash functions, this would not give any information about y and z.

Therefore, in some way, we have at least in this scheme, 3 barriers of security.

V. Refinement

We shall not elaborate much further in this paragraph, but just let us remark that for the sake of simplicity the reader might be aware that we dealt with "traditional" eigenvalues which are in the center of the ring. A refinement would consist in dealing with eigenvalues on the left, say, which are no more the same and which are much harder to find. This complicates things in a non-negligible way.

VI. The choice of the ring

A. General choice

The typical non commutative ring is a ring of matrices (but not only…!). We can easily imagine rings of matrices of matrices and so on. The chance we have with such rings is that the subset of non-invertible elements, that is, of non-invertible matrices is small so that in general most elements chosen at random will be invertible and we shall have no problem in performing the computations above. Apart this constraint of having most elements which should be invertible, we have no other one, which allows choosing among almost all the possible existing rings and therefore out of classification rings.

B. Increasing the computation speed

First of all, at least if we deal with matrices making a specific hardware is very efficient in parallelizing the operations and therefore we can have a very efficient scheme.

Now, if we want to increase the computation speed on a pure software point of view, we need to accelerate the computation of the inverses in the ring. For such a purpose, the best is to consider a fraction ring. The problem is that fraction rings of noncommutative rings do not necessarily exist. For the fraction ring to exist, the ring must possess what is called the left or right Ore condition. The right Ore condition (4) for example says that

$$\forall a,b \in R, \exists a',b' \in R / aa' = bb' \qquad (0.9)$$

Whenever a and b are not divisors of zero. Let us then consider

$$S = \{(a,b), a \in R, b \in R, b \text{ is not a divisor of zero}\} \qquad (0.10)$$

We can define an equivalence relation on S by writing

$$(a,b)\sim(c,d) \leftrightarrow ab' = cd'$$

$$where\ bb' = dd'\ via\ the\ Ore\ condition \qquad (0.11)$$

Then the quotient set $A = S/\sim$ where the operations are defined in the natural way, is the fraction ring of R.

[2] Indeed, giving one vector of the Eigen plane at least gives one bit of information on one of the eigenvectors and we know that in the case of computation taking place in Z/nZ this would be equivalent to be able to factoring in polynomial time [3].

When such a fraction ring exists, it is evident that the inversion is instantaneous since it consists in a swap. The computations therefore become very efficient. Nevertheless, we face a big problem.

C. The problem and its solution

Bob enciphers x and gets $\left(F, h_1(y).x.h_2(z)+y+z\right)$. Alice deciphering gives x. But there is a trick! Indeed, in that case x is an equivalence class because of the construction of the ring of fractions. Mathematically, we have deciphered, but on a computer point of view, we only have a representative of the equivalence class of x and there is no reason for this representative to be the same as the original chosen by Bob! Given the complexity of the computations, there even is a probability of 1 that the obtained representative of the equivalence class that Alice gets is chosen with uniform probability among the elements of this class. In the end, what do we get?

Bob sends a message and Alice gives him back a decipherment. Typically, Bob sends the encryption of x, he gets back x' in the same class as x. He then computes in M $x-x'=0_M$, this latter class having only one element. This typically is a zero-knowledge proof of knowledge which is result indistinguishable in the sense of Fiat Shamir (5).

D. Open problems

Whereas we are very pleased with the properties displayed in §C, we still would like a scheme allowing us deciphering the messages since we want to build a PKC. One idea would be to fix some least significant bits, say, of the coordinates of vector x. The question is: how many such significant bits are necessary to find the good representative in the equivalence class on the one hand and if those bits were to be revealed to an attacker, how much does it weaken the security of the scheme? We do not know the answer to such questions.

Let us see an alternative. Indeed, let us assume we have $x=\begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix}$ and let us assume that $x_i=\frac{a_i}{b_i}$ where b_i has a legal inverse. Now, if Bob sends the $b_i's$ in clear, since all denominators can be any suitable number consistent with the right Ore condition, it seems we do not give any information at all (To Be Confirmed). However, this dramatically lowers down the rate of the scheme since in that case the denominators cannot be part of the enciphered message.

VII. Completing the scheme

Until now, the reader could be not really convinced of what we said because the computations we propose are done in a very specific ring. So let us make a step further by assuming that we are going to choose a ring at random. We shall not describe how to draw such a ring because it is greatly out of the scope of this paper. Nevertheless, this can be done! We therefore have a real improvement of the global security of such a scheme compared with, say, the RSA, since the problem to solve for an attacker may be very different depending on which ring is used by the user he is targeting.

But we can even do better. Indeed, the difficulty of breaking the scheme being attached to a very ring, the best to be protected is to change ring as often as possible. But pseudo-random walks in the class of non-commutative rings are quite easy to program and therefore it suffices to draw a ring at random very often and of course publish a new public key. In our opinion, this greatly prevents from a global attack. In particular, the reader will notice that a chosen ciphertext attack becomes very difficult if not impossible. We can couple this with time slots during which there is a validity of both the ring and the public key like today in fact, but with a frequency of change which can be pretty high. In the end, if we look at a great amount of time with high frequency change in the used ring, not only is a chosen ciphertext attack no more possible, but the only possible algorithm for breaking the global scheme over this time, in today's knowledge when the classification of rings is not accomplished, is only trial and error. This implies resistance to any quantum computer whatever its power. In fact, even with infinite power the right deciphering would appear after a while. However, some contextual analysis would be needed to sort out the right piece of clear text which would be obtained and if we add a scrambler to the initial scheme, then this will prevent such an analysis to be done.

VIII. Generalization

Under the condition that it could be possible, we could imagine other structures as rings, with say, a set, we shall call A, some operations on it, we shall call $\Delta, \nabla, \ldots$, some hash functions such that the global structure $\{A, \Delta, \nabla, \ldots, hash\}$ has enough computing properties for efficient algorithms being built, has poor enough properties for hard problems to easily occur, and the global structure classification has no known solution. Such structures could be good candidates for building PKC's on them and could be used with a high frequency change of the underlying set itself (together with the very operations of course).

Reminding the beginning of cryptography and the context it was described in the famous paper "New directions in cryptography", that is the one of a war, using such strategy to rebuild a system out of a "catacrypt", could be a way to elegantly and efficiently solve the problem.

References

[1] J.J. Quisquater, "*Personal communication*".
[2] T. Y. Lam, "*Lectures on modules and rings*", s.l, Springer, 1999.
[3] J.F. Geneste, "*Séminaire GRECC ENS*", 2002.
[4] —, "*Noncommutative algebra*", s.l, Springer, 1993.
[5] U. Feige, A. Fiat, et A. Shamir, "*Zero-knowledge proofs of indentity*", Journal of Cryptology, pp77-94.