On a key exchange protocol based on Diophantine equations

Noriko Hirata-Kohno, Attila Pethő

***Abstract*—We analyze a recent key exchange protocol proposed by H. Yosh, which is based on the hardness to solve Diophantine equations. In this article, we analyze the protocol and show that the public key is very large. We suggest large families of parameters both in the finite field and in the rational integer cases for which the protocol can be secure.**

I. Introduction

The notion of public key cryptography started with a key exchange protocol [12]. Various protocols have been developed for this purpose, see for example [8], [14]. Hard computational problems lie under these protocols, e.g., factorization into primes of large integers, computation of discrete logarithm, determination of the shortest vector in lattices and decoding of error correcting codes.

D. Hilbert asked in his famous lecture at the second International Congress of Mathematicians in 1900 whether there exists a general procedure which determines the solvability of Diophantine equations. The question was answered 70 years later by Y. Matijasevič, who proved that such an algorithm does not exist [11]. However, the impossibility of a general algorithm does not mean that we cannot solve special equations. There are large classes of Diophantine equations which are algorithmically and numerically solvable, see e.g. [1], [20].

Despite many efforts, finding the solutions to Diophantine equations is usually a hard task. Based on this observation, Lin, Chang and Lee [13] suggested a new public key protocol in 1995. A bit later Cusick showed that this protocol is insecure and it can be broken in polynomial time without solving any Diophantine equations [9]. Although such observations, especially in the case of (non-linear) Diophantine equations of high degree, Yosh [22] proposed a key exchange protocol whose security relies on the hardness to find the solutions to the equations.

We present here a more detailed analysis of the protocol. We show that it can be secure both over finite fields and in the original setting, i.e. over the ring of rational integers. In any case there is a big efficiency bottleneck and indeed the size of the public key is enormous.

It might be true that the theory of cryptography does not profit enough from the theory of Diophantine equation of high degree and vice versa. This is the reason to write these notes.

After the celebrated theorem of Shor [19] that factorization and discrete logarithm can be done with quantum algorithms in polynomial time, there is a big demand to develop new public key protocols. These should be based on problems, which cannot be solved by quantum computers in polynomial time, or at least we should have some evidence. A good overview on such efforts is presented in [3]. We hope that these notes might give a small step toward this direction.

II. The protocol of Harry Yosh

In this section, we describe with minor modifications and generalizations, the key exchange protocol proposed by H. Yosh [22]. Let R be a commutative ring with unity 1. Fix $a \in R$ and $b \in \mathbb{N}$ and for $x \in R$, consider the function

$$T_{a,b}(x) = (x+a)^b.$$

Obviously $T_{a,b}$ is a polynomial map from R to R. Assume that b is chosen such that $T_{a,b}$ is injective, i.e. invertible. Let $f(x_1,\dots,x_m)$, $g(x_1,\dots,x_m) \in R[x_1,\dots,x_m]$.

To exchange a secret key, Alice and Bob perform the following steps:

(i) Alice chooses a polynomial $f(x_1,\dots,x_m) \in R[x_1,\dots,x_m]$ and compute a solution $(r_1,\dots,r_m) \in R^m$ to the Diophantine equation

$$f(x_1,\dots,x_m) = 0.$$

She keeps $(r_1,\dots,r_m)$ secret, but makes f public.

(ii) Bob chooses a polynomial $g(x_1,\dots,x_m) \in R[x_1,\dots,x_m]$ and parameters $a_1,\dots,a_n \in R$ as well as $b_1,\dots,b_n \in \mathbb{N}$ such that T_{a_j,b_j} are invertible for $j = 1,\dots,n$. He computes

$$H(x_1,\dots,x_m) =$$

N. Hirata-Kohno is a professor at the Department of Mathematics, College of Science and Technology, Nihon University, Suruga-dai, Kanda, Chiyoda, Tokyo 101-8308, JAPAN (email:hirata@math.cst.nihon-u.ac.jp).

A. Pethő is a professor at the Department of Computer Science, University of Debrecen, H-4010 Debrecen, P.O. Box 12, HUNGARY (email:Petho.Attila@inf.unideb.hu).

The first author was partially supported by the NEXT Program, GR 087, JSPS for 2010–2013. The second author was partially supported by the JSPS, Grant in aid 21540010 and Invitation Fellowship Program FY2011, L-11514, by the OTKA grant No. 104208 and by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project.

$$= T_{a_n,b_n}(\dots(T_{a_1,b_1}(g(x_1,\dots,x_m)))\dots)$$

and takes an element $h \in H + fR[x_1,\dots,x_n]$.
He keeps $a_1,\dots,a_n,b_1,\dots,b_n$ secret and makes g, h public.

(iii) Knowing g, h Alice computes $s = g(r_1,\dots,r_m)$ and $u = h(r_1,\dots,r_m)$ and sends u to Bob.

(iv) Bob computes $T^{-1}_{a_1,b_1}(\dots(T^{-1}_{a_n,b_n}(u))\dots)$, which is s, the common secret key of Alice and Bob.

For completeness we prove

Proposition 1. *The protocol is correct.*

Proof: Alice can compute s because she knows g and $r_1,\dots,r_m$.
As $f(r_1,\dots,r_m) = 0$ we have

$$u = h(r_1,\dots,r_m) = H(r_1,\dots,r_m).$$

Thus

$$s = H^{-1}(u) = T^{-1}_{a_1,b_1}(\dots(T^{-1}_{a_n,b_n}(u))\dots)$$

and Bob can compute s because he knows $a_1,\dots,a_n,b_1,\dots,b_n$ and $T_{a_j,b_j}, j = 1,\dots,n$ are invertible. ■

In Yosh' analysis, it was only considered one possible attack. The secret key can be computed from common solutions to the system of public equations $f = 0, h = u$. Yosh pointed out that one can choose these equations such that the determination via Gröbner bases technique of the common solution still remains a hard task. Unfortunately only few examples were given in the article.

Here, we present a more detailed cryptoanalysis of the protocol of Yosh. In Yosh's original version, only the case $R = \mathbb{Z}$ was investigated and the finite field case was just mentioned. We investigate two cases, when $R = \mathbb{Z}$ and R is a finite field.

Another difference is that Yosh dealt with the map in three parameters $\hat{T}_{a,b,c}(x) = (x+a)^b + c$, with $a, c \in R$ and $b \in \mathbb{N}$. By the obvious identity

$$\hat{T}_{\hat{a}_n,\hat{b}_n,\hat{c}_n}(\dots(\hat{T}_{\hat{a}_1,\hat{b}_1,\hat{c}_1}(x))\dots) =$$
$$= T_{a_{n+1},b_{n+1}}(T_{a_n,b_n}(\dots(T_{a_1,b_1}(x))\dots)),$$

where $a_1 = \hat{a}_1, a_j = \hat{a}_j + \hat{c}_{j-1}, j = 2\dots,n, a_{n+1} = \hat{c}_n$, $b_j = \hat{b}_j, j = 1,\dots,n$ and $b_{n+1} = 1$ it is enough to work with our map in two parameters.

We point out that the most serious bottleneck is the size of the public key, especially the size of h. To keep this parameter in an acceptable size, we have to use low degree polynomials, in particular $b_1,\dots,b_n$ have to be small.

Another important observation is that the equation $f = 0$ has to be hard to solve. We show in both cases that this can be achieved with large families of polynomials. In the case of $\mathbb{Z}$ we present a concrete example for which the protocol seems to be secure and the public key can be computed within some seconds.

A nice feature of the above algorithm is that the parties are coequal during the key generation, both have own secret, which are not known even by the partner. In this respect it is similar to the celebrated Diffie-Hellmann key exchange protocol [12].

III. Preliminary observations

Remark that in [22] there is no hints for the secure choice of the parameters, only an example and remarks about possible attacks are given. In these notes we concentrate on the possibility of such a choice of the parameters, which is computationally feasible, but seems secure enough. In this part we collected observations, which are independent from the ground ring R.

To break the system, i.e. to compute the common key, the enemy has to find the secret parameters $r_1,\dots,r_m$ or $a_1,\dots,a_n,b_1,\dots,b_n$. The only public information about the former is that $(r_1,\dots,r_m)$ is a solution to the system of equations

$$f(x_1,\dots,x_m) = 0 \quad (1)$$
$$h(x_1,\dots,x_m) = u. \quad (2)$$

To solve such equations one can use Gröbner bases technique [5], [6], [8] or elimination theory. The latter means that choosing one of the unknowns, say x_m, one computes the resultant $Res_{x_m}(f, h-u)$, which has unknowns one less than those of f or h. Moreover the first $m-1$ coordinates of solutions to (1) and (2) are zeroes of the resultant. Thus m has to be at least three because otherwise after the elimination one of the variables in (1) and (2), we would obtain an equation in a univariate polynomial, which is simple to solve.

Key exchange protocols are used several times with the same parameters. In our case f and $(r_1,\dots,r_m)$ can be fixed. After each running the enemy learn a new h and the corresponding u. After ℓ turns he collects $\ell + 1$ public equations for $(r_1,\dots,r_m)$. If $\ell \geq m-2$ then the enemy can easily compute $(r_1,\dots,r_m)$.

Proposition 2. *The protocol can be used with the same polynomial f only at most $m-3$-times.*

A further observation of similar manner is the following.

Proposition 3. *If the adversary can compute many solutions, not necessarily $(r_1,\dots,r_m)$, of (1), then he can compute the element s and break the protocol.*

Proof: Indeed, assume that $(\alpha_1,\dots,\alpha_m) \in R^m$ is a solution to (1) and put $\beta = g(\alpha_1,\dots,\alpha_m)$. As

$$h = H + fV$$

for some $V \in R[x_1, \ldots, x_m]$, we have $h(\alpha_1, \ldots, \alpha_m) = H(\alpha_1, \ldots, \alpha_m)$. Thus we get the equation

$$(((\beta + a_1)^{b_1} + a_2)^{b_2} + \ldots + a_n)^{b_n} = h(\alpha_1, \ldots, \alpha_m). \tag{3}$$

for $a_1, \ldots, a_n, b_1, \ldots, b_n$. Knowing about $2n$ solutions of (1) we obtain about $2n$ equations of form (3), which determine usually the $2n$ unknowns. ■

Now we investigate the possible choice of $a_1, \ldots, a_n, b_1, \ldots, b_n$. Let

$$t(x) = t_{a_1,\ldots,a_n,b_1,\ldots,b_n}(x) =$$
$$= T_{a_n,b_n}(\ldots(T_{a_1,b_1}(x))\ldots) =$$
$$= (((x + a_1)^{b_1} + a_2)^{b_2} + \ldots + a_n)^{b_n}.$$

It is clear that the degree of $t(x)$ is $b_1 \cdots b_n$. On the other hand its value at each point can be computed by n additions and by at most $O(\log b_1 + \ldots + \log b_n)$ multiplications. Furthermore, it can be stored on at most $n(A + B)$ bits, where A and B denote the maximal bit length of the representations of a_i and $b_i, i = 1, \ldots, n$ respectively. This means that t admits a very sparse representation. Since polynomials in sparse representations are rare, we cannot expect that h has a similar simple representation. We have to expect that the representation of h is dense, i.e. most of its coefficients are non-zero.

Put $d_i = \deg_{x_i} g, i = 1, \ldots, m$. Then it is clear that

$$\deg_{x_i} H = b_1 \cdots b_n \cdot d_i$$

holds for $i = 1, \ldots, m$. Thus H has at most $(1 + o(1))d_1 \cdots d_m(b_1 \cdots b_n)^m$ terms. We obtain h in Step (ii) by adding a suitable multiple of f to H. Hence we can control the degree of one of the variables. We may assume that it is x_m. By the argument above, we expect that a big portion of the coefficients of the terms of h is non-zero, i.e. we have to store about

$$O(d_1 \cdots d_{m-1}(b_1 \cdots b_n)^{m-1}) \tag{4}$$

non-zero elements of R. This means that $n, m, b_1, \ldots, b_n$ have to be small. To be more specific $b_1, \ldots, b_n \leq \mathcal{B}$ and $n, m \leq N$, where $\mathcal{B}, N$ are small positive integers.

IV. The protocol over finite fields

Yosh mentioned in [22] that the protocol works over finite fields too, but no detail is given. We analyze this case in the present section. Set $R = \mathbb{F}_q$, where q is a prime power. In practice q is either a large prime or a large power of 2. It is a classical fact that $x \mapsto x^b$ is bijective on $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$ iff $\gcd(q-1, b) = 1$. Combining this fact with the general remarks of Section III we must have $1 \leq b_i \leq \mathcal{B}$ and $\gcd(q-1, b_i) = 1, i = 1, \ldots, n$.

By Proposition 3 the equation $f(x_1, \ldots, x_m) = 0$ has to be hard to solve. The next theorem, which is the combination of Theorem 2.1. and Corollary 2.2. The argument by Bérczes, Folláth and Pethő in [4], enables us to define a large class of $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ such that if q is large then this holds with high probability.

Theorem 1. *Let*

$$F(x_1, \ldots, x_m) := B(x_1, \ldots, x_m) + A(x_1, \ldots, x_m)$$
$$\in \mathbb{F}_q[x_1, \ldots, x_m]$$

with homogeneous polynomials A, B satisfying $\deg A < \deg B = D$, $\deg_{x_i} B = D$ for each $1 \leq i \leq m$. Further, suppose that there exist indices $1 \leq j_1 < j_2 \leq n$ such that the binary form

$$B(0, \ldots, 0, x_{j_1}, 0, \ldots, 0, x_{j_2}, 0, \ldots, 0) \tag{5}$$

has no multiple zero.
Denote by $P_{coll}(F, \gamma)$ the probability that $F(\boldsymbol{x})$ assumes the value $\gamma \in \mathbb{F}_q^*$, when $\boldsymbol{x}$ runs uniformly through the elements of $\mathbb{F}_q^m$. If $q > 5 \cdot D^{13/3}$, then

$$P_{coll}(F, \gamma) \leq \frac{3}{q}.$$

The following construction of f is based on the consequence of Theorem 1.

- Set $q = 2^{127}$, which ensures that $\gcd(q-1, p) = 1$ for $p = 3, 5, 7$.
- Choose homogenous polynomials $A, B \in \mathbb{F}_q[x_1, \ldots, x_m]$ subject to the condition (5) and such that $\deg A < \deg B \sim b_1 \cdots b_n/3$.
- Pick randomly $r_1, \ldots, r_m \in \mathbb{F}_q$ and set $\gamma = B(r_1, \ldots, r_m) + A(r_1, \ldots, r_m)$. If $\gamma = 0$ then choose $r_1, \ldots, r_m$ again, otherwise set $f = B + A - \gamma$.

Then $(r_1, \ldots, r_m)$ is a solution of $f = 0$. As $D \sim b_1 \cdots b_n/3 \sim 7$ the condition $q > 5 \cdot D^{13/3}$ holds too. By Theorem 1 the chance to find $(r_1, \ldots, r_m)$ or a different solution of $f = 0$ is extremely low.

Remark that in the first step q can be replaced by a larger power of 2 or by an odd prime of similar size. We have to be care to the condition $\gcd(q-1, p) = 1$ for all primes $p \leq \mathcal{B}$. In [4] it was proved that there exists a large class of polynomials, which satisfy the assumptions of step 2.

We suggest that Bob chooses $a_1, \ldots, a_n \in \mathbb{F}_q^*$ randomly. This is appropriate because in Step (iii) of the algorithm Alice makes public the value $u = h(r_1, \ldots, r_m)$. Thus the equation

$$(((s + a_1)^{b_1} + a_2)^{b_2} + \ldots + a_n)^{b_n} = u$$

is known for everybody, but the element s is not known. We may assume without loss of generality $b_n = 1$ because one can compute small degree roots in finite fields or in $\mathbb{Z}$ in probabilistic polynomial time. Thus our equation has the form

$$x^b + y = c,$$

where c and b are known, but x, y are unknown elements of $\mathbb{F}_q$. Thus the adversary has no chance to find the hidden solution s.

To hide H we suggest to choose $V \in \mathbb{F}_q[x_1, \ldots, x_m]$ randomly of low degree, and put $h = H + fV$.

Proposition 4. *With the above choice the key exchange protocol of Yosh over finite fields is secure.*

V. The case $R = \mathbb{Z}$

The map $T_{a,b}$ is injective if and only if b is odd.

In Step (iii) of the algorithm, Alice make public the value $u = h(r_1, \ldots, r_m)$. Thus the equation

$$(((s + a_1)^{b_1} + a_2)^{b_2} + \ldots + a_n)^{b_n} = u. \qquad (6)$$

is known for everybody, but s is not known. We pointed out in the finite field case that $b_n = 1$ can be assumed without loss of generality. Thus our equation has the form

$$x^b + y = c,$$

where c is a known integer, b may be assumed to have some small values and x, y are unknown integers. Let y_0 be the nearest integer to $c^{1/b}$ and compute the two sided sequence $(y_0 \pm k)^b, k = 0, 1, \ldots$ until c appears. If the equation has a small solution in y, say $|y| \leq 10^7$, then with the above procedure, it will be quickly found.

Proposition 5. *We may assume $b_n = 1$. The parameters $a_1, \ldots, a_n$ should be sufficiently large, say $|a_i| \geq 10^8, i = 1, \ldots, n$.*

Let $a = \max\{|a_1|, \ldots, |a_n|\}$. We have to expect that the absolute value most of the coefficients of $t(x)$ hence of H, h are as large as $a^{b_1 \cdots b_{n-1}}$, which is 10^{72} even for the smallest possible parameter values $n = 4, b_1 = b_2 = b_3 = 3$. By (4), we have to store and transmit $3^9 \cdot d_1 \ldots d_{m-1}$ integers. In the simplest case, namely choosing g to be linear, we have to transmit about 10^4 coefficients of size 10^{72}. This is a very large amount of data. Below we give a concrete example showing this fact.

Now we come back to the choice of f. By Proposition 3 f has to be such that the equation $f = 0$ is hard to solve. We suggest to choose f a diagonal polynomial, i.e. of form $c_1 x_1^{d_1} + \ldots + c_m x_m^{d_m} - c_{m+1}$ with $d_1, \ldots, d_m \geq 2$. First of all these polynomials are very simple. It is an important aspect to compute h and one solution of the equation $f = 0$.

On the other hand diagonal polynomials are complicated enough, i.e. by careful choice of $c_1, \ldots, c_{m+1}, d_1, \ldots, d_m$ the adversary can hardly find a solution of the diophantine equation $c_1 x_1^{d_1} + \ldots + c_m x_m^{d_m} - c_{m+1} = 0$. Indeed, it is well known that if at most one exponent is equal to two and we fix the values of $m-2$ variables, then the resulting single equation in two-variables has only finitely many solutions. Moreover it is usually hard to find a solution provided the coefficients are large. If two exponents are equal to 2 then we may get equations of form $x^2 - dy^2 = m$ with infinitely many integer solutions, but the computation of the fundamental solutions is hard. For example, it is well known that finding a solution of $x^2 - y^2 = n$ such that $x - y \neq \pm 1, \pm n$ is equivalent to finding a non-trivial factor of n, see e.g. [17].

Choose $d_1 \leq \ldots \leq d_m$ according to the last paragraph and such that they are small, say $d_i \leq 7, i = 1, \ldots, m$. Let v be a positive integer, which we specify later. After fixing $d_1, \ldots, d_m$ it is not wise to choose $c_1, \ldots, c_m$ and c_{m+1}, because the success probability for the solution of a given equation is the same for everybody. Alice has to carry out in a different manner. She chooses a solution and after this she searches for an equation with the prescribed solution. To be more specific, she chooses $r_1, \ldots, r_m, c_{m+1} \in \mathbb{Z}$ randomly subject to the conditions $|r_i|^{d_i} \leq 2^v, i = 1, \ldots, m, |c_{m+1}| \leq 2^v$ and such that $\gcd(r_1, \ldots, r_m) = 1$. The number of possibilities is about $2^{v\left(1 + \frac{1}{d_1} + \ldots + \frac{1}{d_m}\right)}$. Then she computes $c_1, \ldots, c_m$ by solving the linear Diophantine equation

$$c_{m+1} = c_1 r_1^{d_1} + \ldots + c_m r_m^{d_m}.$$

The assumptions are such that this equation is solvable and that it has infinitely many solutions. From this infinite collection we suggest to choose $c_1, \ldots, c_m$ such that they have similar size. Performing this process Alice has the polynomial f and knows a solution to (1). On the other hand, finding a solution for other peoples (or finding another solution for Alice) is hopeless.

It remains to specify v. It must be so large that a brute force attack is hopeless. This means that the number of choices of the parameters must be large, at least 2^{128}. This implies the inequality

$$v\left(1 + \frac{1}{d_1} + \ldots + \frac{1}{d_m}\right) \geq 128.$$

We suggest to choose g randomly among the quadratic or linear polynomials.

There is no canonical choice for $h \in H + f\mathbb{Z}[x_1, \ldots, x_m]$, provided $m > 1$. One can fix a variable, say x_m, and consider H, f as polynomials in x_m with coefficients in the ring $\mathbb{Z}[x_1, \ldots, x_{m-1}]$. Then one can compute the remainder of H modulo f. The choice of the variable considerably influences the size of h. We give an example below. Another possibility for the choice of h is that we pick a polynomial $V \in \mathbb{Z}[x_1, \ldots, x_m]$ randomly and put $h = H + fV$.

Finally we present a concrete example, which might satisfy the security requirements and the size of the public key is beyond the possibilities.[1] Set $m = 4, n = 3$ and choose the polynomials as follows.

$$\begin{aligned}
f &= c_1x_1^2 + c_2x_2^5 + c_3x_3^3 + c_4x_4^7 + c_5;\\
c_1 &= 1004439616068996251566977588899652\\
&\quad 58647,\\
c_2 &= -3498105123011851201811794864519944\\
&\quad 7959092\\
c_3 &= 36379686253405252442775297079115999\\
&\quad 38738364717062704444171396361954364,\\
c_4 &= -70754124560273954620402107149399 5\\
&\quad 810881751202074223992649824 2401,\\
c_5 &= -98765432345678987654321654320567 8\\
&\quad 96543210567,\\
g &= 3x_1 + 5x_2^2 + 7x_1x_2 + 93x_3^3 + 753x_4,\\
H &= ((g + 734367)^3 + 537769)^5 + 56478587.
\end{aligned}$$

A solution of $f = 0$ is

$$\begin{aligned}
x_1 &= 2354524623523531215 12,\ x_2 = 43689743,\\
x_3 &= 43216789765432,\ x_4 = 4567973.
\end{aligned}$$

We left to the readers to find a different solution. With these parameters the computation of h took some seconds. It has 2107 terms and the internal representation in MAPLE has length 800327.

References

[1] A. Baker, *Transcendental Number Theory*, Cambridge Univ. Press, Cambridge, (1975).

[2] Th. Becker and V. Weispfenning in cooperation with H. Kredel, *Gröbner Bases: a Computational Approach to Commutative Algebra,* Graduate Texts in Mathematics, Vol. 141, Springer Verlag, New York, (1993).

[3] D.J. Bernstein, J. Buchmann and E. Dahmen (Editors), *Post-Quantum Cryptography*, Springer Verlag, Berlin, Heidelberg, (2009).

[4] A. Bérczes, J. Folláth and A. Pethő, *On a family of collision-free functions*, Tatra Mountains Math. Publ. **47** (2010), 1–13.

[5] B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems,* Aequ. Math. **4** (1970), 374–383.

[6] B. Buchberger, G. E. Collins and R. Loos eds., in cooperation with R. Albrecht, *Computer Algebra Symbolic and Algebraic Computation,* Springer, (1982).

[7] J. Buchmann, *Introduction to cryptography*, Second edition. Undergraduate Texts in Mathematics. Springer, (2004).

[8] J. Buchmann and H.C. Williams, *A key-exchange system based on imaginary quadratic fields*. J. Cryptology **1**, (1988), 107–118.

[9] T.W. Cusick, *Cryptoanalysis of a public key system based on diophantine equations,* Inform. Processing Letters, **56** (1995), 73–75.

[10] J. H. Davenport, Y. Siret and E. Tournier, *Computer Algebra Systems and Algorithms for Algebraic Computation,* Academic Press, (1988).

[11] M. Davis, Y. Matijasevic and J. Robinson, *Hilbert's tenth problem, Diophantine equations: positive aspects of a negative solution,* in: *Mathematical Developments Arising from Hilbert Problems*, Ed.: F.E. Browder, Symp. in Pure Math., (1974), AMS, Providence, RI., (1976), pp. 323–378.

[12] W. Diffie and M. Hellman, *New direction in cryptography,* IEEE Trans. onInformation Theory, 22 (1976), 644–654.

[13] C. H. Lin, C. C. Chang and R. C. T. Lee, *A New Public-Key Cipher System Based Upon the Diophantine Equations*, IEEE Transactions on Computers, Volume 44, Issue 1, (1995).

[14] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of applied cryptography*, CRC, (1996).

[15] M. Mignotte, *Mathematics for Computer Algebra,* Springer Verlag, Berlin, 1992.

[16] R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signature and public-key cryptosystems*, Communications of the ACM, 21 (1978), 120–126.

[17] H. Riesel, *Prime numbers and computer methods for factorization*, Second edition. Progress in Math., 126. Birkhäuser, MA, (1994), xvi+464 pp.

[18] B. Schneier, *Applied cryptography: protocols, algorithms and source code in C,* (1996).

[19] P. Shor, *Algorithms for Quantum Computation: Discrete Logarithm and Factoring,* Proc. 35th Annual Symposium on Foundations of Computer Science (1994) 124–134 and SIAM J. Comput. 26 (1997), 1484–1509.

[20] N.P. Smart, *The algorithmic resolution of Diophantine equations*, London Mathematical Society Student Texts, 41. Cambridge University Press, Cambridge, 1998.

[21] F. Winkler, *Polynomial Algorithms in Computer Algebra*, Texts and Monographs in Symbolic Computation, Springer, (1996).

[22] H. Yosh, *The key exchange cryptosystem used with higher order Diophantine equations*, International Journal of Network Security & Its Applications **3** (2011), 43–50.

Noriko Hirata-Kohno is a professor at the Department of Mathematics, College of Science and Technology, Nihon University, Suruga-dai, Kanda, Chiyoda, Tokyo 101-8308, JAPAN. Her research interests include Diophantine problems and applications to cryptography. She has a PhD from University of Paris 6. Her email address is hirata@math.cst.nihon-u.ac.jp.

Attila Pethő is a professor and the head of the Department of Computer Science, Faculty of Informatics, University of Debrecen, Hungary. He got PhD degree in mathematics from the Lajos Kossuth University. He is a corresponding member of the Hungarian Academy of Sciences. His research interest are number theory and cryptography. You can contact him: petho.attila@inf.unideb.hu.

[1] It is not at all practical.