AAECC (2021) 32:81–96
https://doi.org/10.1007/s00200-019-00402-1

ORIGINAL PAPER

Symmetric and asymmetric cryptographic key exchange protocols in the octonion algebra

Z. Lipiński[1]

Received: 20 August 2019 / Accepted: 22 October 2019 / Published online: 2 November 2019

Abstract
We propose three cryptographic key exchange protocols in the octonion algebra. Using the totient function, defined for integral octonions, we generalize the RSA public-key cryptosystem to the octonion arithmetics. The two proposed symmetric cryptographic key exchange protocols are based on the automorphism and the derivation of the octonion algebra.

Keywords Non-associative cryptography · Octonion cryptography · Octavian totient function · Octonion RSA algorithm · Quaternion cryptography

Mathematics Subject Classification 68P25 · 94A60 · 11T71

1 Introduction

The development on non-commutative cryptography has its origin in the solutions of the three famous problems in combinatorial group theory proposed by Dehn [1] and Miller [2] . In 1954 Novikov constructed a finitely presented group for which the conjugacy problem is unsolvable, [3]. In 1955, Novikov and Boone independently showed that there are finite group presentations whose word problem is undecidable, [4–8]. In [9] Wagner and Magyarik devised the first public-key protocol based on the unsolvability of the word problem for finitely presented groups. A non-deterministic public-key cryptosystem based on the conjugacy problem on braid group, similar to the Diffie–Hellman key exchange system, was proposed in [10]. The most interesting non-commutative key agreement cryptographic systems were proposed by Anshel, Anshel and Goldfeld (AAG) in [11,12]. To construct a key agreement cryptographic system the authors used the braid groups for which the best known algorithm to solve the conjugacy problem requires at least exponential running time. A natural extension

✉ Z. Lipiński
zlipinski@uni.opole.pl

[1] University of Opole, Opole, Poland

of the non-commutative cryptography is the cryptography on the non-associative algebraic structures. The earliest quasigroup-based public-key cryptosystem was proposed by Koscielny and Mullen, [13]. In [14] the AAG PKC was generalized to the non-associative algebraic structures, called the left self-distributive (LD) systems, [15,16]. In [14] Kalka used the LD-systems to define the non-associative public-key cryptographic protocol.

In this article we propose three cryptographic key exchange protocols based on the octonion algebra, [17]. The first protocol is the generalization of the RSA algorithm to the octonion arithmetic, [18]. The another two, symmetric cryptographic key exchange protocols, are based on the automorphism and the derivation of the octonion algebra.

We denote by $\{e_i\}_{i=0}^{7}$ the basis of the octonion algebra O. The multiplication of the octonions can be described using the set of directed lines

$$\mathrm{L} = \{013, 045, 352, 346, 260, 241, 156\}$$

in the Fano plane, [19]. Each line $ijk \in \mathrm{L}$ contains three points i, j, k which represents three octonions with the multiplication $e_i e_j = e_k$. For the octonion unit $e_7 = 1$ the multiplication in O can be defined by the following relations

$$e_i e_j = -\delta_{i,j} e_7 + \epsilon_{ijk} e_k, \;\; i, j \in [0, 6],$$

where ϵ_{ijk} is an antisymmetric tensor such, that $\epsilon_{ijk} = 1$ if $ijk \in \mathrm{L}$ and $\delta_{i,j}$ is the Kronecker delta. The octonions O form the non-associative, normed division algebra, [17]. The non-associativity of three elements x, y, z from O can be expressed by an associator $(x, y, z) = (xy)z - x(yz)$. It is linear in each of its three variables and vanishes whenever two of its variables are equal, i.e., it is an alternating function, [20]. By linearizing the alternative laws one can show that the associator is skew symmetric, i.e., it changes the sign whenever two of its variables are interchanged.

Any element $x \in \mathrm{O}$ satisfies the minimal polynomial

$$x^2 - 2\,\mathrm{Re}(x)\, x + \mathrm{N}(x) = 0, \tag{1}$$

where $\mathrm{N}(x) = \sum_{i=0}^{7} x_i^2$ is the norm of the octonion x and $\mathrm{Re}(x) = x_7$. The octonion is integer when the trace $\mathrm{tr}(x) = 2\mathrm{Re}(x) \in \mathbf{Z}$ and $\mathrm{N}(x) \in \mathbf{Z}$. We denote by e_{ijkl} the octonion $\frac{1}{2}(e_i + e_j + e_k + e_l)$. On can easily check that the product of the two integral octonions e_{0235} and e_{7235} is non-integral, i.e., $\mathrm{tr}(e_{0235}\ e_{7235}) = -\frac{3}{2}$. From this follows, that the set of octonion integers does not form a ring, and it is necessary to consider subrings of integers, called the orders, [21,22]. There are sixteen orders of integral octonions containing $\mathrm{O}(\mathbf{Z})$, [17]. Among them there are seven isomorphic maximal orders (called the octonion arithmetics). By a maximal order we mean an order which is not contained in any other order. To construct a maximal closed under multiplication set of integral octonions (the octionion arithmetic) we use two sets L_0 and Q

$$\mathrm{L}_0 = \{0137, 0457, 3527, 3467, 2607, 2417, 1567\},$$

$$\mathrm{Q} = \{2456, 6123, 4601, 5012, 1345, 3560, 0234\},$$

where Q is the complements of the lines L on the Fano plane. If we interchange in L_0 and Q the elements 0 and 7 we obtain the Coxeter–Dickson 0-sets (0-integers). The 0-integers are spanned by the following halving-sets of octonions, [17].

$$e_{0137}, e_{0457}, \underline{e_{3520}}, \underline{e_{3460}}, e_{2607}, e_{2410}, \underline{e_{1560}},$$
$$e_{2456}, e_{6123}, e_{4671}, e_{5712}, e_{1345}, \underline{e_{3567}}, e_{7234}, \Omega, \emptyset,$$

where Ω is the set of Kleinian octonions generated by the element $\frac{1}{2}\sum_{i=0}^{7} e_i$, and the empty set $\emptyset$ generated by $\sum_{i=0}^{7} e_i$, called the Graves integers, i.e., the octonions of the form $x = \sum_{i=0}^{7} x_i e_i$, $x_i \in \mathbf{Z}$. The underlined octonions are the generators of the 0-sets over the Gravesian integers. The octonion arithmetic generated by the 0-sets we denote by $\mathbf{O}$ and call the integral octonions. Two octonions are congruent $\mod m$ provided that their difference is m times an octonion integer [22,23].

$$x = y \bmod m \;\Leftrightarrow\; x - y = mz, \quad m \in \mathbf{Z}, x, y, z \in \mathbf{O}.$$

The integral octonion x is invertible $\mod m$ if its norm $\mathrm{N}(x)$ is coprime to m, i.e., $\gcd(\mathrm{N}(x), m) = 1$. We denote by $\mathbf{O}_m^I$ all invertible octonions $\mod m$. The totient function $\lambda(x, m)$ for an (invertible) integral octonion $x \in \mathbf{O}_m^I$ we define as

$$\lambda(x, m) = \min_{\lambda}\{\lambda \in \mathbf{N} : x^{\lambda} - 1 = m\ y \bmod m,$$
$$x, y \in \mathbf{O},\ \gcd(\mathrm{N}(x), m) = 1\}.$$

The totient function $\lambda(x, m)$ can be defined equivalently by means of the $\widetilde{\lambda}(x, m)$ function over arbitrary octonions from $\mathbf{O}_m$

$$\widetilde{\lambda}(x, m) = \min_{\lambda}\{\lambda \in \mathbf{N} : x^{\lambda} - x = m\ y \bmod m, \quad x, y \in \mathbf{O}\}.$$

For invertible octonions $x \in \mathbf{O}_m^I$ we have $\lambda(x, m) = \widetilde{\lambda}(x, m) - 1$. The orbit of an element $a \in \mathrm{O}$ under the adjoint action of the algebra O is the set

$$\mathrm{Orb}(a, \mathrm{O}) = \{xax^{-1} : \forall x \in \mathrm{O}\}.$$

In the following lemma we prove, an important for further applications, property of $\lambda(x, m)$.

Lemma 1 *The totient function is constant on the orbit* $\mathrm{Orb}(x, \mathbf{O}_m^I)$ *of the adjoint action* $\mathbf{O}_m^I$ *onto itself, i.e.,*

$$\lambda(xax^{-1}, m) = \lambda(a, m), \quad a, x \in \mathbf{O}_m^I. \tag{2}$$

Proof From the alternativity of the associator it follows, that $(a, x, a) = 0$. Because $(x, y, z) = (x, y, \mathrm{Im}(z))$ then we have

$$(a, x, a^{-1}) = \frac{1}{\mathrm{N}(a)}(a, x, a^{*}) = -\frac{1}{\mathrm{N}(a)}(\mathrm{Im}(a), x, \mathrm{Im}(a)) = 0.$$

The above formula means that $(ax)a^{-1} = a(xa^{-1})$. From the polynomial equation (1) it follows that $(axa^{-1})^2 = ax^2a^{-1}$ and the formula (2). □

In the next section, based on the property (2) of the totient function we define the public-key agrement cryptographic protocol in the octonion arithmetic $\mathbf{O}$.

2 The octonionic public-key cryptosystem

By k we denote an integral octonion $\mathbf{O}_m^I$ with the known totient function $\lambda(k, m)$. A plain text for encryption we will represent by an integral octonion $u = (u_0, \ldots, u_7) \in \mathbf{O}_m^I$. From Lemma 1 it follows, that any integral octonion of the form uku^{-1}, where $u \in \mathbf{O}_m^I$, has the same totient function $\lambda(uku^{-1}, m) = \lambda(k, m)$. We use k to enocde the plain text u into the octonion $u_k = uku^{-1}$. Let e be a number from the interval $[1, \lambda(k, m))$ that is coprime to $\lambda(k, m)$, i.e., $\gcd(e, \lambda(k, m)) = 1 \mod m$. By d we denote the number inverse to $e \mod \lambda(k, m)$, i.e., $e \cdot d = 1 \mod \lambda(k, m)$. To encrypt the encoded text in the octonion u_k we calculate the expression $C(u_k) = u_k^e \mod m$. The decryption is the operation of taking the cipher text octonion $C(u_k)$ to the power $d \mod m$, i.e., $u_k = C(u_k)^d = u_k^{e \cdot d} \mod m$. To encode the plain text u from u_k, the recipient has to solve the linear equation $uk = u_k u \mod m$ in $\mathbf{O}_m$. In general, a solution of this equation is not unique. To allow the recipient of the cipher $C(u_k)$ uniquely recover the encrypted text one can attached to the cipher a hash value of the encoded octonion or provide to the recipient three parameters which allow to solve the equation uniquely. The general equation $xa = bx \mod m$ for the x variable, where $x, a, b \in \mathbf{O}_m$ and $m \in \mathbf{N}$ is equivalent to

$$xa = bx + my, \quad x, y, a, b \in \mathbf{O}, \quad m \in \mathbf{N}. \tag{3}$$

Solution of the homogeneous equation

$$xa = bx, \quad a, b, x \in \mathrm{O} \tag{4}$$

over the real octonion algebra O exists when $\mathrm{N}(a) = \mathrm{N}(b)$ and $\mathrm{tr}(a) = \mathrm{tr}(b)$ $(a_7 = b_7)$. From the vanishing of the associator $(xax^{-1}, x, a) = 0$ follows, that any solution x_0 of (4) satisfies $(b, x_0, a) = 0$ and that $x_0 a$ is also a solution of (4). Let us take as the solutions of (4) $x_1 = \bar{b} + \bar{a} - 2a_7$ and $x_1 a$, then the general solution of (4) we can write as

$$x_0 = \lambda_1 x_1 + \lambda_2 x_2, \quad \lambda_1, \lambda_2 \in \mathrm{R}, \tag{5}$$

where $x_2 = x_1 a = \bar{b}\bar{a} - \mathrm{N}(a)$. To solve the Eq. (4) we need to fix two parameters. To obtain a unique solution of the congruence equation (3) we need to know three parameters. The recipient should obtain these three parameters from the sender. For example, the first can be the parameter in u which is prescribed to ensure that $\gcd(N(u), m) = 1$ and it is not a part of the encryption text. The next two parameters, can be obtained by the recipient from the decrypted block of data from the previous encryption.

Below we give the exact definition of the octonion public-key cryptosystem (O-PKC).

The public-key generation procedure

1. *The user selects an integer $m \geq 0$ and an integral octonion $k \in \mathbf{O}_m^I$ with the known totient function $\lambda(k, m)$, i.e., $k^{\lambda(k,m)} = 1 \bmod m$. For the selected integer $e \in [1, \lambda(k, m)]$ coprime to $\lambda(k, m)$, the user calculates the integer number d such, that $e \cdot d = 1 \bmod \lambda(k, m)$.*
2. *The user publishes the set $\{k, e, m\}$ and the set $\{k, d, m\}$ keeps private. The two sets form a public and private cryptographic keys respectively.*

The encryption–decryption procedure in O-PKC

1. *The user* A *writes the plain text as an octonion $u = (u_0, \ldots, u_7) \in \mathbf{O}_{m_B}^I$, and calculates*

$$u_{k_B} = u k_B u^{-1} \bmod m_B,$$

where $\{k_B, e_B, m_B\}$ is the public key of the user B. *The plain text encoded in u_{k_B} the user* A *encrypts according to the formula*

$$C(u_{k_B}) = (u_{k_B})^{e_B} \bmod m_B$$

and sends $C(u_{k_B})$ to B.

2. *The user* B *decrypts the cipher text $C(u_{k_B})$ by means of the formula*

$$C(u_{k_B})^{d_B} = ((u_{k_B})^{e_B})^{d_B} = u_{k_B} \bmod m_B.$$

3. *The user* B *decodes the plain text u by solving the linear congruence equation*

$$x k_B = u_{k_B} x \bmod m_B.$$

The security of the octonionic public-key cryptosystem is based on the computational difficulty of factorization of the modulus m and difficulty of finding the values of the totient function $\lambda(k, p)$ for a large prime numbers. Let us assume that, the modulus m is a product of two prime numbers p, q and the norm $\mathrm{N}(k)$ of the octonion k is coprime to $m = pq$, then the following formula is satisfied

$$k^{\lambda(k,p)\lambda(k,q)} = 1 \bmod pq, \quad p \neq q. \tag{6}$$

If the product $\lambda(k, p)\lambda(k, q)$ has the lowest value, among all numbers satisfying (6), then it is equal to the totient $\lambda(k, pq)$. In general, the function $\lambda(k, pq)$ satisfies the inequalities

$$\operatorname{lcm}(\lambda(k, p), \lambda(k, q)) \leq \lambda(k, pq) \leq \lambda(k, p)\lambda(k, q),$$

where lcm means the least common multiple. The above formula allows in a relatively easy way to calculate $\lambda(k, pq)$ having $\lambda(k, p)$ and $\lambda(k, q)$. Unfortunately, an effective algorithm for determining the octonion totient function for a large prime modulus is currently unknown. Partly, the problem can be solved by adopting to the octonion algebra the quantum algorithm for finding the orders of a finite cyclic subgroups of noncommutative group proposed by Mosca and Ekert in [24].

As an example of application of the O-PKC algorithm we encrypt and decrypt a plain text represented by the Gravesian octonion $u = (26, 27, 28, 29, 30, 31, 32, 3) \in \mathbf{O}^I_{35}$ with the norm $\mathrm{N}(u) = 9 \bmod 35$. The octonion $k = \frac{1}{2}(e_0 + e_1 + e_3 + e_7) + (e_2 + 2e_4 + 3e_5 + 4e_6)$ we will use to encode the text u before encryption. The minimal polynomial of k is $k^2 = k - 31$ which means, that k is an integer octonion with the norm $N(k) = 31$. As the modulus we take $m = 35$. The totient function for k in $\mathbf{O}^I_{35}$ is $\lambda(k, 35) = 48$. As the pair of public-private keys we select $\{k, e = 5, m = 35\}$ and $\{k, d = 29, m = 35\}$. The encoded plain text is $u_k = uku^{-1} \bmod 35 = (31, 20, 34, 8, 19, 3, 26, \frac{1}{2})$. The cipher text $C(u_k)$ is $(u_k)^5 \bmod 35 = (24, 20, 6, 22, 26, 17, 19, 33)$. After the decryption $C(u_k)^{29} \bmod 35$ the recipient obtains u_k. For a given three fixed parameters in u the recipient is able uniquely to recover the rest of them.

To determine the totient for the integral octonion we can use the 'matrix representation' of the octonion algebra. For the vector $\overline{e} = (e_0, e_1, \ldots, e_7)$ we define the matrix $M(e_i) \equiv M_i$ as the linear transformation of $\overline{e}$ under the left action of e_i, $i \in [0, 7]$, $M_i\overline{e} = e_i\overline{e}$. The M_i matrices can be deduce from the following equations

$$\begin{aligned}
M_0\overline{e} &= (-e_7, e_3, e_6, -e_1, e_5, -e_4, -e_2, e_0),\\
M_1\overline{e} &= (-e_3, -e_7, e_4, e_0, -e_2, e_6, -e_5, e_1),\\
M_2\overline{e} &= (-e_6, -e_4, -e_7, e_5, e_1, -e_3, e_0, e_2),\\
M_3\overline{e} &= (e_1, -e_0, -e_5, -e_7, e_6, e_2, -e_4, e_3),\\
M_4\overline{e} &= (-e_5, e_2, -e_1, -e_6, -e_7, e_0, e_3, e_4),\\
M_5\overline{e} &= (e_4, -e_6, e_3, -e_2, -e_0, -e_7, e_1, e_5),\\
M_6\overline{e} &= (e_2, e_5, -e_0, e_4, -e_3, -e_1, -e_7, e_6).
\end{aligned}$$

The multiplication

$$M_i \cdot M_j = M_i G_{i,j} M_j,$$

where

$$\begin{aligned}
G_{5,6} = G_{1,6} = G_{1,5} &= \operatorname{diag}(1, -1, 1, 1, 1, -1, -1, -1),\\
G_{4,6} = G_{3,6} = G_{3,4} &= \operatorname{diag}(1, 1, 1, -1, -1, 1, -1, -1),
\end{aligned}$$

$$
\begin{aligned}
G_{4,5} = G_{0,5} = G_{0,4} &= \mathrm{diag}(-1, 1, 1, 1, -1, -1, 1, -1),\\
G_{3,5} = G_{2,5} = G_{2,3} &= \mathrm{diag}(1, 1, -1, -1, 1, -1, 1, -1),\\
G_{2,6} = G_{0,6} = G_{0,2} &= \mathrm{diag}(-1, 1, -1, 1, 1, 1, -1, -1),\\
G_{2,4} = G_{1,4} = G_{1,2} &= \mathrm{diag}(1, -1, -1, 1, -1, 1, 1, -1),\\
G_{1,3} = G_{0,3} = G_{0,1} &= \mathrm{diag}(-1, -1, 1, -1, 1, 1, 1, -1)
\end{aligned}
$$

defines the 'matrix representation' of the octonion algebra. The octonion algebra is power associative which means, that to determine the totient for a given element $x \in \mathbf{O}_m$ we only need to calculate the $M(x)^\lambda \bmod m$, where $M(x) = \sum_{i=0}^{7} x_i M_i$ and $M(x)^\lambda$ is an ordinary matrix multiplication.

From the minimal polynomial equation (1) follows that arbitrary power of an octonion $x \in \mathrm{O}$ is proportional to x, i.e.,

$$x^n = U_n x + V_n, \quad n \neq 1,$$

where

$$
\begin{aligned}
U_n &= \frac{1}{2\sqrt{x_7^2 - \mathrm{N}(x)^2}}((x_7 + \sqrt{x_7^2 - \mathrm{N}(x)^2})^n\\
&\quad -(x_7 - \sqrt{x_7^2 - \mathrm{N}(x)^2})^n), n > 0,\\
V_n &= \frac{-\mathrm{N}(x)^2}{2\sqrt{x_7^2 - \mathrm{N}(x)^2}}((x_7 + \sqrt{x_7^2 - \mathrm{N}(x)^2})^{n-1}\\
&\quad -(x_7 - \sqrt{x_7^2 - \mathrm{N}(x)^2})^{n-1}),
\end{aligned}
\tag{7}
$$

$x_7 = \mathrm{Re}(x)$ and $U_{-n} = \frac{1}{\mathrm{N}(x)^{n-1}} U_n$, $V_{-n} = \frac{1}{\mathrm{N}(x)^n} V_n$. The coefficients U_n and U_n we obtained by solving the following recurrence equation

$$x_{n+2} - 2\mathrm{Re}(x)\ x_{n+1} + \mathrm{N}(x) \quad x_n = 0, \quad x_0 = 1.$$

One of the possible attacks on the totient function $\lambda(k, m)$ there is an attempt to solve the congruence equations $k^\lambda = U_\lambda k + V_\lambda = 1 \bmod m$ or equivalently

$$
\begin{cases}
U_\lambda(k_7, \mathrm{N}(k)) = 0 \bmod m,\\
V_\lambda(k_7, \mathrm{N}(k)) = 1 \bmod m,
\end{cases}
\tag{8}
$$

where U_λ and V_λ are given in (7). To solve the equation (8) it is necessary to expand the functions U_λ and V_λ in x_7 and $N(x)$ variables which makes such attack impractical.

3 The symmetric key exchange algorithm on quaternion algebra

The positive definite bilinear form $(x, y) = \frac{1}{2}\, tr(x\bar{y})$ allows to define the orthogonality in the octonion algebra O. Two octonions x, y are said to be orthogonal if $(x, y) = 0$. The natural orthogonal decomposition of octonion algebra can be defined using the Fano lines. The generators of the octonion algebra which lie on the Fano line form a quaternion subalgebra D of O. The generators which lie in the completion of the Fano line in the Fano plane form the orthogonal completion $D^{\perp}$ of D. If we select an element α from $D^{\perp}$, then $\forall x \in D \;\; (x, \alpha) = 0$ and any octonion can be written in the form $x + y\alpha$, where $x, y \in D$ and $\alpha \in D^{\perp} \equiv D\alpha$. A bijective mapping T of O onto itself is defined to be an automorphism if

$$\begin{aligned} T(a+b) &= T(a) + T(b), \\ T(a\, b) &= T(a)\; T(b), \\ T(u) &= v \end{aligned}$$

for any $a, b \in$ O and u, v units in O. Let T be an automorphism of octonion algebra leaving the quaternion subalgebra D invariant, i.e., $T(D) = D$ and $T(D^{\perp}) = D^{\perp}$. For $x + y\alpha \in D \bigoplus D\alpha$ and $u = T|D$, $w = T|D^{\perp}$ we define T in the following way

$$T(x + y\alpha) = u(x) + u(y)w(\alpha).$$

For $u(x) = cxc^{-1}$ and $w(\alpha) = d\alpha d^{-1}$, $x, c, d \in D$ we obtain

$$\begin{aligned} T(x + y\alpha) &= cxc^{-1} + (cyc^{-1})(d\alpha d^{-1}) \\ &= cxc^{-1} + \frac{1}{N(d)}(d^2cyc^{-1})\alpha, \end{aligned}$$

which is an automorphism of the octonion algebra, [20]. The transformation $T_{c,p} = cxc^{-1} + (pcyc^{-1})\alpha$, where $N(p) = 1$, satisfies the equation

$$T_{c,p}((x_1 + y_1\alpha)(x_2 + y_2\alpha)) = T_{c,p}(x_1 + y_1\alpha)T_{c,p}(x_2 + y_2\alpha), \tag{9}$$

and has the multiplication property $T_{c_1,p_1}T_{c_2,p_2} = T_{c_1k_1,p_1c_1p_2c_1^{-1}}$. We use the splitting $D \bigoplus D\alpha$ of the octonion algebra to construct a symmetric cryptographic key on the quaternion subalgebra D and then we extend the key to the completion $D^{\perp}$.

The construction of the symmetric cryptographic key on the quaternion algebra D is motivated by the AAG symmetric key exchange protocol, [25], in which instead of the formula $K = aba^{-1}b^{-1}$ for the cryptographic key, where a, b are elements of a given non-abelian group G, we use the formula $K = a_0b_Na_N^{-1}b_0^{-1}$, where $a_0, b_N, a_N, b_0 \in D$. To explain the main idea of the construction we begin with a trivial example. Let the user A selects two sets of quaternions $S_a = \{a_0, a_1, a_2\}$, $S_v = \{v_1, v_2\}$ such, that $a_2 = v_1v_2$. Similarly, the user B selects $S_b = \{b_0, b_1, b_2\}$ and $S_u = \{u_1, u_2\}$ such, that $b_2 = u_1u_2$. The users exchange the sets S_v and S_u. The user A calculates the set $S_{aua^{-1}} = \{a_0u_1a_1^{-1}, a_1u_2a_2^{-1}\}$ and sends it to B. Similarly, the user B calculates

$S_{bvb^{-1}} = \{b_0v_1b_1^{-1}, b_1v_2b_2^{-1}\}$ and sends the set $S_{bvb^{-1}}$ to A. Both users are able to calculate the common key $K = a_0b_2a_2^{-1}b_0^{-1}$. In this example, the man-in-the-middle attack can easily recover the cryptographic key K because for a given quaternions $w_1 = a_0u_1a_1^{-1}$, $w_2 = a_1u_2a_2^{-1}$ from $S_{aua^{-1}}$ to recover the unknown variable a_0 it's enough to solve the equation

$$a_0u_1u_2 = w_1w_2a_2.$$

For a known expression $a_2 = v_1v_2$ this equation can be trivially solved. In the proposed algorithm, we hide the variable a_2 in order to make it difficult to find the element a_0 and the key K. To do this, we introduce a graph G in the quaternion algebra and for encryption we use paths between two fixed nodes (quaternions) that allow uniquely define the shared cryptographic key.

By $G = \{E, V_{x,y}\}$ we denote the graph with the set of directed edges E and the set of nodes

$$V_{x,y} = \{x_iy_jx_k^{-1}\}_{i,k\in[0,N],j\in[1,N']},$$

where $x_i, y_j \in D$. We define a directed path $p_{a,v}$ with the set of nodes

$$V_{a,v}(p) = \{a_iv_ja_k^{-1}\}_{i,j,k} \subset V_{x,y}|_{x=a,y=v}$$

such, that the product of subsequent nodes on the path satisfies the equation

$$\prod_{a_iv_ja_k^{-1}\in V_{a,v}(p)} a_iv_ja_k^{-1} = a_0(v_{i_1}\cdots v_{i_{N'}})a_N^{-1} = a_0, \tag{10}$$

which means, that $a_N = v_{i_1}\cdots v_{i_{N'}}$. The paths $p_{a,v} \subset G$ with the property (10) we will use for definition of the cryptographic key exchange algorithm in the quaternion subalgebra D.

The quaternion symmetric key exchange algorithm

1. *The user* A *generates two sets of quaternions* $S_a = \{a_i : a_i \in D\}_{i=0}^{N}$ *and* $S_v = \{v_i : v_i \in D\}_{i=1}^{N'}$. *The set* S_a *is kept secret. The set* S_v *is sent to the user* B. *The user* A *selects a path* $p_A(a, v)$ *in the graph* G *from the root node, represented by the quaternion* $a_0v_1a_i^{-1}$, *to the leaf node* $a_kv_{N'}a_N^{-1}$. *The path is selected such, that the formula* (10) *is satisfied.*
   In a similar way, the user B *generates the two sets* S_b *and* S_u *of quaternions. The set* S_b *is kept secret by* B. *The set* S_u *is sent to the user* A. *The user* B *selects a path* $p_B(b, u)$ *in the graph* G *from the root node* $b_0u_1b_i^{-1}$ *to the leaf node* $b_ku_{N'}b_N^{-1}$. *such, that the formula* (10) *is satisfied, i.e.,* $b_N = u_{i_1}\cdots u_{i_{N'}}$.
2. *The user* A, *based on the received set* S_u, *calculates*

$$S_{aua^{-1}} = \{a_iu_ja_k^{-1} : x_iy_jx_k^{-1} \in V_{x,y}, \quad x_i = a_i, \quad y_j = u_j\}$$

and sends the set $S_{aua^{-1}}$ to B.
The user B *calculates*

$$S_{bvb^{-1}} = \{b_i v_j b_k^{-1} : x_i y_j x_k^{-1} \in V_{x,y}, \quad x_i = b_i, \quad y_j = v_j\}$$

and sends the set $S_{bvb^{-1}}$ to A.

3. *The user* A *uses the set $S_{bvb^{-1}}$ to calculate the product of sequential vertices of the path $p_A(b, v)$. In this way* A *obtains $b_0 a_N b_N^{-1}$.*
The user B*, based on the set $S_{aua^{-1}}$, calculates the product of sequential vertices of the path $p_B(a, u)$. In this way* B *obtains $a_0 b_N a_N^{-1}$.*
4. *The users* A *and* B *calculate $k_A = ((b_0 a_N b_N^{-1}) a_0^{-1})^{-1}] \in D$ and $k_B = (a_0 b_N a_N^{-1}) b_0^{-1} \in D$ respectively. The shared cryptographic key of* A *and* B *is*

$$k_{a,b} = a_0 b_N a_N^{-1} b_0^{-1}.$$

To recover the cryptographic key $k_{a,b}$ the man in the middle can try to find the quaternions a_0, a_N by solving the system of linear equations obtained from the set $S_{aua^{-1}}$, i.e., $\{a_i u_j = w_j a_k\}_{j=1}^{N'}$, or recover the quaternion a_N from the formula (10) by searching the correct encryption path in the graph G. After the reduction, the man in the middle can obtain the set of two equations

$$a_i u = w a_i, \quad i = 0, N.$$

The unique solution of these equations can be obtained from the formula (5) after fixing two parameters in a_i. Also, the attack on the cryptographic key by searching the correct cryptographic path in a sufficiently complex graph G is not very effective. It means that the quaternions a_N, b_N can be regarded as unknown variables.

For the two keys $k_{a,b} = a_0 b_N a_N^{-1} b_0^{-1}$, $k_{c,d} = c_0 d_{N'} c_{N'}^{-1} d_0^{-1}$ generated by the quaternion symmetric key exchange algorithm we define the symmetric octonion cryptographic key as the automorphism

$$K_{k_{a,b}, p_{c,d}}(x + y\alpha) = k_{a,b} x k_{a,b}^{-1} + (p_{c,d} k_{a,b} y k_{a,b}^{-1})\alpha,$$

where $p_{c,d} = \frac{1}{\mathrm{N}(k_{c,d})} k_{c,d}^2$ $(N(p_{c,d}) = 1)$ and $\alpha \in D^{\perp}$. The key $K_{k_{a,b}, p_{c,d}}$ can be applied to encrypt a product of octonions by means of the formula (9). The multiplication property of $K_{k_{a,b}, p_{c,d}}$ allows to compose the keys from several others keys.

Below, we apply defined the quaternion key exchange algorithm to generate the cryptographic key using the graph G given on Fig. 1. In the graph there are four paths between the root node $x_0 y_1 x_1^{-1}$ and the leaf node $x_4 y_7 x_5^{-1}$ which means that there are four possible ways to choose the cryptographic key $k_{a,b}$.

1. *The user* A *generates two sets of quaternions $S_a = \{a_0, a_1, a_2, a_3, a_4, a_5\}$, $S_v = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and selects the path*

$$p_A(a, v) = \{a_0 v_1 a_1^{-1}, a_1 v_2 a_2^{-1}, a_2 v_4 a_3^{-1}, a_3 v_5 a_4^{-1}, a_4 v_7 a_5^{-1}\}.$$

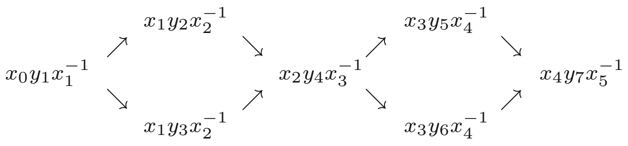

Fig. 1 The graph G for generation of the cryptographic keys

From the relation (10) *it follows that*

$$(a_0v_1a_1^{-1})(a_1v_2a_2^{-1})(a_2v_4a_3^{-1})(a_3v_5a_4^{-1})(a_4v_7a_5^{-1}) = a_0$$

and $a_5 = v_1v_2v_4v_5v_7$.
The user B *generates two sets of quaternions* $S_b = \{b_0, b_1, b_2, b_3, b_4, b_4\}$, $S_u = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$ *and selects the path*

$$p_B(b,u) = \{b_0u_1b_1^{-1}, b_1u_3b_2^{-1}, b_2u_4b_3^{-1}, b_3u_6b_4^{-1}, b_4u_7b_5^{-1}\}.$$

From the relation (10) *it follows that* $b_5 = u_1u_3u_4u_6u_7$.

2. *The user* A *calculates*

$$S_{aua^{-1}} = \{a_0u_1a_1^{-1}, a_1u_2a_2^{-1}, a_1u_3a_2^{-1}, a_2u_4a_3^{-1}, a_3u_5a_4^{-1}, a_3u_6a_4^{-1}, a_4u_7a_5^{-1}\}$$

and sends the set $S_{aua^{-1}}$ *to* B.
The user B *calculates*

$$S_{bvb^{-1}} = \{b_0v_1b_1^{-1}, b_1v_2b_2^{-1}, b_1v_3b_2^{-1}, b_2v_4b_3^{-1}, b_3v_5b_4^{-1}, b_3v_6b_4^{-1}, b_4v_7b_5^{-1}\}$$

and sends $S_{bvb^{-1}}$ *to* A.

3. *The user* A *calculates the product of sequential vertices in the path* $p_A(b,v)$

$$(b_0v_1b_1^{-1})(b_1v_2b_2^{-1})(b_2v_4b_3^{-1})(b_3v_5b_4^{-1})(b_4v_7b_5^{-1}) = b_0a_5b_5^{-1}$$

and calculates the key $k_A = ((b_0a_5b_5^{-1})a_0^{-1})^{-1}$.
The user B *calculates the product of sequential vertices in the path* $p_B(a,u)$

$$(a_0u_1a_1^{-1})(a_1u_3a_2^{-1})(a_2u_4a_3^{-1})(a_3u_6a_4^{-1})(a_4u_7a_5^{-1}) = a_0b_5a_5^{-1}$$

and calculates the key $k_B = (a_0b_5a_5^{-1})b_0^{-1}$. *The shared cryptographic key is* $k_{a,b} = a_0b_5a_5^{-1}b_0^{-1}$.

The security of the cryptographic key $k_{a,b}$ depends on the difficulty of finding the path $p_A(b,v)$ or $p_B(b,u)$ in G. In the example there are four ways to choose the path. In general, for increasing number of nodes in the graph the number of paths grows exponentially.

4 The derivation octonionic symmetric key exchange algorithm

A derivation is transformation with the property

$$D(ab) = D(a)b + aD(b). \tag{11}$$

In an associative algebra any element x defines an inner derivation $\mathrm{ad}_x(y) = xy - yx$. In a non-associative algebra this formula usually does not satisfy (11). We define two operations in octonion algebra $L_x(a) = ax$ and $R_x(a) = xa$, $a, x \in$ O. We note, that these operations for a non-associative algebra have the following composition rules $L_x L_y(a) = x(ya)$ and similarly for $R_x R_y(a) = (ay)x$. The transformation $D_{x,y}$: O → O defined as

$$D_{x,y}(z) = [L_x, L_y](z) + [L_x, R_y](z) + [R_x, R_y](z),$$

is a derivation in the octonion algebra, [26]. The derivation $D_{x,y}$ we write in the standard form

$$D_{x,y}(z) = [[x, y], z] - 3(x, y, z), \tag{12}$$

where (x, y, z) denotes the associator $(xy)z - x(yz)$ and $[x, y] = xy - yx$. To construct the symmetric key exchange algorithm on the octonion algebra we use the following property of the derivation (12)

$$[D_{a,b}, D_{x,y}] = D_{D_{a,b}(x),y} + D_{x,D_{a,b}(y)}. \tag{13}$$

The basic idea that lies behind the construction of the algorithm is that for a given monomial $p(u) = u_1 \cdots u_N$ in octonion algebra its derivative $D_{x,y}(p(u))$ can be expressed by the elements u_i and $D_{x,y}(u_i)$. Let us assume that, the user A generates two polynomials $\{p_{A,1}(\overline{v}), p_{A,2}(\overline{v})\}$ of $\overline{v} = (v_1, \ldots, v_{N'})$ variables and the user B two polynomials $\{p_{B,1}(\overline{u}), p_{B,2}(\overline{u})\}$ of $\overline{u} = (u_1, \ldots, u_N)$ variables. For a selected values of $\overline{v}$, $a_i = p_{A,i}(\overline{v}_0)$, $i = 1, 2$, the user A calculates the derivative $D_{a_1,a_2}(u_i)$ of each octonion in the tuple $\overline{u}$ and sends it to the user B. Using the derivatives $D_{a_1,a_2}(u_i)$ the user B is able to calculate $D_{a_1,a_2}(p_{B,1}(\overline{u}_0))$ and $D_{a_1,a_2}(p_{B,2}(\overline{u}_0))$. The key $K^B_{(a_1,a_2),(p_{B,1},p_{B,2})} = [D_{a_1,a_2}, D_{p_{B,1},p_{B,2}}]$ is determined by the user B using the formula

$$K^B_{(a_1,a_2),(p_{B,1},p_{B,2})} = D_{D_{a_1,a_2}(p_{B,1}),p_{B,2}} + D_{p_{B,1},D_{a_1,a_2}(p_{B,2})}.$$

In a similar way, the user A using the derivatives $D_{b_1,b_2}(v_i)$, where $b_i = p_{B,i}(\overline{u}_0)$, $i = 1, 2$, is able to calculate $D_{b_1,b_2}(p_{A,1}(\overline{v}_0))$, $D_{b_1,b_2}(p_{A,2}(\overline{v}_0))$ and the key

$$K^A_{(b_1,b_2),(p_{A,1},p_{A,2})} = D_{D_{b_1,b_2}(p_{A,1}),p_{A,2}} + D_{p_{A,1},D_{b_1,b_2}(p_{A,2})}.$$

Because both expressions differ by the sign the common cryptographic key is

$$K_{(a_1,a_2),(b_1,b_2)} = -K^{A}_{(b_1,b_2),(p_{A,1},p_{A,2})} = K^{B}_{(a_1,a_2),(p_{B,1},p_{B,2})}.$$

Below, we give the exact definition of the symmetric key exchange cryptographic algorithm based on the octonion derivation.

The octonionic symmetric key exchange algorithm

1. *The user* A *selects three octonions* v_1, v_2, w *and two polynomials* $p_{A,i}$, $i = 1, 2$. *For the selected otonions* v_1, v_2 *the user* A *calculates the secret* $a_i = p_{A,i}(v_1, v_2)$, $i = 1, 2$. *The imaginary parts of* v_1, v_2, *i.e., the set* $S_v = \{\mathrm{Im}(v_1), \mathrm{Im}(v_2)\}$ *and the octonion* w *the user* A *sends to* B.
   The user B *selects two octonions* u_1, u_2, *two polynomials* $p_{B,i}$, $i = 1, 2$ *and calculates the secret* $b_i = p_{B,i}(u_1, u_2)$, $i = 1, 2$.
   The set $S_u = \{\mathrm{Im}(u_1), \mathrm{Im}(u_2)\}$ *is sent to* A.
2. *The user* A *calculates*

   $$D_{a_1,a_2}(S_u) = \{D_{a_1,a_2}(\mathrm{Im}(u_1)), D_{a_1,a_2}(\mathrm{Im}(u_2))\}$$

   and sends the set $D_{a_1,a_2}(S_u)$ *to* B.
   The user B *calculates*

   $$D_{b_1,b_2}(S_v) = \{D_{b_1,b_2}(\mathrm{Im}(v_1)), D_{b_1,b_2}(\mathrm{Im}(v_2))\}$$

   and sends the set $D_{b_1,b_2}(S_v)$ *to* A.
3. *The user* A *calculates the derivation* $D_{b_1,b_2}(p_{A,i}(v_1, v_2))$, $i = 1, 2$

   $$D_{b_1,b_2}(a_i) = p'_{A,i}(v_1, v_2, D_{b_1,b_2}(v_1), D_{b_1,b_2}(v_2))).$$

   The user B *calculates the derivation* $D_{a_1,a_2}(p_{B,i}(u_1, u_2))$, $i = 1, 2$

   $$D_{a_1,a_2}(b_i) = p'_{B,i}(u_1, u_2, D_{a_1,a_2}(u_1), D_{a_1,a_2}(u_2))).$$

4. *For the agreed octonion* $w \in \mathrm{Im(O)}$, *the user* A *calculates the key*

   $$K^{A}_{(b_1,b_2),(a_1,a_2)}(w) = D_{D_{b_1,b_2}(a_1),a_2}(w) + D_{a_1,D_{b_1,b_2}(a_2)}(w), \qquad (14)$$

   and B *calculates*

   $$K^{B}_{(a_1,a_2),(b_1,b_2)}(w) = D_{D_{a_1,a_2}(b_1),b_2}(w) + D_{b_1,D_{a_1,a_2}(b_2)}(w). \qquad (15)$$

   The octonion

   $$K(w) = -K^{A}_{(b_1,b_2),(a_1,a_2)}(w) = K^{B}_{(a_1,a_2),(b_1,b_2)}(w)$$

   defines the common secret key.

Let us show an example how this algorithms works.

1. *The user* A *selects there octonions* $v_1 = \{2, -3, 0, 1, -2, 5, -1, 1\}$, $v_2 = \{5, 1, 2, 7, 4, -1, 0, -2\}$, $w = \{1, 2, 3, -1, 4, -2, 0, 0\}$ *and two polynomials*

$$p_{A,1} = v_1 + 11v_2 + 3v_1v_2 - v_1v_2v_1,$$
$$p_{A,2} = v_1 + 6v_2 + v_2v_1 - 5v_2v_1v_2.$$

Next, the user A *calculates two secret octonions* $a_i = p_{A,i}(v_1, v_2)$, $i = 1, 2$. *The set* $S_v = \{\mathrm{Im}(v_1), \mathrm{Im}(v_2)\}$, *where* $\mathrm{Im}(v_1) = \{2, -3, 0, 1, -2, 5, -1, 0\}$, $\mathrm{Im}(v_2) = \{5, 1, 2, 7, 4, -1, 0, 0\}\}$ *and the octonion* w *is sent to* B.
The user B *selects two octonions* $u_1 = \{3, 1, 7, 2, -4, 0, 5, -2\}$, $u_2 = \{-2, -5, 0, 1, 6, -1, 3, -3\}$ *and two polynomials*

$$b_1 = 3u_1 + u_2 + 3u_1u_2 - u_1u_2u_1,$$
$$b_2 = u_1 - 2u_2 - 5u_2u_1 + u_2u_1u_2.$$

Next, the user B *calculates two secret octonions* $b_i = p_{B,i}(u_1, u_2)$, $i = 1, 2$.
The set $S_u = \{\mathrm{Im}(u_1), \mathrm{Im}(u_2)\}$, *where* $\mathrm{Im}(u_1) = \{3, 1, 7, 2, -4, 0, 5, 0\}$, $\mathrm{Im}(u_2) = \{-2, -5, 0, 1, 6, -1, 3, 0\}$ *is sent to* A.

4. *For the agreed octonion* w, *the user* A *calculates* $-K^{A}_{(b_1,b_2),(a_1,a_2)}(w)$ *and the user* B *calculates* $K^{B}_{(a_1,a_2),(b_1,b_2)}(w)$ *according to the formulas* (14), (15). *The common secret key is*

$$\begin{aligned} K(w) = 8\{&5549509343115, 1850183670668, \\ &-1020644923030, -2086176426557, -4111458908175, \\ &-4085858645391, 338344436964\}. \end{aligned}$$

To recover the secret octonions a_1 and a_2 (b_1, b_2) based on the knowledge of u_1, u_2 and w_1, w_2 it is necessary to solve the set of equations $D_{a_1,a_2}(u_i) = w_i$, $i = 1, 2$. For imaginary octonions this set of equations can be written in the form

$$(a_1a_2)u_i + 3a_1(a_2u_i) = w_i \quad a_1, a_2, u_i, w_i \in \mathrm{Im}(\mathrm{O}). \tag{16}$$

It consists of 12 equations with 14 unknown variables. Because, the key $K(w)$ depends also on the real parts of the octonions v_i and u_i, which are kept secret, to recover the octonions a_1 and a_2 the man in middle should determine 16 parameters having only the system of twelve equations (16).

5 Conclusion

We discussed three key exchange cryptographic protocols in the octonion algebra. The proposed octonionic public-key cryptosystem is the generalization of the RSA algorithm to the octonion arithmetics. We defined a totient function for an invertible, integral octonion k as the order of a multiplicative cyclic group mod m generated by the element k. We proved, that any octonion u_k which lies in the orbit of the element k under the adjoint action of the octonion algebra has the same totient function. This property of the totient function allows to encrypted and decrypted the elements from the orbit by the same pair of the cryptographic keys $\{k, e, m\}$ and $\{k, d, m\}$. The disadvantage of proposed algorithm is that the recipient, to recover the cipher text, must solve the congruence equation $xu = wx \mod m$, which unique solution requires knowledge of a three parameter of x. We also proposed two symmetric cryptographic key exchange protocols based on the automorphism and the derivation of the octonion algebra. We apply the quaternion symmetric key exchange algorithm to generate two quaternionic cryptographic keys and used them to define the shared octonionic key as an automorphism of the octonion algebra which leaves the quaternion subalgebra invariant. To generate a cryptographic key by mens of the quaternion key exchange algorithm it is necessary to select a path in a graph defined over the quaternion algebra. Security of the proposed quaternion key exchange algorithm is based on the complexity of the graph. The second symmetric key exchange protocol we defined using the derivation mapping in the octonion algebra. By calculating the commutator of two derivatives both sides of the data exchange can generate a common cryptographic key. The future work will include detailed analysis of security aspects of the proposed algorithms and creating models of attacks on the cryptographic keys defined in the octonion algebra and in any non-associative algebraic structures.

References

1. Dehn, M.: Über unendliche diskontinuierliche Gruppen. Math. Ann. **71**, 116–144 (1911)
2. Miller III, ChF: On Group-Theoretic Decision Problems and Their Classification, AMS, vol. 68. Princeton University Press, Princeton (1971)
3. Novikov, P.S.: Unsolvability of the conjugacy problem in the theory of groups. Izv. Akad. Nauk SSSR Ser. Mat. **18**(6), 485–524 (1954)
4. Novikov, P.S.: On the algorithmic unsolvability of the word problem in group theory. Trudy Mat. Inst. Steklov. **44**, 3–143 (1955). (in Russian)
5. Boone, W.W.: Certain simple unsolvable problems of group theory. Indag. Math. **16**, 231–237, 492–497 (1954)
6. Boone, W.W.: Certain simple unsolvable problems of group theory. Indag. Math. **17**, 252–256, 571–577 (1955)
7. Boone, W.W.: Certain simple unsolvable problems of group theory. Indag. Math. **19**, 22–27, 227–232 (1957)
8. Boone, W.W.: The word problem. Ann. Math. **70**, 207–265 (1959)
9. Magyarik, M.R., Wagner, N.R.: A public key cryptosystem based on the word problem. In: Advances in Cryptology—CRYPTO 1984, Lecture Notes in Computer Science, vol. 196, pp. 19–36. Springer, Berlin (1985)
10. Ko, K.H., Lee, S.J., Cheon, J.H., Han, J.W., Kang, J., Park, Ch.: New public-key cryptosystem using braid groups. In: Advances in Cryptology—CRYPTO 2000, Lecture Notes in Computer Science, vol. 1880, pp. 166–183. Springer, Berlin (2000)

11. Anshel, I., Anshel, M., Goldfeld, D.: An algebraic method for public-key cryptography. Math. Res. Lett. **6**, 1–5 (1999)
12. Anshel, I., Anshel, M., Fisher, B., Goldfeld, D.: In: Naccacne, D. (ed.), New Key Agreement Protocols in Braid Group Cryptography, CT-RSA 2001, LNCS, vol. 2020, pp. 13–27. Springer, Berlin (2001)
13. Koscielny, C., Mullen, G.L.: A quasigroup-based public-key cryptosystem. Int. J. Appl. Math. Comput. Sci. **9**(4), 955–963 (1999)
14. Kalka, A.: Non-associative public-key cryptography. arXiv:1210.8270v1 (2012)
15. Dehornoy, P.: Braids and Self-distributivity, Progress in Mathematics, vol. 192. Birkhäuser, Basel (2000)
16. Myasnikov, A., Shpilrain, V., Ushakov, A.: Non-commutative Cryptography and Complexity of Group-theoretic Problems, Mathematical Surveys and Monographs, vol. 177. AMS, Providence (2011)
17. Conway, J.H., Smith, D.A.: On Quaternions and Octonions: Their Geometry, Arithmetic, and Symmetry. Peters, Ltd., Natick (2003)
18. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
19. Baez, J.C.: The octonions. Bulletin (New Series) of the AMS **39**(2), 45–205 (2001)
20. Springer, T.A., Veldkamp, F.D.: Octonions, Jordan Algebras, and Exceptional Groups, Springer Monographs in Mathematics. Springer, Berlin (2000)
21. Lamont, P.J.C.: Arithmetics in Cayley's algebra. Proc. Glasg. Math. Assoc. **6**, 99–106 (1963)
22. Rehm, H.P.: Prime factorization of integral Cayley octaves. Ann. Fac. Sci. Toulouse 6e sér. Tome **2**(2), 271–289 (1993)
23. Lamont, P.J.C.: Factorization and congruence in the arithmetics of Cayleys algebra. Glasg Math. J. **33**(2), 171–180 (1991)
24. Mosca, M., Ekert, A.: The hidden subgroup problem and eigenvalue estimation on a quantum computer. In: Williams, C.P. (ed.) QCQC'98, LNCS, vol. 1509, pp. 174–188. Springer, Berlin (1999)
25. Anshel, I., Anshel, M., Goldfeld, D.: Non-abelian key agreement protocols. Discrete Appl. Math. **130**, 3–12 (2003)
26. Schafer, R.D.: Introduction to Non-associative Algebras. Dover, New York (1995)