The Matrix-based Knapsack Cipher in the Context of Additively Homomorphic Encryption

Aleksei Vambol[1][0000-0003-1929-7783]

[1] Department of Computer Systems, Networks and Cybersecurity,
National Aerospace University «KhAI», Kharkiv, Ukraine
o.vambol@csn.khai.edu

Abstract. The purpose of this paper is the research of the matrix-based knapsack cipher in the context of additively homomorphic encryption and its application in secret electronic voting. Group-based knapsack ciphers represent a novel approach for building knapsack-like encryption schemes, which is based on the properties of generating sets of finite groups and isomorphic transformations. The matrix-based knapsack cipher is a cryptosystem of the given class, which is constructed using the direct product of diagonal subgroups of a general linear group over a finite field. In the given paper two new results are proposed. The first one is represented by the proof that the considered encryption scheme is additively homomorphic. This property allows the investigated cipher to be used in several application fields besides key agreement in hybrid cryptosystems. The second result is a brief description of a new secret electronic voting protocol, which is built on the basis of the matrix-based knapsack cipher.

Keywords: matrix-based knapsack cipher, homomorphic encryption, knapsack cryptosystems, electronic voting, asymmetric ciphers.

1 Introduction

Asymmetric ciphers, also known as public-key encryption schemes, are of great importance for ensuring the confidentiality of data exchange as they allow users of insecure communication channels to establish a common secret key for a symmetric cryptosystem [1]. For this reason the given ciphers are used in such widespread network protocols like TLS, SSH and IKEv2 [2].

The knapsack-like ciphers constitute the class of asymmetric encryption schemes, which has been among the earliest representatives of public-key cryptosystems. The Merkle-Hellman knapsack cipher is the historically first encryption scheme of the given class. Although this cryptosystem, as well as many other ciphers of the aforementioned class, has been broken, knapsack-based cryptography continues to attract attention of researchers [1]. Attempts to develop new or improve the existing cryptosystems of this class are undertaken as the general knapsack problem has been proven to be NP-complete [3].

Group-based knapsack ciphers represent a novel approach for building knapsack-like encryption schemes, which is based on the properties of generating sets of finite groups and isomorphic transformations. The matrix-based knapsack cipher is a cryptosystem of the given class, which is constructed using the direct product of diagonal subgroups of a general linear group over a finite field [4, 5]. This encryption scheme, as well as its class, has been proposed in [4].

The purpose of the present work is to continue and deepen the matrix-based knapsack cipher research, which has been started in [5]. In the given paper two new results are proposed. The first one is represented by the proof that the considered encryption scheme is additively homomorphic. This property can be useful, since homomorphic ciphers are used in several fields of application such as electronic voting, secure multi-party computation and private information retrieval [6]. The second result is a brief description of a new secret electronic voting protocol, which is built on the basis of the matrix-based knapsack cipher.

2 The Matrix-based Knapsack Cipher

Let n and q be parameters of this cryptosystem. Consider a group G, which is the diagonal subgroup of general linear group GL(n, GF(q)). As the multiplication of diagonal matrices is commutative, G is an abelian group. Choose a generating set of G and represented it by a tuple (g_1, g_2, ..., g_n). A value of g_i is chosen as a diagonal matrix, where the i-th entry of the main diagonal is some primitive element z_i of GF(q) and other elements equal the corresponding entries of n-dimensional identity matrix. Thus, the generating set of G can be completely described by (z_1, z_2, ..., z_n). The multiplicative order for g_i equals q - 1, so for each d ∈ G there is a single integer tuple (x_1, x_2, ..., x_n) such that

$$d = g_1^{x_1} \cdot g_2^{x_2} \cdot ... \cdot g_n^{x_n}, \tag{1}$$

where $0 \le x_i < q - 1$ for all $i \le n$ [5].

For purpose of brevity, define $\text{ent}_i(x)$ to be the i-th element of the main diagonal of x ∈ GL(n, GF(q)). If b ∈ G, then $\text{ent}_i(b \cdot g_i) = z_i \cdot \text{ent}_i(b)$ and $\text{ent}_j(b \cdot g_i) = \text{ent}_j(b)$ for $j \neq i$. Therefore, $\text{ent}_i(d)$ equals z_i to the power of x_i. Hence,

$$x_i = \log_{z_i}(\text{ent}_i(d)), \tag{2}$$

where $\log_g(x)$ is a discrete logarithm in GF(q).

Select a secret value s ∈ GL(n, GF(q)) to define an isomorphism f: G → H and its inverse f^{-1}: H → G, where H is some subgroup of GL(n, GF(q)), as follows:

$$f: \ x \rightarrow s^{-1} \cdot x \cdot s,$$
$$f^{-1}: y \rightarrow s \cdot y \cdot s^{-1}.$$

Owing to (1) and the isomorphism f, any c ∈ H has a unique representation

$$c = e_1^{x_1} \cdot e_2^{x_2} \cdot ... \cdot e_n^{x_n}, \ e_i = f(g_i), \qquad (3)$$

where integer $x_i \in [0, q - 2]$ for all $i \le n$ [5].

The key generation procedure for the matrix-based knapsack cipher consists of choosing a private key s ∈ GL(n, GF(q)) and computation of a tuple $(e_1, e_2, ..., e_n)$, which is a public key [5].

The encryption lies in using (3) for obtaining a ciphertext c from a plaintext, which is represented by a nonnegative integer tuple $(x_1, x_2, ..., x_n)$, where all elements are less than q - 1 [5].

The decryption is performed as follows [5]:

1. A value of d is calculated using the formula $d = f^{-1}(c)$. A tuple $(z_1, z_2, ..., z_n)$, which is required for the next stage of decryption, is either stored along with a private key or computed by the formula $z_i = ent_i(f^{-1}(e_i))$.
2. A plaintext tuple is obtained in accordance with (2). Since discrete logarithms in GF(q) are computed at this stage, its efficient performance requires q - 1 to be smooth or small. In the current paper the last approach is accepted.

In absence of a correct private key the recovery of a plaintext from a ciphertext and a public key requires to solve (3) for an integer tuple $(x_1, x_2, ..., x_n)$, i.e. to compute a multidimensional discrete logarithm. There are no known algorithms for solving the general case of this problem in polynomial time by means of non-quantum computers [2]. Thus, to date the investigated cipher can be considered secure provided that its parameters are chosen properly.

A toy example of the matrix-based knapsack cipher, where q = 11 and n = 4, is presented below. In this case, G is a diagonal subgroup of GL(4, GF(11)), so a tuple (g_1, g_2, g_3, g_4) can be chosen in the following way:

$$g_1 = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ g_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ g_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ g_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 8 \end{pmatrix}.$$

Hence, (z_1, z_2, z_3, z_4) equals (2, 6, 7, 8). The multiplicative order for each g_i is 10, so the elements of a plaintext tuple must not exceed 9.

A private key s and its inverse s^{-1} are selected as follows:

$$s = \begin{pmatrix} 3 & 8 & 5 & 2 \\ 1 & 3 & 4 & 5 \\ 1 & 10 & 8 & 5 \\ 9 & 0 & 9 & 7 \end{pmatrix}, \ s^{-1} = \begin{pmatrix} 2 & 2 & 0 & 9 \\ 8 & 8 & 10 & 10 \\ 8 & 5 & 2 & 10 \\ 6 & 2 & 10 & 4 \end{pmatrix}.$$

The corresponding public key (e_1, e_2, e_3, e_4) is described in the following way:

$$e_1 = \begin{pmatrix} 7 & 5 & 10 & 4 \\ 2 & 10 & 7 & 5 \\ 2 & 9 & 8 & 5 \\ 7 & 4 & 8 & 2 \end{pmatrix},\ e_2 = \begin{pmatrix} 0 & 8 & 7 & 6 \\ 7 & 0 & 6 & 2 \\ 3 & 9 & 2 & 4 \\ 10 & 8 & 7 & 7 \end{pmatrix},\ e_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 5 & 7 & 7 & 3 \\ 1 & 10 & 9 & 5 \\ 5 & 6 & 7 & 4 \end{pmatrix},\ e_4 = \begin{pmatrix} 7 & 0 & 6 & 1 \\ 3 & 1 & 3 & 6 \\ 3 & 0 & 4 & 6 \\ 10 & 0 & 10 & 10 \end{pmatrix}.$$

A plaintext to be encrypted is (9, 1, 3, 4). A ciphertext c is obtained as follows:

$$c = e_1^9 \cdot e_2^1 \cdot e_3^3 \cdot e_4^4 = \begin{pmatrix} 9 & 0 & 3 & 6 \\ 0 & 2 & 6 & 1 \\ 10 & 8 & 4 & 7 \\ 9 & 7 & 4 & 3 \end{pmatrix}.$$

The decryption starts with computation of d as shown below:

$$d = s \cdot c \cdot s^{-1} = \begin{pmatrix} 6 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}.$$

The last stage of the decryption lies in obtaining the elements of the plaintext tuple from d in the following way:

$$x_1 = \log_{z_1}(\mathrm{ent}_1(d)) = \log_2(6) = 9,\ x_2 = \log_{z_2}(\mathrm{ent}_2(d)) = \log_6(6) = 1,$$
$$x_3 = \log_{z_3}(\mathrm{ent}_3(d)) = \log_7(2) = 3,\ x_4 = \log_{z_4}(\mathrm{ent}_4(d)) = \log_8(4) = 4.$$

Thus, the result of the decryption is (9, 1, 3, 4), which equals the original plaintext.

3 The Proof of Additive Homomorphism

A cipher is additively homomorphic if for all parameters and keys it possesses all of the following features [7, 8]:

1. A set of plaintexts with $\oplus$ and a ciphertexts set with $\otimes$ are additive and multiplicative abelian groups, respectively. The symbols $\oplus$ and $\otimes$ designate some binary operations.
2. If an encryption procedure converts m_1 to c_1 and m_2 to c_2, then the decryption of $c_1 \otimes c_2$ yields $m_1 \oplus m_2$.

For the matrix-based knapsack cipher define operations $\oplus$ and $\otimes$ as follows:

$$(u_1, ..., u_n) \oplus (v_1, ..., v_n) = ((u_1 + v_1) \bmod (q - 1), ..., (u_n + v_n) \bmod (q - 1))$$
$$c_1 \otimes c_2 = c_1 \cdot c_2$$

It is easy to show that the set of plaintexts and $\oplus$ constitute an additive abelian group. The ciphertexts set and $\otimes$ form the group H, which is isomorphic to the multiplicative abelian group G. Thus, the investigated cipher fulfills the first of the aforesaid necessary conditions of additive homomorphism.

Consider this cipher in the context of the second feature from the list above. Let the plaintexts be $m_1 = (u_1, u_2, ..., u_n)$ and $m_2 = (v_1, v_2, ..., v_n)$. The ciphertexts c_1 and c_2 are obtained by encryption of m_1 and m_2, respectively. As the group H is abelian, it follows from (3) that

$$c_1 \otimes c_2 = e_1^{u_1 + v_1} \cdot e_2^{u_2 + v_2} \cdot ... \cdot e_n^{u_n + v_n} . \qquad (4)$$

Since e_i is defined by (3), it has the same multiplicative order as g_i, i.e. q - 1. Thus, it is possible to transform (4) into the following expression:

$$c_1 \otimes c_2 = e_1^{(u_1 + v_1) \bmod (q - 1)} \cdot e_2^{(u_2 + v_2) \bmod (q - 1)} \cdot ... \cdot e_n^{(u_n + v_n) \bmod (q - 1)} \qquad (5)$$

In (5) each e_i is raised to the nonnegative power, which is less than q - 1. Therefore, the decryption of $c_1 \otimes c_2$ yields $((u_1 + v_1) \bmod (q - 1)$, ..., $(u_n + v_n) \bmod (q - 1))$, which is equal to $m_1 \oplus m_2$. So the second necessary condition from the aforesaid list is fulfilled by the considered cryptosystem.

Thus, the matrix-based knapsack cipher is additively homomorphic, as it has all the features from the list above.

4 Application in Secret Electronic Voting

Secret electronic voting systems can be built on the basis of additively homomorphic encryption schemes. However, such systems require using additional cryptographic tools, among which there are secret-sharing schemes and zero-knowledge proof protocols [9].

As the matrix-based knapsack cipher is additively homomorphic, it can be used to construct the secret e-voting protocol, which is proposed below. Secret sharing and zero-knowledge proof schemes are not considered in the current paper, since choosing them can be the subject of a separate research. The focus of this section is on the role, which the investigated cryptosystem plays in the described e-voting protocol. Using this cipher allows voters to keep their votes secret during the tallying process.

The proposed secret e-voting protocol consist of the preparatory, voting and tallying stages. The first stage comprises registering voters and candidates, choosing parameters of the used cryptographic tools, generating key data and distributing required information among the participants of a voting process. The second stage lies in collecting and validating the encrypted votes of authenticated users. The last stage is represented by tallying without decrypting the data submitted by voters.

The preparatory stage includes the following steps:

1. Voters and candidates are registered in the e-voting system. Let n_v and n_c be quantities of voters and candidates, respectively.
2. The tallying committee chooses positive integer number t_v, which determines the range of marks used by a voter, and publishes it. Accordingly, a vote is a tuple of marks given to candidates, where each element belongs to $\{0, 1, 2, ..., t_v\}$. The value of t_v should be small to provide computational efficiency, since the use of a large t_v leads to calculation of discrete logarithms in a large finite field during the tallying stage.
3. The parameters of the matrix-based knapsack cipher are chosen. The value of q must be no less than $t_v \cdot n_v + 2$ to ensure that any total mark given to a candidate is less than $q - 1$. The parameter n is chosen to equal $n_c + n_p$, where n_p is the length of a random padding tuple used to provide a probabilistic encryption of a vote. Consequently, an encrypted vote is a ciphertext corresponding to the plaintext obtained by concatenating a vote tuple with a random padding. The value of q to the power of n_p must be sufficiently large to provide semantic security against chosen-plaintext attack. Choosing the value of n_p can be considered in further research.
4. The public and private keys for the matrix-based knapsack cipher are generated. A secret sharing scheme is used to split the private key into n_t unique secret shares, where n_t is the number of members of tallying committee. Each member receives only one of these shares. No member must obtain the private key or a share given to another member. All n_t shares are required to recover the decryption key. After the secret shares are generated, the public key is published, whereas the private one is deleted as no one must know it.

The voting stage can be described as follows:

1. The voting server and a voter perform mutual authentication.
2. A voter gives to each candidate some nonnegative integer mark, which does not exceed t_v, and performs probabilistic encryption of his vote by means of the considered cipher using the public key of the voting system. Let $w_{i,j}$ denote the mark given by the i-th voter to the j-th candidate. The k-th element of the random padding tuple generated by the i-th voter is designated as $p_{i,k}$. The i-th encrypted vote c_i is obtained as the result of encryption of the plaintext

$$\left(w_{i,1}, w_{i,2}, ..., w_{i,n_c}, p_{i,1}, p_{i,2}, ..., p_{i,n_p}\right),$$

where $w_{i,j} \in \{0, 1, 2, ..., t_v\}$ for all $j \le n_c$, $p_{i,k} \in [0, q - 2]$ for all $k \le n_p$. Thus, it follows from (3) that

$$c_i = e_1^{w_{i,1}} \cdot e_2^{w_{i,2}} \cdot \ldots \cdot e_{n_c}^{w_{i,n_c}} \cdot e_{n_c+1}^{p_{i,1}} \cdot e_{n_c+2}^{p_{i,2}} \cdot \ldots \cdot e_n^{p_{i,n_p}} .$$

3. The value of c_i and proof of its correctness are sent to the voting server. This proof, which is generated by means of a non-interactive zero-knowledge range proof protocol, allows voting server to make sure that for c_i every $w_{i,j}$ is not greater than t_v.

The tallying stage consists of the following steps:

1. The voting server calculates h, which is defined as the product of all correct encrypted votes received on the previous stage, and sends its value to the tallying committee. Let n_r denote the number of such votes. It follows from (4) that

$$h = pw\left(e_1, \sum_{i=1}^{n_r} w_{i,1}\right) \cdot \ldots \cdot pw\left(e_{n_c}, \sum_{i=1}^{n_r} w_{i,n_c}\right) \cdot pw\left(e_{n_c+1}, \sum_{i=1}^{n_r} p_{i,1}\right) \cdot \ldots \cdot pw\left(e_n, \sum_{i=1}^{n_r} p_{i,n_p}\right),$$

where pw(x, y) denotes x^y. This formula can be transformed into the expression

$$h = e_1^{b_1} \cdot \ldots \cdot e_{n_c}^{b_{n_c}} \cdot e_{n_c+1}^{m_1} \cdot \ldots \cdot e_n^{m_{n_p}} , \qquad (6)$$

where b_j is total mark given to the j-th candidate and m_k is the sum of the k-th elements of all random padding tuples, which have been used for generating correct encrypted votes.

2. The members of the tallying committee give their secret shares, as well as the ciphertext h, to the voting system, which restores the private key, decrypts h and truncates the obtained plaintext tuple, leaving only the first n_c elements. The resulting tuple, which describes an outcome of the voting, is sent to the tallying committee. In accordance with (5) and (6), the decryption of h yields the tuple

$$\left(b_1 \bmod (q-1),\ \ldots, b_{n_c} \bmod (q-1),\ m_1 \bmod (q-1),\ \ldots,\ m_{n_p} \bmod (q-1)\right),$$

where $b_j < q - 1$ for all $j \leq n_c$, which follows from the way of choosing q on the third step of the preparatory stage. Thus, the resulting tuple received by the tallying committee is equal to

$$\left(b_1, b_2, \ldots, b_{n_c-1}, b_{n_c}\right).$$

The restored private key, as well as the received shares, are kept secret by the voting system. These data must be deleted after h is decrypted.

3. The tallying committee announces the result of the voting.

Consider a toy example of execution of this secret e-voting protocol, where 3 candidates are assessed by 7 voters, which give to each candidate some mark from the set {0, 1, 2, 3, 4, 5}. In this case, $n_v = 7$, $n_c = 3$ and $t_v = 5$.

The parameters for the underlying cipher are chosen as follows. The smallest possible value of q is $n_v \cdot t_v + 2 = 37$, since this number is prime. Therefore, let q be 37 as this choice is suitable for a toy example. The value of n_p is chosen to equal 2, so $n = n_c + n_p = 5$. Accordingly, for the given parameters of the matrix-based knapsack cipher the group G is a diagonal subgroup of GL(5, GF(37)). The elements of the tuple $(g_1, g_2, g_3, g_4, g_5)$ can be selected as follows:

$$g_1 = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; g_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; g_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$g_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; g_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 17 \end{pmatrix}.$$

Thus, $(z_1, z_2, z_3, z_4, z_5)$ is equal to (2, 5, 13, 15, 17).

A private key s and its inverse s^{-1} are chosen in the following way:

$$s = \begin{pmatrix} 24 & 15 & 21 & 13 & 15 \\ 9 & 22 & 18 & 35 & 8 \\ 11 & 27 & 3 & 12 & 5 \\ 24 & 21 & 26 & 32 & 11 \\ 19 & 20 & 36 & 16 & 26 \end{pmatrix}, \; s^{-1} = \begin{pmatrix} 32 & 7 & 26 & 36 & 36 \\ 12 & 2 & 1 & 16 & 4 \\ 11 & 36 & 17 & 17 & 2 \\ 7 & 28 & 4 & 10 & 35 \\ 19 & 33 & 14 & 0 & 36 \end{pmatrix}.$$

Hence, the public key $(e_1, e_2, e_3, e_4, e_5)$ is described as follows:

$$e_1 = \begin{pmatrix} 29 & 36 & 6 & 9 & 36 \\ 29 & 33 & 30 & 8 & 32 \\ 5 & 17 & 10 & 32 & 17 \\ 20 & 31 & 36 & 18 & 31 \\ 12 & 26 & 29 & 25 & 27 \end{pmatrix}, \; e_2 = \begin{pmatrix} 31 & 24 & 23 & 18 & 2 \\ 35 & 29 & 33 & 21 & 27 \\ 1 & 23 & 3 & 8 & 5 \\ 9 & 22 & 18 & 36 & 8 \\ 4 & 18 & 8 & 32 & 21 \end{pmatrix},$$

$$e_3 = \begin{pmatrix} 29 & 25 & 11 & 7 & 6 \\ 21 & 29 & 36 & 33 & 23 \\ 24 & 32 & 21 & 6 & 21 \\ 10 & 1 & 33 & 22 & 18 \\ 35 & 22 & 23 & 18 & 27 \end{pmatrix}, \; e_4 = \begin{pmatrix} 35 & 2 & 6 & 33 & 31 \\ 11 & 6 & 15 & 27 & 22 \\ 14 & 3 & 10 & 31 & 28 \\ 30 & 17 & 14 & 4 & 23 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$e_5 = \begin{pmatrix} 30 & 13 & 16 & 3 & 28 \\ 32 & 23 & 10 & 25 & 36 \\ 16 & 11 & 6 & 31 & 18 \\ 21 & 26 & 32 & 7 & 19 \\ 29 & 13 & 16 & 3 & 29 \end{pmatrix}.$$

The value of s is split into unique secret shares, which are distributed among of the members of tallying committee. The values of s and s^{-1} are deleted as soon as all secret shares are generated.

The first voter gives marks 2, 4 and 5 to the first, second and third candidates, respectively. The random values used to encrypt the vote are 27 and 34. Accordingly, $w_1 = (2, 4, 5)$ and $p_1 = (27, 34)$. The encrypted vote c_1 is obtained as follows:

$$c_1 = e_1^2 \cdot e_2^4 \cdot e_3^5 \cdot e_4^{27} \cdot e_5^{34} = \begin{pmatrix} 3 & 20 & 6 & 13 & 10 \\ 0 & 5 & 14 & 0 & 1 \\ 5 & 0 & 26 & 22 & 35 \\ 33 & 10 & 2 & 1 & 14 \\ 3 & 20 & 8 & 25 & 27 \end{pmatrix}.$$

The actions of another voters can be briefly described in the following way:

$$w_2 = (0,0,2), p_2 = (13,17), w_3 = (5,5,4), p_3 = (3,29),$$
$$w_4 = (0,1,0), p_4 = (2,3), w_5 = (3,2,1), p_5 = (31,5),$$
$$w_6 = (4,3,0), p_6 = (25,15), w_7 = (0,5,0), p_7 = (11,35),$$
$$c_2 = \begin{pmatrix} 5 & 7 & 5 & 1 & 7 \\ 29 & 4 & 35 & 6 & 30 \\ 34 & 15 & 34 & 0 & 31 \\ 26 & 9 & 24 & 34 & 16 \\ 3 & 31 & 1 & 23 & 16 \end{pmatrix}, \; c_3 = \begin{pmatrix} 8 & 27 & 10 & 36 & 16 \\ 28 & 6 & 24 & 17 & 27 \\ 14 & 36 & 25 & 0 & 22 \\ 10 & 3 & 8 & 35 & 32 \\ 29 & 26 & 27 & 19 & 22 \end{pmatrix},$$
$$c_4 = \begin{pmatrix} 6 & 14 & 36 & 24 & 29 \\ 8 & 18 & 13 & 25 & 35 \\ 31 & 7 & 17 & 31 & 22 \\ 17 & 25 & 2 & 2 & 30 \\ 27 & 13 & 36 & 28 & 33 \end{pmatrix}, \; c_5 = \begin{pmatrix} 12 & 21 & 15 & 7 & 33 \\ 28 & 22 & 35 & 13 & 28 \\ 25 & 24 & 28 & 12 & 28 \\ 30 & 30 & 1 & 11 & 34 \\ 23 & 26 & 33 & 23 & 9 \end{pmatrix},$$

$$c_6 = \begin{pmatrix} 7 & 4 & 13 & 9 & 31 \\ 6 & 24 & 24 & 6 & 33 \\ 9 & 32 & 22 & 23 & 15 \\ 29 & 23 & 15 & 24 & 36 \\ 20 & 22 & 30 & 12 & 7 \end{pmatrix},\ c_7 = \begin{pmatrix} 29 & 35 & 17 & 16 & 26 \\ 31 & 9 & 17 & 5 & 7 \\ 8 & 1 & 1 & 15 & 31 \\ 4 & 25 & 25 & 22 & 2 \\ 23 & 19 & 18 & 19 & 1 \end{pmatrix}.$$

After all encrypted votes are received, the voting server computes h as shown below:

$$h = \prod_{i=1}^{7} c_i = \begin{pmatrix} 35 & 30 & 29 & 20 & 3 \\ 12 & 1 & 30 & 0 & 8 \\ 8 & 16 & 6 & 21 & 4 \\ 34 & 11 & 27 & 34 & 18 \\ 1 & 28 & 30 & 19 & 33 \end{pmatrix}.$$

The voting system obtains the secret shares, as well as the value of h, from the tallying committee, restores the private key s and decrypts the ciphertext h. The first stage of the decryption lies in calculation of d in the following way:

$$d = s \cdot h \cdot s^{-1} = \begin{pmatrix} 30 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 11 \end{pmatrix}.$$

The decryption ends with obtaining the plaintext tuple (x_1, x_2, x_3, x_4, x_5) as follows:

$$x_1 = \log_{z_1}(\mathrm{ent}_1(d)) = \log_2(30) = 14,\ x_2 = \log_{z_2}(\mathrm{ent}_2(d)) = \log_5(12) = 20,$$
$$x_3 = \log_{z_3}(\mathrm{ent}_3(d)) = \log_{13}(10) = 12,\ x_4 = \log_{z_4}(\mathrm{ent}_4(d)) = \log_{15}(9) = 4.$$
$$x_5 = \log_{z_5}(\mathrm{ent}_5(d)) = \log_{17}(11) = 30.$$

The plaintext truncated to the first 3 elements, which represents the outcome of the voting, is sent to the tallying committee. Its members receive the tuple (14, 20, 12) and publish it as the voting results. Since $w_1 + ... + w_7 = (14, 20, 12)$, the published total marks are correct.

5 Conclusion

The property of additive homomorphism, which is proven for the matrix-based knapsack cipher in the current work, allows this encryption scheme to be used in several fields of application, which include but are not limited to key agreement in hybrid cryptosystems. In particular, a new secret electronic voting protocol constructed on the basis of the given cipher is proposed in the given paper in the form of a brief description. This protocol allows a voter to assess each candidate by means of using nonnegative integer marks.

Future research can be focused on choosing the parameters for the considered cryptosystem, which would provide 128-bit and 256-bit security levels. Another possible research direction encompasses the probabilistic encryption by means of the given cipher in the context of the proposed secret e-voting protocol.

References

1. Schneier, B.: Applied Cryptography: Protocols, Algorithms and Source Code in C. 20th edn. John Wiley & Sons (2015)
2. ETSI White Paper No. 8. Quantum Safe Cryptography and Security: An introduction, benefits, enablers and challenges. European Telecommunications Standards Institute (2015).
3. Van Tilborg, H.: An Introduction to Cryptology. Springer (2012)
4. Zhivotova, A., Ziuliarkina, N., Kostygina, Y.: Modification of the Cryptosystem with Public Key on the Basis of Knapsack Problem. UrFR Newsletter. Information Security 1 (11), pp. 16-20 (in Russian) (2014)
5. Vambol, A.: The Prospects for Group-based Knapsack Ciphers in the Post-Quantum Era. In: 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT), pp. 271-275 (2018)
6. Liu, J., Mesnager, S., Chen, L.: Partially homomorphic encryption schemes over finite fields. In: the 6th International Conference on Security, Privacy and Applied Cryptography Engineering (SPACE 2016), pp. 109-123 (2016)
7. Bendlin, R., Damgård, I., Orlandi, C., Zakarias, S.: Semi-Homomorphic Encryption and Multiparty Computation. In: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011), pp. 169-188 (2011)
8. Barshap, G., Tassa, T.: Privacy-Preserving Planarity Testing of Distributed Graphs. In: Data and Applications Security and Privacy XXXII (DBSec 2018), pp. 131-147 (2018)
9. The Future of Voting: End-to-End Verifiable Internet Voting Specification and Feasibility Assessment Study. U.S. Vote Foundation (2015)