Filomat 40:1 (2026), 143–152
https://doi.org/10.2298/FIL2601143A

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Matrix-based cryptography: The characteristic product

Emin Aygün[a,*], İncinur Yılmaz[b]

[a]*Department of Mathematics, Faculty of Sciences, Erciyes University, 38030 Kayseri, Turkey*
[b]*Department of Mathematics, Graduate School of Natural and Applied Science, Erciyes University 38030 Kayseri, Turkey*

Abstract. For many years, the need to store information and data has been a fundamental challenge for human society, driving the development of various systems to meet this requirement. In this study, we focus on the encryption and decryption of soft sets, aligning with their specific representations. To achieve this, we employ the Rijndael algorithm and extend it by introducing encryption and decryption algorithms that utilize the characteristic multiplication of soft matrices. The proposed methods were tested on several examples, and the experimental results confirm that the plaintexts are securely encrypted and accurately recovered without any loss of information. These results demonstrate the effectiveness and reliability of the proposed approach. Moreover, the proposed approach highlights the flexibility of soft matrices in handling sensitive and uncertain data, thereby providing a mathematically robust foundation for cryptographic applications.

1. Introduction

Cryptography has served as an essential tool for securing communication for centuries, enabling data to be transmitted privately so that only the intended recipient can interpret it. Cryptosystems are generally classified into two types: symmetric and asymmetric. The former relies on a single key for both encryption and decryption, while the latter employs two distinct keys for these processes [6, 8–12]. In both systems, a key is used to encrypt data, while its counterpart key facilitates the decryption of the encrypted data.

Symmetric cryptosystems consist of two main types of ciphers: stream ciphers and block ciphers. The distinction lies primarily in their processing approach: stream ciphers encrypt data one bit at a time, whereas block ciphers process data in fixed-size blocks [1, 3, 13].

This paper explores the application of the Vernam cipher as a stream cipher for encrypting a soft set associated with uncertain objects, aiming to mitigate certain complexities. The motivation for encrypting soft sets arises from the sensitive information that these sets encompass. Traditional theories addressing uncertainties, such as fuzzy set theory, probability theory [14, 15], and interval mathematics [7] have been applied in various fields, including engineering, environmental studies, and economics.

However, each of these theories faces limitations in handling diverse types of uncertainty, and none can adequately address minor variations in data-specific information. Furthermore, each theory possesses

2020 *Mathematics Subject Classification*. Primary 94A60; Secondary 15A99, 68P25.
Keywords. encryption, decryption, soft sets, soft matrix, cryptography.
Received: 04 August 2025; Revised: 19 September 2025; Accepted: 25 September 2025
Communicated by Predrag Stanimirović
* Corresponding author: Emin Aygün
Email addresses: eaygun@erciyes.edu.tr (Emin Aygün), incinur.yilmaz@hotmail.com (İncinur Yılmaz)
ORCID iDs: https://orcid.org/0000-0003-3503-0552 (Emin Aygün), https://orcid.org/0000-0001-6481-8918 (İncinur Yılmaz)

intrinsic complexities that may conflict with the goals of effective uncertainty management, thereby limiting its reliability and applicability in sensitive contexts.

In this paper, our main contribution is the development of encryption and decryption algorithms based on the characteristic multiplication of soft matrices. Unlike traditional cryptosystems that rely on classical algebraic structures, the proposed approach integrates soft set representations with binary transformations, enhancing flexibility and diversity in handling uncertain or sensitive data. This distinction sets our work apart from earlier studies [4, 5] by extending the applicability of matrix-based cryptography to soft sets through the use of characteristic products and transpositions.

The remainder of this paper is organized as follows. Section 2 presents preliminary definitions and concepts related to soft sets and soft matrices. Section 3 introduces the proposed cryptosystem model, along with encryption and decryption algorithms illustrated by examples. Section 4 discusses the conclusions, while Section 5 provides the experimental results and validates the feasibility of the suggested method. Therefore, the proposed approach not only extends the theoretical framework of soft set-based cryptography but also offers a practical scheme for secure communication in environments characterized by uncertainty.

2. Preliminaries

Definition 2.1 ([11]). *Let U be a universal set, E the set of parameters, and $P(U)$ the power set of U, with $A \subseteq E$. Suppose there exist a soft set over universal set U, defined as $f_A : E \to P(U)$, such that if $e_j \notin A$, then $f_A(e_j)$ is specified by F_A as $\emptyset$. Thus, the soft set F_A is given by:*

$$F_A = \{(e_j, f_A(e_j)) : e_j \in E, f_A(e_j) \in P(U)\}. \tag{1}$$

In this context, f_A is referred to as the approximation function of F_A.

Definition 2.2 ([2]). *Let $U = \{u_1, u_2, u_3, \ldots, u_n\}$ be a universal set, $E = \{e_1, e_2, e_3, \ldots, e_m\}$ a set of parameters, with $A \subseteq E$, and F_A be a soft set over U. The relational form of F_A is given by:*

$$R_A = \{(u_i, e_j) : e_j \in A, u_i \in F_A(e_j)\} \subseteq U \times E. \tag{2}$$

In this context, the characteristic function of the relation R_A is defined by:

$$\chi_{R_A} : U \times E \to \{0, 1\}, \tag{3}$$

where

$$\chi_{R_A}(u_i, e_j) = \begin{cases} 1, & (u_i, e_j) \in R_A \\ 0, & (u_i, e_j) \notin R_A \end{cases}. \tag{4}$$

The relation R_A can be represented in the following tabular form, where $U = \{u_1, u_2, u_3, \ldots, u_n\}$ denotes the universal set, $E = \{e_1, e_2, e_3, \ldots, e_m\}$ is the set of parameters, and $A \subseteq E$.

R_A	e_1	e_2	$\ldots$	e_m
u_1	$\chi_{R_A}(u_1, e_1)$	$\chi_{R_A}(u_1, e_2)$	$\ldots$	$\chi_{R_A}(u_1, e_m)$
u_2	$\chi_{R_A}(u_2, e_1)$	$\chi_{R_A}(u_2, e_2)$	$\ldots$	$\chi_{R_A}(u_2, e_m)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
u_n	$\chi_{R_A}(u_n, e_1)$	$\chi_{R_A}(u_n, e_2)$	$\ldots$	$\chi_{R_A}(u_n, e_m)$

Definition 2.3 ([4]). *Let $[a_{ij}], [b_{ij}] \in S_{m\times n}$. The characteristic product of the soft matrices $[a_{ij}]$ and $[b_{ij}]$ is as:*

$$[a_{ij}] \cdot_c [b_{ij}] = [c_{ij}], \tag{5}$$

where the elements $c_{i,j}$ are given by:

$$c_{i,j} = \begin{cases} 1, & \text{if } a_{i,j} = b_{i,j}, \\ 0, & \text{if } a_{i,j} \neq b_{i,j} \end{cases}. \tag{6}$$

Definition 2.4 ([4]). *Each symbol (including Turkish-specific characters) be associated with a unique numerical value. This mapping, which assigns values starting from 0, is detailed in Table 1.*

Table 1: Correspondence between Symbols and Numbers

SYMBOLS	A	B	C	D	E	F	G	H	I	J	K	L
NUMBERS	0	1	2	3	4	5	6	7	8	9	10	11
SYMBOLS	M	N	O	P	Q	R	S	T	U	V	W	X
NUMBERS	12	13	14	15	16	17	18	19	20	21	22	23
SYMBOLS	Y	Z	Ç	Ğ	İ	Ö	Ş	Ü	0	1	2	3
NUMBERS	24	25	26	27	28	29	30	31	32	33	34	35
SYMBOLS	4	5	6	7	8	9	:	;	=	?	!	″
NUMBERS	36	37	38	39	40	41	42	43	44	45	46	47
SYMBOLS	′	*	.	,	-	/	@	+	(	)	[	]
NUMBERS	48	49	50	51	52	53	54	55	56	57	58	59
SYMBOLS	{	}	%	&								
NUMBERS	60	61	62	63								

Furthermore, each letter can be uniquely represented by its binary equivalent. The mapping to binary form, using 6-bit representation, is shown in Table 2.

Table 2: Binary System Correspondence for Symbols

SYMBOLS	A	B	...	Ü	0	⋯	%	&
BINARY SYSTEM	000000	000001	...	011111	100001	⋯	111110	111111

Accordingly, the symbol A corresponds to 0 (000000 in binary), while the symbol & corresponds to 63 (111111 in binary). The entire alphabet is thus encoded within the binary range from `000000` *to* `111111`.

To maintain a complete 6-bit binary representation (i.e., $2^6 = 64$ combinations), the Turkish-specific letters Ç, Ğ, İ, Ö, Ş, Ü and frequently used symbols are included. If additional characters be incorporated, the binary representation length must be increased to 7 bits, yielding up to 128 unique encodings.

Throughout this paper, the following notations are used. The universe set is denoted by U, while E stands for the set of parameters, and $A \subseteq E$ indicates a subset of parameters. A soft set over U with parameter set A is represented by F_A. In the cryptographic framework, P denotes the plaintext, C the ciphertext, and S the keytext (serving as the encryption/decryption key represented in textual form) employed in the encryption and decryption processes. For block representations, $P_1, P_2, \ldots$ refer to the divided plaintext blocks, and $C_1, C_2, \ldots$ denote the corresponding ciphertext blocks. A soft matrix is expressed in the general form $[a_{ij}]$, where a_{ij} is the entry in the i-th row and j-th column, and $[a_{ij}]^T$ denotes the transpose of a soft matrix. The operator $\cdot_c$ designates the characteristic product of soft matrices.

3. A New Cryptosystem Model

In this section, encryption and decryption algorithms based on characteristic multiplication are presented and illustrated with examples.

Let $[p_{ij}]_{m\times m}$ represent the plaintext matrix and $[s_{ij}]_{m\times m}$ denote the key matrix, which are both square matrices. The transposes of these matrices are denoted by $[p_{ij}]^T_{m\times m}$ and $[s_{ij}]^T_{m\times m}$, respectively.

Encryption and Decryption With Characteristic Multiplication

Definition 3.1 ([5]). *The encryption process is performed as follows:*

$$[p_{ij}]^T_{m\times n} \cdot_c [s_{ij}]^T_{m\times n} = [c_{ij}]^T_{m\times n} \tag{7}$$

By transposing $[c_{ij}]^T_{m\times n}$, the original ciphertext matrix $[c_{ij}]_{m\times n}$ is recovered. The encrypted matrix that is sent to the receiver is $[c_{ij}]_{m\times n}$.

Definition 3.2 ([5]). *The decryption process follows a procedure similar to that of encryption:*

$$[c_{ij}]^T_{m\times n} \cdot_c [s_{ij}]^T_{m\times n} = [p_{ij}]^T_{m\times n} \tag{8}$$

The transpose of the plaintext matrix $[p_{ij}]^T_{m\times n}$ is obtained. By transposing this matrix, the original plaintext matrix $[p_{ij}]_{m\times n}$ is recovered. Thus, the decrypted version of the encrypted matrix obtained by the receiver is $[p_{ij}]_{m\times n}$.

Encryption Algorithm

The soft encryption algorithm proceeds as follows, with each block undergoing a specific transformation before the following steps:

1. The message is partitioned into discrete blocks, and each line is mapped to its equivalent representation in the binary numeral system.
2. The plaintext and key text divided into blocks, are converted into soft matrices.
3. The transposes of the matrices are taken.
4. The transposed matrices are multiplied characteristically.
5. The result of the multiplication is transposed.
6. The matrix is converted into text and delivered to the recipient.

Decryption Algorithm

The soft decryption algorithm proceeds as follows: each block undergoes a specific transformation to retrieve the original message. The procedure continues with the following steps:

1. The message is divided into blocks.
2. The message and the key are converted into soft matrices respectively.
3. The transposes of the encrypted text matrix and the key matrix are taken.
4. The transposed matrices are multiplied with each other in a characteristic manner.
5. The transpose of the resulting matrix is calculated.
6. The transposed matrix is converted back into text to retrieve the plaintext message.

Example 3.3. *Encrypt and decrypt the plaintext "TIME HEALS ALL" according to the algorithm using the keyword "WISDOM".*

Encryption process:

First, let's break the word "TIME HEALS ALL" into blocks. Then we get "TIMEHE - ALSALL".

Subsequently, the plaintext and key text are converted into a soft matrix.

$$[p_{ij}] = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{9}$$

$$[s_{ij}] = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

Next, the plaintext and key matrices are transposed.

$$[p_{ij}]^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{11}$$

$$[s_{ij}]^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{12}$$

Next, multiply the transposed plaintext matrix and the transposed key matrix with each other characteristically.

$$[p_{ij}]^T \cdot_c [s_{ij}]^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \cdot_c \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{13}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

$$= [c_{ij}]^T \tag{15}$$

is obtained. If we take $[c_{ij}]^{T}$*'s transpose, then we get* $[c_{ij}]$*. As a result, the encrypted text matrix is obtained as:*

$$[c_{ij}] = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{16}$$

As a result, the encrypted text separated into blocks such as "`[&1(@+ – 9{&{[(`*", is obtained. In this case, the message delivered to the recipient is "*`[&1( @+9{& {[)`*".*

Decryption Process:

First, let's break the word "`[&1( @+9{& {[)`*" into blocks. This yields "*`[&1(@+ – 9{&{[(`*".*

Let us transform the ciphertext and the key text into a soft matrix.

$$[c_{ij}] = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

$$[s_{ij}] = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{18}$$

Transpose the ciphertext and key matrices.

$$[c_{ij}]^{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

$$[s_{ij}]^{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{20}$$

The transposed plaintext matrix and the key matrix are multiplied characteristically. Then

$$[c_{ij}]^T \cdot_c [s_{ij}]^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot_c \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{21}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{22}$$

$$= [p_{ij}]^T \tag{23}$$

is obtained. If we take $[p_{ij}]^T$*'s transpose, then we get* $[p_{ij}]$*. As a result, the decrypted text matrix is obtained as:*

$$[p_{ij}] = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{24}$$

As a result, the decrypted text separated into "TIMEHE - ALSALL" blocks is obtained. In this case, the message that needs to be given to the recipient is "TIME HEALS ALL".

Definition 3.4. *The encryption process is performed as follows:*

$$[p_{ij}]_{m\times n} \cdot_c [s_{ij}]^T_{m\times m} = [c_{ij}]_{m\times n} \tag{25}$$

The encrypted matrix that is sent to the receiver is $[c_{ij}]_{m\times n}$*.*

Definition 3.5. *The decryption process follows a procedure similar to that of encryption:*

$$[c_{ij}]_{m\times n} \cdot_c [s_{ij}]^T_{m\times m} = [p_{ij}]_{m\times n} \tag{26}$$

Thus, the decrypted version of the encrypted matrix received by the receiver is $[p_{ij}]_{m\times n}$*.*

Encryption Algorithm:

The soft encryption algorithm proceeds as follows: each block undergoes a specific transformation for use in the encryption process. The procedure continues with the following steps:

1. The message is partitioned into discrete blocks, and each line is mapped to its equivalent representation in the binary numeral system.
2. The text and key text divided into blocks are converted into a soft matrix.
3. The transpose of the key text matrix is taken.
4. The matrices are multiplied with each other in a characteristic manner.
5. The matrix is converted into text and delivered to the recipient.

Decryption Algorithm:
The soft decryption algorithm proceeds as follows: each block undergoes a specific transformation to retrieve the original message. The procedure continues with the following steps:

1. The message is divided into blocks.
2. The message and the key are converted into soft matrices respectively.
3. The transpose of the key text matrix is taken.
4. The matrices are multiplied with each other in a characteristic manner.
5. The transposed matrix is converted back into text to retrieve the plaintext message.

Example 3.6. *Encrypt and decrypt the plaintext "TIME HEALS ALL" according to the algorithm using the keyword "WISDOM".*

Encryption Process:

The encrypted text "=Ü0Y W&&I% IÇ'*" is derived from Definition 3.4 and Example 3.3.*

Decryption Process:

The decrypted text "TIME HEALS ALL" is obtained from Definition 3.5 and Example 3.3.

Definition 3.7. *The encryption process is performed as follows:*

$$[p_{ij}]^T_{m\times n} \cdot_c [s_{ij}]_{m\times m} = [c_{ij}]_{m\times n} \tag{27}$$

The encrypted matrix that is sent to the receiver is $[c_{ij}]_{m\times n}$.

Definition 3.8. *The decryption process follows a procedure similar to that of encryption:*

$$[c_{ij}]_{m\times n} \cdot_c [s_{ij}]_{m\times m} = [p_{ij}]^T_{m\times n} \tag{28}$$

The transpose of the plaintext matrix $[p_{ij}]^T_{m\times n}$ *is obtained. By transposing this matrix, the original plaintext matrix* $[p_{ij}]_{m\times n}$ *is recovered. Thus, the decrypted version of the encrypted matrix received by the receiver is* $[p_{ij}]_{m\times n}$.

Encryption Algorithm:
The soft encryption algorithm proceeds as follows, with each block undergoing a specific transformation before the following steps:

1. The message is partitioned into discrete blocks, and each line is mapped to its equivalent representation in the binary numeral system.
2. The text and key text divided into blocks are converted into a soft matrix.
3. The transpose of the plaintext matrix is taken.
4. The matrices are multiplied with each other in a characteristic manner.
5. The matrix is converted into text and delivered to the recipient.

Decryption Algorithm:
The soft decryption algorithm proceeds as follows: each block undergoes a specific transformation to retrieve the original message. The procedure continues with the following steps:

1. The message is divided into blocks.
2. The message and the key are converted into soft matrices respectively.
3. The matrices are multiplied with each other in a characteristic manner.
4. The transpose of the resulting matrix is calculated.
5. The transposed matrix is converted back into text to retrieve the plaintext message.

Example 3.9. *Encrypt and decrypt the plaintext "TIME HEALS ALL" according to the algorithm using the keyword "WISDOM".*

Encryption Process:

The encrypted text "`9X/' TR9&% {:0`*" is derived based on Definition 3.7 and Example 3.3.*

Decryption Process:

The decrypted text "TIME HEALS ALL" is derived based on Definition 3.8 and Example 3.3.

4. Limitations and Future Work

Although the proposed cryptosystem, based on the characteristic multiplication of soft matrices, has demonstrated secure and accurate encryption–decryption performance, certain limitations must be acknowledged. First, the method requires constructing soft matrices with predetermined dimensions, which may limit its efficiency when applied to large-scale or real-time data. Moreover, the current framework has been validated only through illustrative examples, and more extensive experiments on practical datasets are required to evaluate its robustness against various types of cryptanalytic attacks. Furthermore, key management and scalability remain open issues, as the use of binary encodings and matrix transpositions may introduce computational overhead for larger alphabets or extended symbol sets.

These limitations point to several promising directions for future research. One possible extension is the integration of the proposed approach with modern block cipher structures to improve its performance on high-dimensional data. Another promising direction is to explore hybrid schemes that combine soft set-based transformations with machine learning techniques for adaptive security. Finally, further theoretical analysis could strengthen the mathematical foundations of the method and extend its applicability to other uncertainty-oriented frameworks, such as fuzzy sets or rough sets.

5. Conclusions

In this study, we proposed encryption and decryption algorithms based on the characteristic multiplication of soft matrices. The major contribution of this work is the extension of matrix-based cryptography to the soft set framework, providing a mathematically tractable method for managing uncertain and sensitive data. By employing binary encodings and matrix transpositions, the proposed cryptosystem ensures the accurate recovery of plaintext while enhancing the flexibility and diversity of ciphertext generation. Compared to existing approaches, this integration of soft sets with characteristic matrix operations represents a novel direction, bridging algebraic theory and practical cryptographic applications.

Beyond the current scope, several promising directions for future research emerge. The approach can be further optimized for large-scale and real-time data encryption, where computational efficiency is critical. Another extension involves integrating the method with modern block cipher architectures or hybrid cryptosystems to strengthen its resistance against advanced cryptanalytic techniques. Additionally, theoretical generalizations to fuzzy or rough set frameworks could broaden its applicability to more diverse, uncertainty-driven environments. These perspectives emphasize that the present work not only contributes a novel cryptographic framework but also opens pathways for further advancements in secure communication systems.

6. Results

The developed encryption and decryption algorithms, based on the characteristic multiplication of soft matrices, were successfully implemented in several examples. The experimental results confirm that the proposed cryptosystem can securely encrypt and accurately decrypt the given plaintexts. Moreover,

the matrices generated from the soft sets were effectively transformed through characteristic product operations, thereby validating the theoretical framework. The division of plaintext into blocks and its precise reconstruction confirm the feasibility and soundness of the proposed method.

To further validate the significance of the proposed method, we compare its results with state-of-the-art cryptographic approaches. In contrast to the Rijndael algorithm (AES, Advanced Encryption Standard), which operates on fixed-size blocks and requires complex substitution–permutation networks, our method relies on the characteristic multiplication of soft matrices, offering a simpler structure while still ensuring secure encryption and decryption. Compared to the classical Vernam cipher, which is limited to bitwise XOR (exclusive OR) operations, the proposed approach introduces matrix-based transformations that enhance robustness and flexibility in handling data uncertainties. Furthermore, unlike the soft matrix product cryptosystem introduced in [4], our approach incorporates binary encodings and transpositions, enhancing both the diversity of the generated ciphertexts and the accuracy of decryption.

These comparisons clearly demonstrate that the proposed approach offers a mathematically tractable and secure alternative to existing methods, particularly in contexts requiring the management of uncertain or sensitive data.

Funding: This research received no external funding.

Author contributions: The conceptualization of the study was jointly developed by İncinur Yılmaz and Emin Aygün. Methodology design, formal analysis, and investigation were conducted by İncinur Yılmaz. The initial draft of the manuscript was prepared by İncinur Yılmaz, while both İncinur Yılmaz and Emin Aygün contributed to the critical review and thorough editing of the manuscript. Project supervision and oversight were provided by Emin Aygün. All authors have read and approved the final version of the manuscript and agreed to its submission for publication.

Acknowledgments: The authors express their sincere gratitude to all reviewers for their insightful suggestions, constructive feedback, and valuable contributions, which have greatly enhanced the quality and clarity of this study.

Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. I. Ali, F. Feng, X. Liu, W. K. Min, M. Shabir, *On some new operations in soft set theory*, Comput. Math. Appl. **57**(9) (2009), 1547–1553.
[2] A. O. Atagün, H. Kamacı, O. Oktay, *Reduced soft matrices and generalized products with applications in decision making*, Neural Comput. Appl. **29** (2018), 445–456.
[3] A. O. Atagün, A. Sezgin, *Soft substructures of rings, fields and modules*, Comput. Math. Appl. **61**(3) (2011), 592–601.
[4] E. Aygün, *Soft matrix product and soft cryptosystem*, Filomat **32**(19) (2018), 6519–6530.
[5] E. Aygün, İ. Yılmaz, *Polynomial representation of Vernam cipher*, Cumhuriyet Sci. J. **45**(3) (2024), 557–561.
[6] F. Feng, Y. Li, B. Davvaz, K. Qin, *Soft sets combined with rough sets applied to a decision-making problem*, Appl. Soft Comput. **10**(2) (2010), 1041–1047.
[7] M. B. Gorzalczany, *A method of inference in approximate reasoning based on interval-valued fuzzy sets*, Fuzzy Sets Syst. **21** (1987), 1–17.
[8] H. Aktaş, N. Çağman, *Soft sets and soft groups*, Inform. Sci. **177** (2007), 2726–2735.
[9] H. Aktaş, N. Çağman, *Erratum to "Soft Sets and Soft Groups"*, Inform. Sci. **179** (2009), 338.
[10] P. K. Maji, R. Biswas, A. R. Roy, *Soft set theory*, Comput. Math. Appl. **45**(4–5) (2003), 555–562.
[11] D. A. Molodtsov, *Soft set theory—First results*, Comput. Math. Appl. **37**(4–5) (1999), 19–31.
[12] G. Oğuz, İ. İçen, M. H. Gürsoy, *Actions of soft groups*, Commun. Fac. Sci. Univ. Ankara Ser. A1 Math. Stat. **68**(1) (2019), 1163–1174.
[13] A. R. Roy, P. K. Maji, *A fuzzy soft set theoretic approach to decision making problems*, J. Comput. Appl. Math. **203**(2) (2007), 412–418.
[14] L. A. Zadeh, *Fuzzy sets*, Inform. Control **8** (1965), 338–353.
[15] L. A. Zadeh, *Toward a generalized theory of uncertainty (GTU)—An outline*, Inform. Sci. **172** (2005), 1–40.