

Article

A Block Encryption Construction Based on Bi-Periodic Pell Matrices and Elliptic Curve Group Operations

Ersen Akinci * and Sukran Uygun *

Department of Mathematics, Science and Art Faculty, Gaziantep University, Gaziantep 27310, Turkey

* Correspondence: ersenakinci33@gmail.com (E.A.); suygun@gantep.edu.tr (S.U.)

Abstract

This paper presents a theoretical algebraic block transformation based on bi-periodic Pell matrices and elliptic-curve point-valued blocks. The construction uses block-dependent scalar key matrices $K_i = \tilde{P}_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q)$ to transform plaintext blocks whose entries lie in a selected elliptic-curve subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ of prime order q . The encryption operation is defined by the elliptic-curve scalar matrix action $M_i = B_i \star K_i$, where the entries of K_i act as scalar coefficients modulo q , and point additions are performed in G . The main purpose of the construction is to formalize the interaction between recursive Pell-type matrix sequences over $\mathbb{F}_q$ and elliptic-curve point-valued block transformations. The algebraic correctness follows from the invertibility of K_i over $\mathbb{F}_q$ and the compatibility of the $\star$ -action with the matrix multiplication. The deterministic version is not claimed to provide semantic security or IND-CPA security without additional randomization, nonce-dependent masking, and formal security proofs.

Keywords: elliptic curve cryptography; bi-periodic Pell sequence; bi-periodic Pell matrix sequence

MSC: 68P30; 11T71; 14G50; 11B39; 11B83; 11B37

1. Introduction

Elliptic curve cryptography was independently introduced by Koblitz and Miller as a public-key cryptographic framework based on the elliptic curve discrete logarithm problem [1,2]. Since then, elliptic curves have been widely used in key exchange, public-key encryption, digital signatures, and hybrid cryptographic protocols [3–5]. In many ECC-based hybrid schemes, the elliptic curve component is mainly used for key establishment or key encapsulation, whereas the actual data encryption is performed by a separate symmetric encryption mechanism [6]. In contrast, the present study uses elliptic curve points directly as plaintext and ciphertext entries. Thus, the elliptic curve group is not only used for a key-related component, but also as an algebraic domain in which the block transformation is performed.

Matrix-based cryptographic constructions originate from the classical Hill cipher, in which plaintext vectors are transformed by an invertible matrix over a finite alphabet or finite-field [7]. Several studies have extended this idea by using recursive sequences, dynamic matrices, bi-periodic Fibonacci matrix structures were studied in [8]. Fibonacci coding and Fibonacci-based cryptographic methods have been studied in different contexts

Academic Editors: Dongxi Ye and Fabio Caldarola

Received: 20 May 2026

Revised: 9 July 2026

Accepted: 10 July 2026

Published: 14 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

[9], and generalized Fibonacci matrices have been combined with affine Hill-type encryption [10–13]. Related matrix-based public-key constructions were considered in [14–16]. More specifically, bi-periodic Pell matrix structures were studied in [17], whereas a Pell-equation-based encryption and decryption algorithm and coding theory with $h(x)$ -fibonacci was proposed in [18–20].

Although these studies provide useful algebraic tools for matrix-based cryptography, many of them perform encryption as finite-field or modular matrix transformations. In such settings, if sufficiently many plaintext-ciphertext pairs are available, the recovery of a fixed key matrix may often be reduced to linear algebra over the underlying field or ring. This is a well-known limitation of classical Hill-type schemes and purely finite-field recursive matrix constructions.

The proposed construction is situated at the intersection of recursive matrix-based cryptography and elliptic-curve point-valued block transformations. It uses bi-periodic Pell matrices as block-dependent scalar key matrices

$$K_i = \tilde{P}_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q),$$

where the parity function controls the alternating structure of the bi-periodic Pell matrix sequence. The plaintext and ciphertext blocks are matrices whose entries lie in a selected elliptic curve subgroup

$$G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$$

of prime order q . Hence, the construction distinguishes between elliptic curve base field $\mathbb{F}_p$, subgroup order q , and the scalar field $\mathbb{F}_q$.

The encryption operation is not an ordinary finite-field matrix multiplication. Instead, it is defined through the elliptic-curve scalar matrix action

$$M_i = B_i \star K_i,$$

where the entries of K_i act as scalar coefficients modulo q , and the resulting point combinations are computed using the elliptic curve group law in G . The corresponding decryption operation is

$$B_i = M_i \star K_i^{-1},$$

which is well-defined because the key matrices are invertible over $\mathbb{F}_q$.

This formulation distinguishes the proposed construction from previous Hill cipher variants, Fibonacci/Pell matrix constructions, and standard ECC-based hybrid schemes. Compared with the Hill-type and Pell-matrix schemes, the construction does not operate only on scalar plaintext entries over a finite field. Compared with standard ECC-based hybrid schemes, elliptic-curve points are used directly in the block transformation rather than only in the key establishment phase. Therefore, the main contribution of this study is the formal combination of block-dependent bi-periodic Pell matrices with elliptic-curve point-valued plaintext blocks.

The construction should be interpreted as a theoretical algebraic block transformation rather than a deployable authenticated encryption scheme in its deterministic form. Once the parameters a , b , and m are fixed, the key-matrix sequence is deterministic and consists of consecutive terms of the recursive matrix sequence. Therefore, deterministic parameter reuse may reveal equality patterns or algebraic relations in the known-plaintext or chosen-plaintext settings. In particular, under a chosen point-block model, specially structured inputs may reveal elliptic-curve point images of the scalar entries of K_i ; recovering the corresponding scalar coefficients from these point images is related to the elliptic-curve discrete logarithm problem in G . However, this observation does not formally prove semantic security or IND-CPA security.

The main contributions of this study are summarized as follows:

- A bi-periodic Pell matrix structure is used to generate block-dependent scalar key matrices over $\mathbb{F}_q$.
- A elliptic-curve scalar matrix action is defined on point-valued plaintext blocks in $G^{2 \times 2}$.
- The construction explicitly separates the elliptic-curve base field $\mathbb{F}_p$, subgroup order q , and scalar field $\mathbb{F}_q$.
- The algebraic correctness of encryption and decryption is established through the invertibility of $K_i \in GL_2(\mathbb{F}_q)$ and the action identity of $\star$.
- The deterministic limitations of the construction are analyzed, including effective key-state size, chosen point-block reconstruction, and the need for additional randomization or masking.
- A small numerical example over $\mathbb{F}_{47}$ is provided only to illustrate the algebraic steps of the construction.

The remainder of this paper is organized as follows. Section 2 introduces the necessary background information and notations. Section 3 presents the algebraic properties of the bi-periodic Pell matrices used for the construction. Section 4 describes the block structure, scalar matrix action, and encryption and decryption procedures. Section 5 provides a numerical example illustrating the algebraic operation of the construction. Section 6 discusses the adversary model, security limitations, brute-force considerations, computational complexity, and qualitative comparison with related frameworks. Finally, Section 7 concludes the paper and outlines the possible directions for future work.

2. Preliminaries and Notation

In this section, we introduce the basic definitions and notations required for remainder of the paper. In particular, we recall the bi-periodic Pell sequence, its associated matrix representation in a compact form, and essential concepts from elliptic-curve cryptography.

Definition 1 (Bi-Periodic Pell Sequences). *The bi-periodic Pell sequence is defined in [17] by*

$$P_m = \begin{cases} 2aP_{m-1} + P_{m-2}, & m \text{ even} \\ 2bP_{m-1} + P_{m-2}, & m \text{ odd} \end{cases}$$

with $P_0 = 0$ and $P_1 = 1$, where a and b are fixed nonzero parameters. This sequence generalizes the classical Pell numbers by allowing the recurrence coefficients to alternate according to the parity of the index.

Definition 2 (Parity Function). *For $m \in \mathbb{N}$, the parity function $\xi(m)$ is defined in [17] by*

$$\xi(m) = \begin{cases} 0 & \text{for } m \text{ is even} \\ 1 & \text{for } m \text{ is odd} \end{cases}$$

Definition 3 (Bi-periodic Pell Matrix). *The bi-periodic Pell matrix sequence is defined in [17] by*

$$\tilde{P}_m(a, b) = \begin{bmatrix} \left(\frac{b}{a}\right)^{\xi(m)} P_{m+1} & \frac{b}{a} P_m \\ P_m & \left(\frac{b}{a}\right)^{\xi(m)} P_{m-1} \end{bmatrix}$$

where P_m is the m^{th} bi-periodic Pell number.

Definition 4 (Elliptic Curve). *Let p be a prime and $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, an elliptic-curve E over $\mathbb{F}_p$ is the set of points (x, y) with $x, y \in \mathbb{F}_p$ which satisfy the equation*

$$y^2 = x^3 + cx + d$$

where $c, d \in \mathbb{F}_p$ such that $4c^3 + 27d^2 \neq 0 \pmod{p}$ together with a single element denoted by “ $\mathcal{O}$ ” and called the “point at infinity”. The elliptic-curve over $\mathbb{F}_p$ is denoted by $E_{\mathbb{F}_p}(c, d)$. The points on $E_{\mathbb{F}_p}(c, d)$ form an additive abelian group with “ $\mathcal{O}$ ” as identity element [3].

For further background on elliptic curves and their arithmetic over finite fields, see [3,20]. In the proposed construction, the elliptic-curve E is defined over the base field $\mathbb{F}_p$. However, the plaintext and ciphertext points are taken from a cyclic subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ of prime order q . Thus, p determines the coordinate field of the elliptic-curve, whereas q determines the scalar modulus for elliptic-curve scalar multiplication on G .

For the convenience of the reader, the main symbols and parameters used throughout this study are summarized below. These tables also distinguish scalar matrices over the finite-field from elliptic-curve point-valued plaintext and ciphertext blocks. For the convenience of the reader, the main symbols and parameters used throughout this study are summarized in Table 1.

Table 1. Glossary of symbols and parameters.

Symbol	Description
p	Prime defining the base field $\mathbb{F}_p$ of the elliptic-curve.
$\mathbb{F}_p$	Coordinate field over which the elliptic-curve E is defined.
$E(\mathbb{F}_p)$	Elliptic curve defined over the finite-field $\mathbb{F}_p$.
$\mathcal{O}$	Point at infinity on the elliptic-curve $E(\mathbb{F}_p)$.
G_0	A base point generates the selected elliptic-curve subgroup.
G	The cyclic subgroup of $E(\mathbb{F}_p)$ used for plaintext and ciphertext points.
q	Prime order of subgroup G . It is the scalar modulus for elliptic-curve scalar multiplication on G .
$\mathbb{F}_q$	Scalar field associated with subgroup G . The entries of the key matrices are interpreted over $\mathbb{F}_q$.
β	Primitive element used in the Diffie-Hellman-type parameter agreement process.
r	Alice’s secret exponent.
e	Bob’s secret exponent.
E_1	Alice’s public value, $E_1 = \beta^r \in H \subseteq \mathbb{F}_q^\times$
a	Public parameter, $a = \beta^e \in H \subseteq \mathbb{F}_q^\times$
b	Shared secret parameter, $b = E_1^e = a^r \in H \subseteq \mathbb{F}_q^\times$
m	Pre-agreed initial index parameter of the bi-periodic Pell matrix sequence.
$\mathcal{M}$	Set of initial indices m
T_K	actual period of the key-matrix sequence
T_{eff}	effective period used in the security/key-state analysis
$\xi(m)$	Parity function used in the bi-periodic Pell matrix sequence.
P_m	The m -th Pell number or Pell-type sequence term, depending on context.
$\tilde{P}_m(a, b)$	Bi-periodic Pell matrix of index m .
K_i	Block-dependent key matrix, $K_i = \tilde{P}_{m+i-1}(a, b)$
$M_2(\mathbb{F}_q)$	Set of all 2×2 matrices over $\mathbb{F}_q$.
B	Plaintext matrix of size $2n \times 2n$.
M	Ciphertext matrix of size $2n \times 2n$.
B_i	The i -th plaintext block of size 2×2 .
M_i	The i -th ciphertext block of size 2×2 .
$G^{2 \times 2}$	Set of 2×2 matrices whose entries are elliptic-curve points over $\mathbb{F}_p$.
n	Number of block rows and block columns in plaintext matrix.
$N = n^2$	The total number of 2×2 plaintext blocks.
$\star$	Elliptic-curve group action combining scalar multiplication and point addition.

Σ	Plaintext alphabet used in the symbol-to-point encoding map.
$\phi: \Sigma \rightarrow G$	Fixed public injective map assigning plaintext symbols to elliptic-curve point in G
$\phi^{-1}: \phi(\Sigma) \rightarrow \Sigma$	An inverse decoding map used to recover plaintext symbols from decrypted elliptic-curve points.
s	A plaintext symbol in Σ .
$P_s = \phi(s)$	Elliptic-curve point corresponding to plaintext symbol s .
PAD	A fixed padding symbol appended when the encoded message length is insufficient to form a square matrix of even order.
P_{PAD}	Elliptic-curve point assigned to the padding symbol.

Note. The matrices K_i are scalar coefficient matrices over $\mathbb{F}_q$, whereas the plaintext and ciphertext blocks B_i and M_i are 2×2 matrices whose entries are points in the selected elliptic-curve subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$. The symbol $\star$ denotes the elliptic-curve scalar matrix action, in which the entries of K_i act as scalar multipliers modulo q on the elliptic-curve points in B_i .

3. Algebraic Properties of the Bi-Periodic Pell Matrix

In this section, we investigate the fundamental algebraic properties of the bi-periodic Pell matrix introduced in Section 2. These properties enable the application of the proposed encryption and decryption procedures.

Lemma 1 (Determinant). *Let $\tilde{P}_m(a, b) \in M_2(\mathbb{F}_q)$ be bi-periodic Pell matrix, where $a, b \in \mathbb{F}_q^\times$. Then the determinant of $\tilde{P}_m(a, b)$ is given by*

$$\det(\tilde{P}_m(a, b)) = (-1)^m \left(\frac{b}{a}\right)^{\xi(m)}.$$

Since $a, b \in \mathbb{F}_q^\times$, the quotient b/a is well-defined in $\mathbb{F}_q$. Therefore, $\det(\tilde{P}_m(a, b)) \neq 0$ in $\mathbb{F}_q$.

Proof. We prove the result by mathematical induction on m . Let $D_m = \det \tilde{P}_m$. For $m = 0$,

$$\tilde{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and therefore

$$D_0 = 1.$$

Since $\xi(0) = 0$,

$$(-1)^0 \left(\frac{b}{a}\right)^{\xi(0)} = 1$$

Hence the result holds for $m = 0$.

Assume that

$$\det \tilde{P}_m = (-1)^m \left(\frac{b}{a}\right)^{\xi(m)}$$

for some $m \geq 0$.

From [12], we have $\left(\frac{b}{a}\right)^{\xi(m)} \tilde{P}_{m+1} = \tilde{P}_m \tilde{P}_1$. Taking determinants on both sides gives

$$\det \left[\left(\frac{b}{a}\right)^{\xi(m)} \tilde{P}_{m+1} \right] = \det(\tilde{P}_m \tilde{P}_1).$$

Since $\tilde{P}_{m+1}$ is a 2×2 matrix, we obtain $\left(\frac{b}{a}\right)^{2\xi(m)} D_{m+1} = D_m D_1$.

Moreover,

$$D_1 = \det \tilde{P}_1 = -\frac{b}{a}.$$

Therefore,

$$\left(\frac{b}{a}\right)^{2\xi(m)} D_{m+1} = -\frac{b}{a} D_m,$$

and hence

$$D_{m+1} = -\left(\frac{b}{a}\right)^{1-2\xi(m)} D_m.$$

Since

$$1 - 2\xi(m) = \xi(m+1) - \xi(m),$$

we get

$$D_{m+1} = -\left(\frac{b}{a}\right)^{\xi(m+1)-\xi(m)} D_m.$$

Assume that

$$D_m = (-1)^m \left(\frac{b}{a}\right)^{\xi(m)}.$$

Then

$$D_{m+1} = -\left(\frac{b}{a}\right)^{\xi(m+1)-\xi(m)} (-1)^m \left(\frac{b}{a}\right)^{\xi(m)}.$$

Thus,

$$D_{m+1} = (-1)^{m+1} \left(\frac{b}{a}\right)^{\xi(m+1)}.$$

Hence, by induction,

$$\det \tilde{P}_m = (-1)^m \left(\frac{b}{a}\right)^{\xi(m)}$$

for all $m \geq 0$. $\square$

Lemma 2. Let q be a prime and let $a, b \in \mathbb{F}_q^\times$. For each m , let $\tilde{P}_m(a, b) \in M_2(\mathbb{F}_q)$ be the bi-periodic Pell matrix. Then $\tilde{P}_m(a, b) \in GL_2(\mathbb{F}_q)$. Equivalently, $\det(\tilde{P}_m(a, b)) \neq 0$ in $\mathbb{F}_q$.

Proof. From Lemma 1, the determinant of the bi-periodic Pell matrix is given by

$$\det(\tilde{P}_m(a, b)) = (-1)^m \left(\frac{b}{a}\right)^{\xi(m)}.$$

Since $a, b \in \mathbb{F}_q^\times$, both a and b are nonzero in $\mathbb{F}_q$. Hence a^{-1} exists in $\mathbb{F}_q$, and therefore

$$\frac{b}{a} = ba^{-1} \in \mathbb{F}_q^\times.$$

Thus,

$$\left(\frac{b}{a}\right)^{\xi(m)} \neq 0$$

in $\mathbb{F}_q$. Also, $(-1)^m \neq 0$ in $\mathbb{F}_q$. Consequently,

$$\det(\tilde{P}_m(a, b)) \neq 0$$

in $\mathbb{F}_q$. Therefore, $\tilde{P}_m(a, b)$ is an invertible 2×2 matrix over $\mathbb{F}_q$. Hence, $\tilde{P}_m(a, b) \in GL_2(\mathbb{F}_q)$. $\square$

Corollary 1. For each block-dependent key matrix $K_i = \tilde{P}_{m+i-1}(a, b)$, where $a, b \in \mathbb{F}_q^\times$, we have $K_i \in GL_2(\mathbb{F}_q)$. Therefore, the inverse matrix K_i^{-1} used in the decryption process is well-defined over the scalar field $\mathbb{F}_q$.

Proposition 1. Let $K_i = \tilde{P}_{m+i-1}(a, b)$ be the i -th block-dependent key matrix over $\mathbb{F}_q$, where $a, b \in \mathbb{F}_q^\times$. Then the consecutive key matrices satisfy

$$K_{i+1} = \left(\frac{a}{b}\right)^{\xi(m+i-1)} K_i \tilde{P}_1(a, b).$$

Proof. From the multiplicative relation of the bi-periodic Pell matrix sequence, we have

$$\tilde{P}_{t+1}(a, b) = \left(\frac{a}{b}\right)^{\xi(t)} \tilde{P}_t(a, b) \tilde{P}_1(a, b).$$

Taking $t = m + i - 1$, we obtain

$$\tilde{P}_{m+i}(a, b) = \left(\frac{a}{b}\right)^{\xi(m+i-1)} \tilde{P}_{m+i-1}(a, b) \tilde{P}_1(a, b).$$

Since

$$K_i = \tilde{P}_{m+i-1}(a, b)$$

and

$$K_{i+1} = \tilde{P}_{m+i}(a, b),$$

it follows that

$$K_{i+1} = \left(\frac{a}{b}\right)^{\xi(m+i-1)} K_i \tilde{P}_1(a, b).$$

This proves this assertion. $\square$

The above proposition shows that the block-dependent key matrices K_i are not independent. Each matrix K_{i+1} is obtained from K_i by multiplication with the fixed matrix $\tilde{P}_1(a, b)$, together with the parity-dependent scalar factor

$$\left(\frac{a}{b}\right)^{\xi(m+i-1)}.$$

Thus, the parity function controls not only the alternating structure of the bi-periodic Pell matrix sequence, but also the transition between consecutive block-dependent key matrices. Over the finite field $\mathbb{F}_q$, periodic behavior is not merely possible; it follows from the fact that the recurrence is represented by an invertible transition on a finite augmented state space.

Proposition 2. Periodicity over $\mathbb{F}_q$.

Let q be a prime and let $a, b \in \mathbb{F}_q^\times$. Consider the bi-periodic Pell recurrence over $\mathbb{F}_q$, and define the augmented state by

$$S_n = (P_n, P_{n-1}, n \bmod 2) \in \mathbb{F}_q^2 \times \{0, 1\}.$$

Then the recurrence induces a bijective transition map on this finite augmented state space. Consequently, each orbit is purely periodic. In particular, there exists a positive integer T , with $T \leq 2q^2$, such that $S_{n+T} = S_n$ for all $n \geq 0$. Hence, the associated bi-periodic Pell sequence and the block-dependent key-matrix sequence derived from it are periodic over $\mathbb{F}_q$.

Proof.

For the appropriate parity coefficient $c_\epsilon \in \{a, b\}$, the one-step transition on the pair (P_n, P_{n-1}) is represented by the matrix

$$\begin{pmatrix} P_{n+1} \\ P_n \end{pmatrix} = A_{\varepsilon_n} \begin{pmatrix} P_n \\ P_{n-1} \end{pmatrix},$$

where

$$A_{\varepsilon_n} = \begin{pmatrix} 2c_{\varepsilon_n} & 1 \\ 1 & 0 \end{pmatrix}.$$

Since

$$\det(A_{\varepsilon}) = -1 \neq 0$$

in $\mathbb{F}_q$, each parity-dependent transition matrix is invertible over $\mathbb{F}_q$. Moreover, the parity component $n \pmod{2}$ is updated deterministically and reversibly. Therefore, the augmented transition map on

$$\mathbb{F}_q^2 \times \{0,1\}$$

is a bijection on finite sets. Because every orbit of a bijection on a finite set is periodic from its initial state, the sequence is purely periodic over $\mathbb{F}_q$. This proves the assertion. $\square$

Let T_K denote the period of the resulting key matrix sequence. Then the initial index m is effective only modulo T_K . In particular, two initial indices that are congruent modulo T_K generate the same shifted sequence of block-dependent key matrices. Consequently, m should be interpreted as a pre-agreed index offset determining the starting position of the key-matrix sequence, rather than an unlimited independent source of cryptographic security.

4. Algebraic Block Encryption Construction

In this section, we introduce the proposed algebraic block encryption construction based on bi-periodic Pell matrices and elliptic-curve point-valued plaintext blocks. Let $E/\mathbb{F}_p$ be an elliptic-curve defined over the base field $\mathbb{F}_p$, and let $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ be a cyclic subgroup of prime order q . In this construction, the plaintext and ciphertext blocks are matrices whose entries are points in G , and the key matrices are scalar coefficient matrices over $\mathbb{F}_q$.

Throughout the construction, the parameters $a, b \in \mathbb{F}_q^\times$ are assumed to be nonzero. This assumption ensures that the quotient $b/a = ba^{-1}$ is well-defined in $\mathbb{F}_q$. For the i -th plaintext block $B_i \in G^{2 \times 2}$, the corresponding block-dependent key matrix is defined by

$$K_i = \tilde{P}_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q),$$

where m is a pre-agreed initial index parameter. The parameter m is fixed in advance by the communicating parties during the session setup and is not transmitted publicly during the encryption process. Thus, the sequence of key matrices starts from the agreed index m , and consecutive plaintext blocks are transformed by consecutive bi-periodic Pell matrices over the scalar field $\mathbb{F}_q$.

The encryption of the i -th plaintext block is not ordinary matrix multiplication over $\mathbb{F}_p$. Instead, it is defined by the elliptic-curve scalar matrix action

$$M_i = B_i \star K_i,$$

$$\star: G^{2 \times 2} \times M_2(\mathbb{F}_q) \rightarrow G^{2 \times 2}.$$

where $K_i \in M_2(\mathbb{F}_q)$, $B_i, M_i \in G^{2 \times 2}$, and the entries of K_i act as scalar multipliers modulo q on the elliptic-curve points in B_i . Therefore, different plaintext blocks are transformed by different scalar key matrices from the sequence

$$\tilde{P}_m(a, b), \tilde{P}_{m+1}(a, b), \tilde{P}_{m+2}(a, b), \dots$$

The parity function $\xi(m)$ has an algebraic role in the generation of bi-periodic Pell matrices. It controls the alternating occurrence of parameters a and b in the consecutive terms of the matrix sequence. In particular, the key matrices do not evolve as independent random matrices; rather, they are consecutive terms of a structured parity-dependent matrix sequence. Hence, the use of $\xi(m)$ produces an alternating algebraic structure in the sequence

$$K_i = \tilde{P}_{m+i-1}(a, b),$$

this distinguishes the proposed construction from classical Pell and Fibonacci matrix-based schemes.

Definition 5 (Elliptic-Curve Scalar Matrix Action). Let $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ be a cyclic subgroup of prime order q . For

$$B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \in G^{2 \times 2}$$

and

$$K = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix} \in GL_2(\mathbb{F}_q),$$

we define the elliptic-curve scalar matrix action by

$$B \star K = \begin{pmatrix} [k_1]b_1 \oplus [k_3]b_2 & [k_2]b_1 \oplus [k_4]b_2 \\ [k_1]b_3 \oplus [k_3]b_4 & [k_2]b_3 \oplus [k_4]b_4 \end{pmatrix}.$$

Here, $[k]P$ denotes scalar multiplication of the elliptic-curve point $P \in G$ by the scalar $k \in \mathbb{F}_q$, and $\oplus$ denotes elliptic-curve group addition in G . Since $|G| = q$, all scalar coefficients are interpreted modulo q .

Throughout the construction, this is used as a right scalar matrix action on point-valued blocks: encryption is written consistently as $M_i = B_i \star K_i$ and decryption as $B_i = M_i \star K_i^{-1}$ with $B_i, M_i \in G^{2 \times 2}$ and $K_i \in GL_2(\mathbb{F}_q)$. Thus, the order of the action, block indices, and matrix dimensions is fixed throughout the manuscript.

Lemma 3 (Compatibility of the Scalar Matrix Action).

Let $B \in G^{2 \times 2}$ and let $K, L \in GL_2(\mathbb{F}_q)$. Then

$$B \star I_2 = B$$

and

$$(B \star K) \star L = B \star (KL).$$

Proof. The first identity follows directly from the definition of $\star$, since multiplication by $1 \in \mathbb{F}_q$ fixes every point of G , and multiplication by $0 \in \mathbb{F}_q$ gives the identity point $\mathcal{O}$.

For the second identity, write

$$B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}, K = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix}, L = \begin{pmatrix} u & v \\ w & z \end{pmatrix}.$$

Using the definition of $\star$, the first entry of $(B \star K) \star L$ is

$$[u]([k_1]b_1 \oplus [k_3]b_2) \oplus [w]([k_2]b_1 \oplus [k_4]b_2).$$

By distributivity of elliptic-curve scalar multiplication over point addition, this equals

$$[uk_1]b_1 \oplus [uk_3]b_2 \oplus [wk_2]b_1 \oplus [wk_4]b_2.$$

Grouping the terms gives

$$[uk_1 + wk_2]b_1 \oplus [uk_3 + wk_4]b_2.$$

This is the first entry of $B \star (KL)$. The other entries are obtained in the same way. Hence,

$$(B \star K) \star L = B \star (KL).$$

This proves the assertion. $\square$

4.1. Message Representation and Block Structure

Let $E/\mathbb{F}_p$ be an elliptic-curve over the base field $\mathbb{F}_p$, and let $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ be a cyclic subgroup of prime order q . Plaintext symbols are encoded into points of G by a fixed public symbol-to-point mapping. The encoded message is then arranged as a $2n \times 2n$ point matrix and partitioned into 2×2 blocks $B_i \in G^{2 \times 2}$. The corresponding ciphertext blocks also belong to $G^{2 \times 2}$, whereas the block-dependent key matrices K_i are scalar matrices over $\mathbb{F}_q$. The plaintext message is first encoded by a fixed public symbol-to-point map into elliptic-curve points. If necessary, padding symbols are inserted so that the encoded message can be arranged into a square matrix of even order. The plaintext matrix is written as

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,2n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{2n,1} & b_{2n,2} & \dots & b_{2n,2n} \end{bmatrix} \text{ where } b_{j,k} \in G \ 1 \leq j, k \leq 2n$$

The matrix B is partitioned into n^2 submatrices of size 2×2 sequentially from left to right

$$\begin{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & \begin{bmatrix} b_{1,3} & b_{1,4} \\ b_{2,3} & b_{2,4} \end{bmatrix} & \dots & \begin{bmatrix} b_{1,2n-1} & b_{1,2n} \\ b_{2,2n-1} & b_{2,2n} \end{bmatrix} \\ \begin{bmatrix} b_{3,1} & b_{3,2} \\ b_{4,1} & b_{4,2} \end{bmatrix} & \begin{bmatrix} b_{3,3} & b_{3,4} \\ b_{4,3} & b_{4,4} \end{bmatrix} & \dots & \begin{bmatrix} b_{3,2n-1} & b_{3,2n} \\ b_{4,2n-1} & b_{4,2n} \end{bmatrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{bmatrix} b_{2n-1,1} & b_{2n-1,2} \\ b_{2n,1} & b_{2n,2} \end{bmatrix} & \begin{bmatrix} b_{2n-1,3} & b_{2n-1,4} \\ b_{2n,3} & b_{2n,4} \end{bmatrix} & \dots & \begin{bmatrix} b_{2n-1,2n-1} & b_{2n-1,2n} \\ b_{2n,2n-1} & b_{2n,2n} \end{bmatrix} \end{bmatrix}$$

Here, n denotes the number of block rows and block columns of the plaintext matrix. Thus, the plaintext matrix B can be written in block form as

$$\begin{bmatrix} B_1 & B_2 & \dots & B_n \\ B_{n+1} & B_{n+2} & \dots & B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{n^2-n+1} & B_{n^2-n+2} & \dots & B_{n^2} \end{bmatrix}$$

The indices r and s denote the block row and block column, respectively, where $1 \leq r, s \leq n$. For each pair (r, s) , the corresponding 2×2 plaintext block is defined by

$$B_i = \begin{bmatrix} b_{2r-1, 2s-1} & b_{2r-1, 2s} \\ b_{2r, 2s-1} & b_{2r, 2s} \end{bmatrix} \text{ where } j = 2r - 1, k = 2s - 1 \text{ and } b_{j,k} \in B_i.$$

Equivalently, this block decomposition can be expressed as B_i , $i = (r - 1)n + s$ where $1 \leq r, s \leq n$. For example, the first block is obtained by taking $r = 1$ and $s = 1$. Hence,

$$B_1 = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix}.$$

Similarly, the second block is obtained by taking $r = 1$ and $s = 2$. Since $(r - 1)n + s = (1 - 1)n + 2 = 2$, we obtain

$$B_2 = \begin{bmatrix} b_{1,3} & b_{1,4} \\ b_{2,3} & b_{2,4} \end{bmatrix}.$$

Therefore, the $2n \times 2n$ plaintext matrix is decomposed into n^2 plaintext blocks, each of size 2×2 . Each block B_i , $1 \leq i \leq n^2$, consists of four elliptic-curve points and is processed separately during the encryption procedure. The encryption of the i -th block is performed by using the corresponding key matrix K_i , generated from the bi-periodic Pell matrix sequence, as follows:

$$M_i \equiv B_i \star K_i \pmod{p}, i = 1, 2, \dots, n^2.$$

Here, M_i denotes the i -th ciphertext block corresponding to the plaintext block B_i . The corresponding block-dependent key matrix is defined by $K_i = \tilde{P}_{m+i-1}(a, b)$, where $a, b \in \mathbb{F}_q^\times$, and m is the pre-agreed initial index parameter. If

$$K_i = \begin{bmatrix} k_{11}^{(i)} & k_{12}^{(i)} \\ k_{21}^{(i)} & k_{22}^{(i)} \end{bmatrix}, k_{11}^{(i)}, k_{12}^{(i)}, k_{21}^{(i)}, k_{22}^{(i)} \in \mathbb{F}_q,$$

then the encryption of the block B_i is defined by

$$M_i = B_i \star K_i,$$

where $\star$ denotes the action of the finite-field coefficients of K_i on the elliptic-curve points of B_i . More explicitly,

$$M_i = \begin{bmatrix} [k_{11}^{(i)}]b_{2r-1,2s-1} \oplus [k_{21}^{(i)}]b_{2r-1,2s} & [k_{12}^{(i)}]b_{2r-1,2s-1} \oplus [k_{22}^{(i)}]b_{2r-1,2s} \\ [k_{11}^{(i)}]b_{2r,2s-1} \oplus [k_{21}^{(i)}]b_{2r,2s} & [k_{12}^{(i)}]b_{2r,2s-1} \oplus [k_{22}^{(i)}]b_{2r,2s} \end{bmatrix}.$$

Here, $[k]P$ denotes the scalar multiplication of the elliptic-curve point $P \in G$ by the scalar $k \in \mathbb{F}_q$, and $\oplus$ denotes the elliptic-curve group addition in G . Since G has prime order q , all scalar coefficients are interpreted modulo q . Therefore, the ciphertext block also consists of elliptic-curve points, namely $M_i \in G^{2 \times 2}$. Accordingly, the encryption of the i -th block is written as $M_i = K_i \star B_i$. This expression should not be interpreted as ordinary matrix multiplication over $\mathbb{F}_p$. Instead, $\star$ denotes the elliptic-curve scalar matrix action: the entries of $K_i \in M_2(\mathbb{F}_q)$ act as scalar coefficients on the elliptic-curve points in $B_i \in G^{2 \times 2}$, and the resulting point multiples are combined by using the elliptic-curve group law.

The bi-periodic Pell matrix $K_i = \tilde{P}_{m+i-1}(a, b)$ determines the scalar coefficients used in the transformation of the i -th plaintext block. Here, $a, b \in \mathbb{F}_q^\times$, and hence $K_i \in M_2(\mathbb{F}_q)$. On the other hand, the plaintext and ciphertext blocks are formed by elliptic-curve points in the subgroup G . Hence, the encryption process combines two algebraic structures: the scalar field $\mathbb{F}_q$, which is used in the generation of the bi-periodic Pell key matrices, and the elliptic curve subgroup $G \subseteq E(\mathbb{F}_p)$, which is used in the transformation of plaintext blocks into ciphertext blocks.

The parity function $\xi(m + i - 1)$ preserves its role in the generation of consecutive key matrices. Since the key matrix of the i -th block is given by $K_i = \tilde{P}_{m+i-1}(a, b)$, the scalar coefficients acting on the elliptic-curve points vary from block to block according to the parity-dependent bi-periodic Pell matrix sequence. Therefore, consecutive plaintext blocks are not transformed by a fixed scalar matrix. Instead, each block is transformed by a different elliptic-curve scalar matrix action induced by the corresponding matrix K_i . This makes elliptic-curve group operations a central part of the proposed encryption mechanism rather than a secondary mapping layer.

Encoding, Padding, Validation, and Decoding Rules

In the proposed construction, plaintext symbols are encoded into elliptic-curve points by a fixed public injective map

$$\phi: \Sigma \rightarrow G,$$

where Σ denotes the plaintext alphabet and $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ is the selected cyclic subgroup of prime order q . The map ϕ is not secret; it is used only to represent plaintext symbols as elliptic-curve points. For a plaintext symbol $s \in \Sigma$, the corresponding point is denoted by

$$P_s = \phi(s) \in G.$$

If the number of encoded symbols is not sufficient to form a square matrix of even order, a fixed padding symbol, denoted by PAD, is appended until the encoded sequence has length $(2n)^2$ for some positive integer n . The padding symbol is also assigned to a fixed point

$$P_{\text{PAD}} = \phi(\text{PAD}) \in G.$$

After decryption, the inverse map

$$\phi^{-1}: \phi(\Sigma) \rightarrow \Sigma$$

is applied to the recovered elliptic-curve points, and the trailing padding symbols are removed.

All scalar coefficients appearing in the matrices $K_i \in M_2(\mathbb{F}_q)$ are interpreted modulo q . Thus, for a scalar coefficient $k \in \mathbb{F}_q$ and a point $P \in G$, the scalar multiplication $[k]P$ is computed using the integer representative of k modulo q . Since G has order q , this interpretation is consistent with elliptic-curve scalar multiplication.

For validation, every received or recovered elliptic-curve point must satisfy the curve equation over $\mathbb{F}_p$ and must belong to the selected subgroup G . Equivalently, for a point P , one checks $P \in E(\mathbb{F}_p)$ and $[q]P = \mathcal{O}$. The point at infinity $\mathcal{O}$ is the identity element of the elliptic-curve group and may occur during intermediate group operations. However, it is not assigned to an ordinary plaintext symbol unless explicitly included as a padding or special symbol. Therefore, if $\mathcal{O}$ occurs in a non-padding position after decryption, or if a recovered point is not in the image of the encoding map ϕ , then decoding is declared unsuccessful.

A decoding failure occurs if at least one of the following conditions holds: the received point does not lie on $E(\mathbb{F}_p)$, the point does not belong to G , the recovered point is not contained in the image of ϕ , or the padding pattern is invalid. These rules make the encoding and decoding procedure reproducible and separate it from the cryptographic secrecy of the construction.

Theorem 1 (Correctness). *Let $B_i \in G^{2 \times 2}$ be the i -th plaintext block and let $K_i \in GL_2(\mathbb{F}_q)$ be the corresponding block-dependent key matrix. If $M_i = B_i \star K_i$, then decryption by K_i^{-1} recovers the original plaintext block:*

$$M_i \star K_i^{-1} = B_i.$$

Proof.

$$M_i \star K_i^{-1} = (B_i \star K_i) \star K_i^{-1} = B_i \star (K_i K_i^{-1}) = B_i \star I_2 = B_i.$$

□

4.2. Public Parameters Setup

The scalar parameters a and b used in the bi-periodic Pell matrix sequence are generated in the scalar field $\mathbb{F}_q$, where q is the prime order of the selected elliptic-curve subgroup $G = \langle G_0 \rangle$. Let $H = \langle \beta \rangle$ be a cyclic subgroup of $\mathbb{F}_q^\times$ generated by β . The

elements of H are nonzero elements of $\mathbb{F}_q$, and therefore they can be used as admissible scalar parameters in the Pell matrix sequence.

Alice selects a secret integer r and computes $E_1 = \beta^r \in H$. Then Alice sends the public information (β, E_1) to Bob.

Bob selects a secret integer e and computes $a = \beta^e \in H$ and $b = E_1^e = \beta^{re} \in H$. Since $H \subseteq \mathbb{F}_q^\times$, the obtained values satisfy $a, b \in \mathbb{F}_q^\times$. These parameters are then incorporated into the bi-periodic Pell matrix sequence. The value a is transmitted publicly as part of the ciphertext data, whereas b is the shared scalar parameter used to reconstruct the block-dependent key matrices. Alice can recover the same value b from Bob's public value a by computing

$$b = a^r = \beta^{er}.$$

Thus, both parties obtain the same scalar parameter pair (a, b) over $\mathbb{F}_q$, while the block-dependent key matrices are generated as $K_i = \tilde{P}_{m+i-1}(a, b) \in M_2(\mathbb{F}_q)$. This parameter setup is used to derive scalar coefficients for the Pell matrix sequence. It is not an elliptic-curve key exchange by itself; the elliptic-curve is used in the subsequent scalar matrix action on point-valued plaintext blocks.

4.3. Key Generation Using Bi-Periodic Pell Matrices

Using the scalar parameters $a, b \in \mathbb{F}_q^\times$ generated in Section 4.2, the block-dependent key matrices are constructed from the bi-periodic Pell matrix sequence over the scalar field $\mathbb{F}_q$. For the i -th plaintext block B_i , where $1 \leq i \leq n^2$, the corresponding key matrix is defined by

$$K_i = \tilde{P}_{m+i-1}(a, b) \in M_2(\mathbb{F}_q),$$

where m is the pre-agreed initial index parameter of the matrix sequence. By Lemma 2, since $a, b \in \mathbb{F}_q^\times$, each key matrix satisfies $K_i \in GL_2(\mathbb{F}_q)$. Therefore, the inverse matrix K_i^{-1} used in the decryption process is well-defined over $\mathbb{F}_q$.

The parameter m determines the starting position of the bi-periodic Pell matrix sequence. It is fixed in advance by the communicating parties during the session setup and is not transmitted publicly during encryption. Hence, the sequence of key matrices used in the encryption process starts from the agreed index m , namely

$$\tilde{P}_m(a, b), \tilde{P}_{m+1}(a, b), \tilde{P}_{m+2}(a, b), \dots$$

Equivalently, the first plaintext blocks are transformed by the consecutive key matrices

$$K_1 = \tilde{P}_m(a, b), K_2 = \tilde{P}_{m+1}(a, b), K_3 = \tilde{P}_{m+2}(a, b), \dots$$

These block-dependent key matrices should not be interpreted as independent random matrices. They are consecutive terms of the same structured bi-periodic Pell matrix sequence. In particular, as shown in Proposition 1, consecutive key matrices satisfy

$$K_{i+1} = \begin{pmatrix} a \\ b \end{pmatrix}^{\xi(m+i-1)} K_i \tilde{P}_1(a, b).$$

This relation shows that the transition from K_i to K_{i+1} is controlled by the parity function and the fixed matrix $\tilde{P}_1(a, b)$. Thus, the parity-dependent structure of the bi-periodic Pell sequence determines how the scalar coefficients vary from block to block.

4.4. Encryption-Decryption Process

The parameter/key generation stage derives the scalar parameters a and b over the multiplicative group $\mathbb{F}_q^\times$. This procedure is summarized in Algorithm 1.

Algorithm 1 Parameter/Key Generation

Input: prime q , primitive element $\beta \in \mathbb{F}_q^\times$, Alice's public value E_1 , pre-agreed index parameter m

Output: scalar parameters a and b

1. Choose a secret integer e , where $1 < e < q - 1$.
 2. Compute $a \equiv \beta^e \pmod{q}$.
 3. Compute $b \equiv E_1^e \equiv \beta^{re} \pmod{q}$.
 4. Output the parameters (a, b) .
-

In this stage, the parameters a and b are generated over the multiplicative group $\mathbb{F}_q^\times$, where q is the prime order of the selected elliptic-curve subgroup $G = \langle G_0 \rangle$. They are not generated by elliptic-curve point operations. These parameters are subsequently used as nonzero scalar inputs in the bi-periodic Pell matrix sequence. Since $a, b \in \mathbb{F}_q^\times$, the block-dependent key matrices are generated as $K_i = \tilde{P}_{m+i-1}(a, b) \in M_2(\mathbb{F}_q)$. By Lemma 2, each K_i is invertible over $\mathbb{F}_q$. Hence, $K_i \in GL_2(\mathbb{F}_q)$, and the inverse matrix K_i^{-1} used in decryption is well-defined over the scalar field $\mathbb{F}_q$.

After the scalar parameters have been generated, the plaintext is encoded into elliptic-curve points, partitioned into 2×2 blocks, and transformed by block-dependent key matrices. The complete encryption procedure is given in Algorithm 2.

Algorithm 2 Encryption

Input: plaintext message P , elliptic-curve $E/\mathbb{F}_p$, selected subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ of prime order q , public symbol-to-point encoding map ϕ , block size 2×2 , pre-agreed index parameter m , scalar parameters $a, b \in \mathbb{F}_q^\times$

Output: ciphertext pair (a, M)

1. Encode the plaintext message P into elliptic-curve points in G by using the public encoding map

$$\phi: \Sigma \rightarrow G.$$

2. If necessary, append padding symbols and arrange the encoded points as a matrix

$$B \in G^{2n \times 2n}.$$

3. Partition B into n^2 non-overlapping plaintext blocks $B_1, B_2, \dots, B_{n^2}$, where $B_i \in G^{2 \times 2}$.

4. For each block index $i = 1, 2, \dots, n^2$, compute the block-dependent key matrix

$$K_i = \tilde{P}_{m+i-1}(a, b) \in M_2(\mathbb{F}_q).$$

By Lemma 2, since $a, b \in \mathbb{F}_q^\times$, we have $K_i \in GL_2(\mathbb{F}_q)$.

5. For each plaintext block B_i , compute the ciphertext block by the elliptic-curve scalar matrix action

$$M_i = B_i \star K_i.$$

Here, $\star: G^{2 \times 2} \times M_2(\mathbb{F}_q) \rightarrow G^{2 \times 2}$ denotes the elliptic-curve scalar matrix action.

6. Assemble the encrypted blocks M_i in their original block positions to obtain
-

$$M \in G^{2n \times 2n}$$

7. Return the ciphertext pair (a, M) .

The block-level encryption flow corresponding to Algorithm 2 is illustrated in Figure 1.

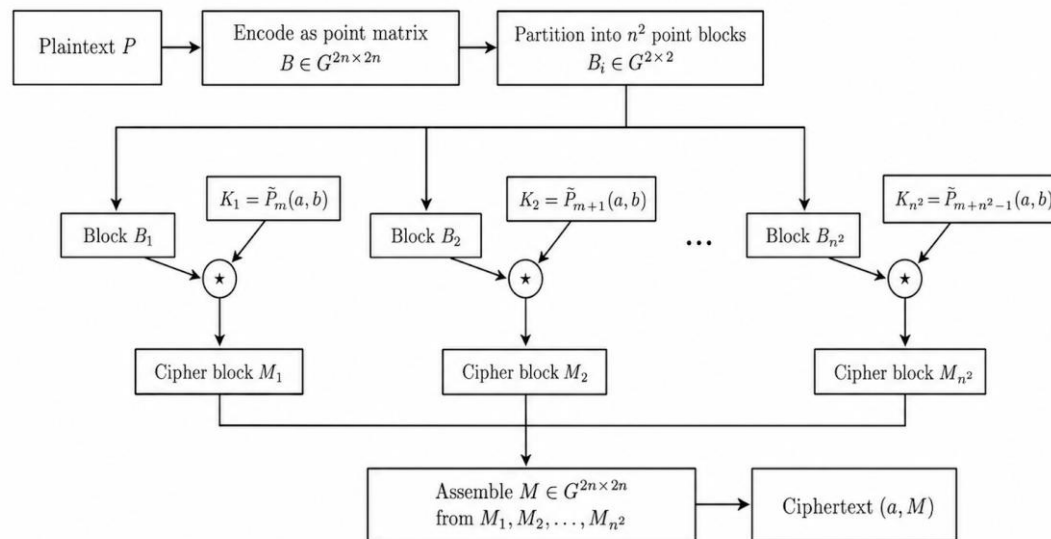


Figure 1. Encryption flow. The plaintext is encoded as $B \in G^{2n \times 2n}$ and partitioned into blocks $B_i \in G^{2 \times 2}$. Each block is transformed as $M_i = B_i \star K_i$, where $K_i = \tilde{P}_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q)$. The symbol $\star$ denotes the elliptic-curve scalar matrix action; scalar coefficients are interpreted modulo q , and point additions are performed in $G \subseteq E(\mathbb{F}_p)$. The parameter m is pre-agreed and is not transmitted.

The corresponding decryption procedure reconstructs the same block-dependent key matrices and applies the inverse scalar matrix action to recover the plaintext blocks. This procedure is presented in Algorithm 3.

Algorithm 3 Decryption

Input: ciphertext pair (a, M) , elliptic-curve $E/\mathbb{F}_p$, selected subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ of prime order q , Alice's secret integer r , pre-agreed index parameter m , public decoding map ϕ^{-1}

Output: plaintext message P

1. Compute

$$b \equiv a^r \pmod{q}.$$

2. Partition the ciphertext matrix M into n^2 non-overlapping blocks

$$M_1, M_2, \dots, M_{n^2},$$

where $M_i \in G^{2 \times 2}$.

3. For $i = 1, 2, \dots, n^2$, reconstruct the block-dependent key matrix

$$K_i \leftarrow \tilde{P}_{m+i-1}(a, b) \in M_2(\mathbb{F}_q).$$

4. Compute the inverse key matrix

$$K_i^{-1} \leftarrow \tilde{P}_{m+i-1}(a, b)^{-1},$$

Since $a, b \in \mathbb{F}_q^\times$, by Lemma 2 we have $K_i \in GL_2(\mathbb{F}_q)$, and hence K_i^{-1} is well-defined over $\mathbb{F}_q$.

5. Recover each plaintext block by the elliptic-curve scalar matrix action

$$B_i \leftarrow M_i \star K_i^{-1}.$$

6. Assemble the recovered plaintext blocks B_i to obtain

$$B \in G^{2n \times 2n}.$$

7. Decode the elliptic-curve points in B by using the inverse mapping ϕ^{-1} , remove the trailing padding symbols if present, and recover the original plaintext message P .

The decryption flow corresponding to Algorithm 3 is illustrated in Figure 2.

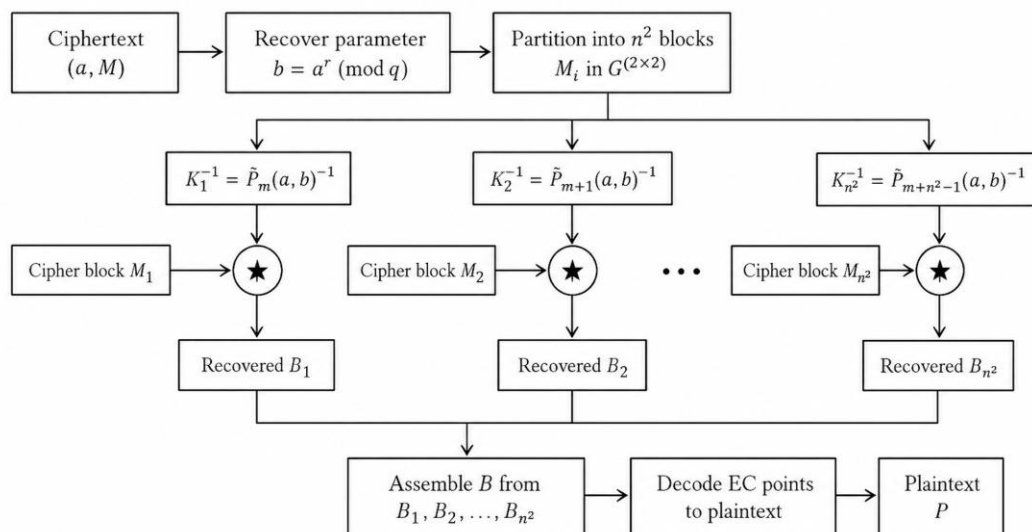


Figure 2. Decryption flow. The ciphertext pair (a, M) is processed by first recovering $b = a^r \pmod{q}$. The ciphertext matrix is partitioned into blocks $M_i \in G^{2 \times 2}$, and each plaintext block is recovered through the elliptic-curve scalar matrix action $B_i = M_i \star K_i^{-1}$, where $K_i = \tilde{P}_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q)$. The recovered blocks are assembled into B and decoded to obtain the plaintext P . The parameter m is pre-agreed and is not transmitted.

5. Numerical Example

In this section, the proposed construction is illustrated by a small numerical example. The purpose of this example is only to demonstrate the algebraic steps of the proposed procedure. The parameters used in this section are deliberately small and should not be interpreted as cryptographically secure parameters for real-world applications. Consider the plaintext

“A BI-PERIODIC PELL MATRIX BASED ENCRYPTION METHOD OVER ECC”.

Let $E/\mathbb{F}_{47} : y^2 = x^3 + 3x + 41 \pmod{47}$ be an elliptic-curve over $\mathbb{F}_{47}$. By exhaustive enumeration over $\mathbb{F}_{47}$, the curve has 46 affine points. Together with the point at infinity $\mathcal{O}$, this gives

$$\#E(\mathbb{F}_{47}) = 46 + 1 = 47.$$

Since 47 is prime, $E(\mathbb{F}_{47})$ is a group of prime order. Hence, every non-identity point is a generator of the group. In particular,

$$G_0 = (2, 14) \neq \mathcal{O}$$

generates the whole group. Thus, in this illustrative example, we take

$$G = \langle G_0 \rangle = E(\mathbb{F}_{47})$$

and

$$q = |G| = 47.$$

Therefore, in this example, the elliptic-curve base-field prime and the subgroup order coincide, namely $p = q = 47$. This equality is used only for computational simplicity. In general, p and q may be distinct, and scalar coefficients acting on elliptic-curve points must be interpreted modulo the subgroup order q , not necessarily modulo the base-field prime p .

The complete list of elliptic-curve points used in this example is given in Appendix A.1.

The point list was obtained by checking all pairs $(x, y) \in \mathbb{F}_{47} \times \mathbb{F}_{47}$ satisfying the curve equation. This example is used only for illustration. In practical cryptographic applications, substantially larger finite-fields, secure elliptic-curve parameters, and suitable subgroup orders must be used.

The characters of the plaintext are mapped to elliptic-curve points according to the symbol-to-point mapping given in Appendix A. The plaintext contains 58 characters. Since the proposed block construction uses a square matrix of even order and 2×2 plaintext blocks, the message is padded to length 64. Thus, the plaintext is arranged as an 8×8 matrix. Therefore,

$$2n = 8, n = 4, n^2 = 16.$$

Hence, the plaintext matrix is partitioned into exactly 16 blocks of size 2×2 . For display purposes, the plaintext characters are arranged as follows:

$$\begin{bmatrix} A & - & B & I & - & P & E & R \\ I & O & D & I & C & - & P & E \\ L & L & - & M & A & T & R & I \\ X & - & B & A & S & E & D & - \\ E & N & C & R & Y & P & T & I \\ O & N & - & M & E & T & H & O \\ D & - & O & V & E & R & - & E \\ C & C & - & - & - & - & - & - \end{bmatrix}$$

Here, “-” denotes the blank-space symbol and padding symbol contained in the plaintext. After decryption, the padding symbols are removed to recover the original plaintext. Using the elliptic-curve point mapping defined in Appendix A, the above character matrix is converted into an 8×8 plaintext matrix $B = (b_{j,k})_{8 \times 8}$, $b_{j,k} \in G$. For instance, the plaintext matrix begins as

$$B = \begin{bmatrix} (2,14) & (2,33) & \cdots & (30,23) \\ \vdots & \vdots & \ddots & \vdots \\ (21,24) & (21,24) & \cdots & (2,33) \end{bmatrix}$$

The plaintext matrix B is then partitioned into 2×2 blocks. Since B is an 8×8 matrix, there are 4 block rows and 4 block columns. Therefore, the number of plaintext blocks is $4 \cdot 4 = 16$. The blocks are indexed as $B_1, B_2, \dots, B_{16}$. For $1 \leq r, s \leq 4$, the block index is given by $i = (r - 1)4 + s$. Thus, the (r, s) -th block is

$$B_i = \begin{bmatrix} b_{2r-1,2s-1} & b_{2r-1,2s} \\ b_{2r,2s-1} & b_{2r,2s} \end{bmatrix} \in G^{2 \times 2}.$$

For example, the first block is obtained by taking $r = 1$ and $s = 1$. Hence,

$$B_1 = \begin{bmatrix} A & - \\ I & 0 \end{bmatrix} = \begin{bmatrix} (2,14) & (2,33) \\ (27,26) & (43,24) \end{bmatrix}.$$

Similarly, the second block is obtained by taking $r = 1$ and $s = 2$. Since

$$i = (1 - 1)4 + 2 = 2,$$

we obtain

$$B_2 = \begin{bmatrix} B & I \\ D & I \end{bmatrix} = \begin{bmatrix} (28,9) & (27,26) \\ (33,34) & (27,26) \end{bmatrix}.$$

The remaining blocks $B_3, \dots, B_{16}$ are obtained in the same way by taking adjacent 2×2 submatrices of the 8×8 plaintext matrix. Therefore, the blocks should not be interpreted as four consecutive characters in the one-dimensional plaintext string. Instead, they correspond to adjacent 2×2 positions after the plaintext has been arranged into the square matrix B .

5.1. Construction of Public Parameters

The scalar parameters a and b are generated over the multiplicative group $\mathbb{F}_{47}^\times$. They are not generated by elliptic-curve point operations. In this example, Alice chooses the primitive element $\beta = 5$ of $\mathbb{F}_{47}^\times$ and her secret integer $r = 21$. The initial index parameter is pre-agreed as $m = 29$. Alice computes

$$E_1 \equiv \beta^r \equiv 5^{21} \equiv 15 \pmod{47}.$$

Then Alice publishes $(\beta, E_1) = (5, 15)$, while $r = 21$ is kept secret. The parameter $m = 29$ is fixed in advance by Alice and Bob during the session setup and is not transmitted as part of the ciphertext.

5.2. Key Generation and Encryption

Bob uses Alice's public information (β, E_1) to generate the parameters a and b . Bob chooses a secret integer

$$e = 5, 1 < e < 46.$$

Then he computes

$$a \equiv \beta^e \equiv 5^5 \equiv 23 \pmod{47},$$

and

$$b \equiv E_1^e \equiv 15^5 \equiv 43 \pmod{47}.$$

Thus, in this numerical example $(a, b) = (23, 43)$. Using the pre-agreed value $m = 29$, the block-dependent key matrices are constructed as

$$\tilde{P}_{m+i-1}(a,b)(\text{mod}47) \equiv \begin{bmatrix} \left(\frac{b}{a}\right)^{\xi(m+i-1)} P_{m+i} & \frac{b}{a} P_{m+i-1} \\ P_{m+i-1} & \left(\frac{b}{a}\right)^{\xi(m+i-1)} P_{m+i-2} \end{bmatrix} (\text{mod}47), \text{ for } 1 \leq i \leq 16,$$

$$\tilde{P}_{m+i-1}(a,b) \in GL_2(\mathbb{F}_{47}) \text{ and}$$

$$K_i = \tilde{P}_{m+i-1}(23,43), 1 \leq i \leq 16.$$

Equivalently,

$$K_1 = \tilde{P}_{29}(23,43), K_2 = \tilde{P}_{30}(23,43), K_3 = \tilde{P}_{31}(23,43), \dots$$

The first three key matrices are

$$K_1 = \tilde{P}_{29}(23,43) = \begin{bmatrix} 18 & 46 \\ 41 & 20 \end{bmatrix} (\text{mod}47),$$

$$K_2 = \tilde{P}_{30}(23,43) = \begin{bmatrix} 22 & 18 \\ 14 & 41 \end{bmatrix} (\text{mod}47),$$

and

$$K_3 = \tilde{P}_{31}(23,43) = \begin{bmatrix} 24 & 35 \\ 22 & 18 \end{bmatrix} (\text{mod}47).$$

The remaining key matrices $K_4, \dots, K_{16}$ are obtained analogously from the consecutive terms of the bi-periodic Pell matrix sequence. Since the entries of B_i are elliptic-curve points, the encryption operation is not an ordinary matrix multiplication over $\mathbb{F}_{47}$. Instead, each scalar entry of K_i acts on the elliptic-curve points in B_i . Therefore, the encryption of the i -th block is written as

$$M_i = B_i \star K_i,$$

where $\star$ denotes the elliptic-curve scalar matrix action. More explicitly, if

$$B_i = \begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix} \in G^{2 \times 2}, K_i = \begin{bmatrix} k_{11}^{(i)} & k_{12}^{(i)} \\ k_{21}^{(i)} & k_{22}^{(i)} \end{bmatrix} \in GL_2(\mathbb{F}_{47}).$$

then

$$B_i \star K_i = \begin{bmatrix} [k_{11}^{(i)}]P_1 \oplus [k_{21}^{(i)}]P_2 & [k_{12}^{(i)}]P_1 \oplus [k_{22}^{(i)}]P_2 \\ [k_{11}^{(i)}]P_3 \oplus [k_{21}^{(i)}]P_4 & [k_{12}^{(i)}]P_3 \oplus [k_{22}^{(i)}]P_4 \end{bmatrix}.$$

Here, $[k]P$ denotes scalar multiplication of the elliptic-curve point $P \in G$ by the scalar $k \in \mathbb{F}_{47}$, and $\oplus$ denotes elliptic-curve group addition on $G \in E(\mathbb{F}_{47})$. For the first plaintext block, we have

$$B_1 = \begin{bmatrix} (2,14) & (2,33) \\ (27,26) & (43,24) \end{bmatrix}, K_1 = \begin{bmatrix} 18 & 46 \\ 41 & 20 \end{bmatrix}.$$

Thus, $M_1 = B_1 \star K_1$. This gives

$$M_1 = \begin{bmatrix} [18](2,14) \oplus [41](2,33) & [46](2,14) \oplus [20](2,33) \\ [18](27,26) \oplus [41](43,24) & [46](27,26) \oplus [20](43,24) \end{bmatrix}.$$

After performing scalar multiplication and point addition on $E/\mathbb{F}_{47}$, we obtain

$$[41](2,33) = (11,29),$$

$$[46](2,14) = (2,33),$$

$$[20](2,33) = (38,32),$$

$$[18](27,26) = (14,17),$$

$$[41](43,24) = (33,34),$$

$$[46](27,26) = (27,21),$$

and

$$[20](43,24) = (30,23).$$

Therefore, the corresponding point additions are

$$(30,23) \oplus (11,29) = (13,31),$$

$$(2,33) \oplus (38,32) = (14,30),$$

$$(14,17) \oplus (33,34) = (34,22),$$

and

$$(27,21) \oplus (30,23) = (27,26).$$

Hence,

$$M_1 = \begin{bmatrix} (13,31) & (14,30) \\ (34,22) & (27,26) \end{bmatrix}.$$

Similarly, for the second plaintext block,

$$B_2 = \begin{bmatrix} (28,9) & (27,26) \\ (33,34) & (27,26) \end{bmatrix}, K_2 = \begin{bmatrix} 22 & 18 \\ 14 & 41 \end{bmatrix}.$$

Hence,

$$M_2 = B_2 \star K_2.$$

This yields

$$M_2 = \begin{bmatrix} (30,24) & (30,24) \\ (14,30) & (30,23) \end{bmatrix}.$$

The remaining ciphertext blocks $M_3, \dots, M_{16}$ are obtained analogously by using the corresponding key matrices

$$K_i = \tilde{P}_{m+i-1}(23,43), 1 \leq i \leq 16.$$

After all sixteen ciphertext blocks are computed, they are assembled in their original block positions to form the ciphertext matrix $M \in G^{8 \times 8}$. Finally, Bob transmits the ciphertext pair (a, M) to Alice. The parameter $m = 29$ is not transmitted, since it is pre-agreed during the session setup. The full numerical computations of all ciphertext blocks are provided in Appendix A.2

5.3. Decryption

Upon receiving the ciphertext pair (a, M) , Alice first reconstructs the scalar parameter b by using her secret integer $r = 21$. Since $a = 23$, Alice computes

$$b \equiv a^r \equiv 23^{21} \equiv 43 \pmod{47}.$$

Thus, Alice obtains the same scalar parameter pair $(a, b) = (23, 43)$. The ciphertext matrix M is then partitioned into sixteen 2×2 ciphertext blocks

$$M_1, M_2, \dots, M_{16}, M_i \in G^{2 \times 2}.$$

For each block index i , Alice reconstructs the corresponding block-dependent key matrix $K_i = \tilde{P}_{m+i-1}(23,43)$. Since $a, b \in \mathbb{F}_{47}^\times$, Lemma 2 gives $K_i \in GL_2(\mathbb{F}_{47})$.

Hence, the inverse matrix K_i^{-1} is well-defined over $\mathbb{F}_{47}$. The plaintext block is recovered by applying the inverse matrix through the same elliptic-curve scalar matrix action:

$$B_i = M_i \star K_i^{-1}.$$

For example, for $i = 1$, we have

$$M_1 = \begin{bmatrix} (13,31) & (14,30) \\ (34,22) & (27,26) \end{bmatrix}.$$

The corresponding inverse key matrix is

$$K_1^{-1} = \begin{bmatrix} 29 & 32 \\ 4 & 12 \end{bmatrix} \pmod{47}.$$

Therefore,

$$B_1 = M_1 \star K_1^{-1}.$$

More explicitly,

$$B_1 = \begin{bmatrix} [29](13,31) \oplus [4](14,30) & [32](13,31) \oplus [12](14,30) \\ [29](34,22) \oplus [4](27,26) & [32](34,22) \oplus [12](27,26) \end{bmatrix}.$$

After performing the scalar multiplications and point additions on $G = E(\mathbb{F}_{47})$, we obtain

$$B_1 = \begin{bmatrix} (2,14) & (2,33) \\ (27,26) & (43,24) \end{bmatrix}.$$

Thus, the first plaintext block is recovered exactly. After all blocks $B_1, B_2, \dots, B_{16}$ are recovered, they are assembled in their original block positions to reconstruct the plaintext matrix

$$B \in G^{8 \times 8}.$$

Finally, the elliptic-curve point representation is inverted according to the symbol-to-point mapping table in Appendix A, and the trailing padding entries are removed. This gives the original plaintext message

“A BI-PERIODIC PELL MATRIX BASED ENCRYPTION METHOD OVER ECC”.

The full numerical computations of all sixteen ciphertext blocks and their corresponding decryption steps are provided in Appendix A.2.

This numerical example is provided strictly as an illustrative algebraic computation. It should not be interpreted as evidence of practical cryptographic security, IND-CPA security, semantic security, authenticated-encryption security, or performance efficiency.

6. Security Analysis

This section discusses the algebraic and cryptographic properties of the proposed block encryption construction under explicitly stated assumptions. Standard background on classical cryptographic principles, modern security notions, and applied cryptographic constructions can be found in [21–26]. The construction combines three algebraic components: a Diffie-Hellman-type scalar parameter agreement over the multiplicative group $\mathbb{F}_q^\times$, block-dependent bi-periodic Pell key matrices over $\mathbb{F}_q$, and elliptic-curve group operations used to transform point-valued plaintext blocks into ciphertext blocks.

The construction uses two different algebraic domains. The elliptic-curve is defined over the base field $\mathbb{F}_p$, and plaintext and ciphertext entries are represented as points in the selected subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ of prime order q . On the other hand, the scalar parameters a and b , as well as the entries of the block-dependent key matrices $K_i = \tilde{P}_{m+i-1}(a, b)$, belong to the scalar field $\mathbb{F}_q$. Therefore, scalar multiplication on elliptic-curve points is interpreted modulo q , whereas point addition is performed in the subgroup G . The parameters a and b used in the bi-periodic Pell matrices are obtained through a Diffie-Hellman-type parameter agreement process over $\mathbb{F}_q^\times$. The elliptic-curve structure is then used to represent plaintext and ciphertext entries as points in G , and the block transformation is performed through the elliptic-curve scalar matrix action. Thus, the i -th encryption step has the form $M_i = B_i \star K_i$, where $B_i, M_i \in G^{2 \times 2}$ and $K_i \in GL_2(\mathbb{F}_q)$.

The security discussion is organized according to these roles. First, the secrecy of the shared scalar parameter b is related to the difficulty of recovering a Diffie-Hellman-type shared value from public information in $\mathbb{F}_q^\times$. Second, the block-dependent matrices $K_i = \tilde{P}_{m+i-1}(a, b)$ provide different scalar matrix actions for different plaintext blocks. However, these matrices are consecutive terms of the same bi-periodic Pell matrix sequence and are therefore structurally dependent rather than independent random matrices. Third, since the entries of plaintext and ciphertext blocks are elliptic-curve points, scalar recovery from point relations is connected to discrete-logarithm-type problems in the selected subgroup G , provided that the elliptic-curve parameters and subgroup order are chosen appropriately.

The numerical example over $\mathbb{F}_{47}$ given in Section 5 is used only for illustrating the algebraic steps of the construction and does not provide cryptographic security. In practical applications, the prime p , the subgroup order q , the elliptic-curve, and all scalar parameter sizes must be selected according to standard cryptographic security requirements.

The aim of this section is not to provide a formal proof of indistinguishability under chosen-plaintext attacks. In its deterministic form, the proposed construction is not claimed to achieve semantic security or IND-CPA security as a standalone encryption scheme. Once the parameters a , b , and m are fixed, the sequence of key matrices is deterministic. Therefore, additional randomization, nonce-dependent masking, or integration into a standard authenticated-encryption framework would be required for deployment in a modern cryptographic setting. Accordingly, the following analysis focuses on the algebraic validity of the key matrices, the scalar parameter space, the structural dependence and finite periodicity of the key sequence, and the limitations of the deterministic construction under known-plaintext and chosen-plaintext attack models [21–23].

6.1. Adversary Model

We consider an adversary who knows the complete description of the proposed construction. In particular, the adversary is assumed to know the elliptic-curve $E/\mathbb{F}_p$, the selected subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ of prime order q , the public symbol-to-point encoding map, the bi-periodic Pell matrix recurrence, the definition of the elliptic-curve scalar matrix action $\star$, and the encryption and decryption algorithms. Thus, the analysis follows the standard principle that the security of a cryptographic construction should not rely on the secrecy of the algorithm.

For a given encryption session, the adversary may observe the public scalar parameter a and the ciphertext matrix M . The adversary may also know the public information used in the Diffie-Hellman-type scalar parameter agreement, such as β and $E_1 = \beta^r$, when such a parameter setup is used. However, the adversary is not assumed to know Alice's secret exponent r , Bob's secret exponent e , or the resulting shared scalar parameter $b = \beta^{er}$.

The pre-agreed initial index m is also not transmitted as part of the ciphertext. However, m should not be regarded as an unlimited independent source of security. By Proposition 2, the bi-periodic Pell recurrence over $\mathbb{F}_q$ induces an invertible transition on a finite augmented state space. Hence the corresponding key-matrix sequence has a finite period T_k . Therefore, the contribution of m is bounded by T_k and m should be interpreted modulo the period of the sequence.

The main computational objectives of the adversary may be described as follows:

recover the shared scalar parameter b ,

recover the block-dependent matrices $K_i = \tilde{P}_{m+i-1}(a, b)$,

or

recover the plaintext point blocks B_i from $M_i = B_i \star K_i$.

A passive adversary is assumed to have access to ciphertexts and public session data. In a known-plaintext setting, the adversary may additionally know some plaintext blocks B_i together with their corresponding ciphertext blocks M_i . In a chosen-plaintext setting, the adversary may obtain ciphertexts for plaintexts of its choice under the same session parameters. These stronger models are considered in order to clarify the limitations of the deterministic version of the construction.

Since the construction is deterministic once a , b , and m are fixed, reusing the same session parameters may reveal structural information. In particular, the same plaintext block at the same block position is transformed by the same scalar matrix action. Consequently, equality patterns or algebraic relations between plaintext and ciphertext blocks may be exposed under known-plaintext or chosen-plaintext settings.

For this reason, the proposed deterministic construction is not claimed to provide semantic security or IND-CPA security as a standalone encryption scheme. The adversary model considered here is used to analyze the algebraic structure and the limitations of the construction, rather than to establish a complete modern security proof.

The adversary model does not include implementation-level attacks such as timing attacks, side-channel leakage, invalid-curve attacks, fault attacks, or attacks arising from improper point validation. It is assumed that all received elliptic-curve points are validated to lie in G , that scalar coefficients are reduced modulo q , and that the encoding and decoding procedures are implemented consistently.

Under this model, the following subsections analyze the role of the scalar parameter b , the invertibility and structural dependence of the key matrices K_i , and the limitations of the deterministic construction under known-plaintext and chosen-plaintext attacks [22,23].

6.2. Scalar Parameter Agreement and Key-State Recovery

The scalar parameters used in the bi-periodic Pell key matrices are obtained through a Diffie-Hellman-type parameter agreement over a multiplicative subgroup of $\mathbb{F}_q^\times$. Let $H = \langle \beta \rangle \subseteq \mathbb{F}_q^\times$ be the subgroup generated by β , and let $N = |H|$. Alice chooses a secret exponent r and publishes $E_1 = \beta^r$. Bob chooses a secret exponent e and computes $a = \beta^e$, together with the shared scalar parameter $b = E_1^e = \beta^{re}$. Alice can recover the same value by computing $b = a^r = \beta^{er}$. Thus, both parties obtain the same scalar pair $(a, b) \in \mathbb{F}_q^\times \times \mathbb{F}_q^\times$. The public information available to an adversary may include $\beta, E_1 = \beta^r, a = \beta^e$, whereas the shared scalar parameter $b = \beta^{re}$ is not transmitted.

The following statement formalizes the relation between recovering b and the computational Diffie-Hellman-type problem in H .

Theorem 2. Let $H = \langle \beta \rangle \subseteq \mathbb{F}_q^\times$ be a cyclic subgroup of order N . Let r, e be chosen uniformly from the exponent space of H , and let

$$E_1 = \beta^r, a = \beta^e, b = \beta^{re}.$$

Suppose that an adversary $\mathcal{A}$, given the public tuple $(\beta, E_1, a) = (\beta, \beta^r, \beta^e)$, outputs the shared scalar parameter $b = \beta^{re}$ with probability ε . Then $\mathcal{A}$ gives rise to an algorithm that solves the computational Diffie-Hellman problem in H with the same success probability ε .

Proof. An instance of the computational Diffie-Hellman problem in H is given by $(\beta, \beta^r, \beta^e)$, and the goal is to compute β^{re} . This is exactly the value $b = E_1^e = a^r$ used as the shared scalar parameter in the proposed construction. Therefore, any adversary that recovers b from the public tuple (β, E_1, a) also solves the computational Diffie-Hellman

problem in H . Hence, the success probability of recovering b is bounded by the adversary's success probability against the computational Diffie-Hellman problem in H . $\square$

Equivalently, if no algorithm significantly better than solving the computational Diffie-Hellman problem in H is available, then recovering b from the public data is computationally infeasible for appropriately chosen parameters. In the case of blind guessing, the probability of correctly guessing b is

$$\Pr[\hat{b} = b] = \frac{1}{N}.$$

However, the secrecy of b should be interpreted only as one component of the construction. It does not, by itself, imply semantic security or IND-CPA security of the whole deterministic block transformation. The block-dependent key matrices are determined by $K_i = \tilde{P}_{m+i-1}(a, b)$, $1 \leq i \leq n^2$. Thus, once the values a , b , and m are known, all key matrices K_i can be reconstructed. Since a is transmitted as public data, the hidden part of the key state consists of the shared scalar parameter b and the pre-agreed initial index m .

By Proposition 2, let T_k denote the period of the bi-periodic Pell key-matrix sequence over F_q . Since this period arises from a finite invertible augmented state space, the contribution of m is bounded by T_k . Hence, m should be interpreted modulo T_k . For a fixed public value a , define the key-state space by

$$\mathcal{S} = H \times \mathbb{Z}/T_{\text{eff}}\mathbb{Z}.$$

Each hidden state $(b, m) \in \mathcal{S}$ determines the block-dependent key matrix $K_i = \tilde{P}_{m+i-1}(a, b)$. For a fixed block index i , define the map

$$\Phi_i: \mathcal{S} \rightarrow GL_2(\mathbb{F}_q)$$

by

$$\Phi_i(b, m) = \tilde{P}_{m+i-1}(a, b).$$

The probability of recovering K_i by blindly guessing from the structured key-state space depends on the number of collisions of the map Φ_i . Let

$$\mu_i = \max_{K \in GL_2(\mathbb{F}_q)} |\{(b, m) \in \mathcal{S} : \Phi_i(b, m) = K\}|.$$

Then the optimal blind-guessing probability for the i -th key matrix is bounded by

$$\Pr[\hat{K}_i = K_i] \leq \frac{\mu_i}{|H| T_{\text{eff}}}.$$

In particular, if Φ_i is injective on the considered key-state space, then

$$\Pr[\hat{K}_i = K_i] \leq \frac{1}{|H| T_{\text{eff}}}.$$

More generally, for the full key-matrix sequence

$$(K_1, K_2, \dots, K_{n^2}),$$

one may define

$$\Phi(b, m) = (\tilde{P}_m(a, b), \tilde{P}_{m+1}(a, b), \dots, \tilde{P}_{m+n^2-1}(a, b)).$$

If the corresponding collision multiplicity is

$$\mu = \max_{\mathcal{K}} |\{(b, m) \in \mathcal{S} : \Phi(b, m) = \mathcal{K}\}|,$$

then the blind-guessing probability for the full key-matrix sequence satisfies

$$\Pr[(\hat{K}_1, \dots, \hat{K}_{n^2}) = (K_1, \dots, K_{n^2})] \leq \frac{\mu}{|H| T_{\text{eff}}}.$$

This bound should be interpreted as a structured key-state guessing bound, not as a proof of modern indistinguishability security. It shows that the effective uncertainty of the key-matrix sequence is governed by the size of the subgroup H , the effective period T_{eff} , and possible collisions in the map from hidden states (b, m) to key matrices.

In particular, the key matrices are not independent random matrices from $GL_2(\mathbb{F}_q)$. They are consecutive terms of the same bi-periodic Pell matrix sequence. Therefore, the effective key space is not the full space $GL_2(\mathbb{F}_q)$, but the structured family generated by the hidden state (b, m) .

The small numerical example in Section 5 uses $q = 47$ only for illustration. In that case, the group sizes and effective search spaces are trivially small and provide no cryptographic security. For practical parameter choices, the subgroup H , the subgroup order q , and the effective period T_{eff} must be sufficiently large.

Finally, even if recovering b or blindly guessing the full key state is computationally infeasible, this does not imply semantic security for the deterministic construction. Under known-plaintext or chosen-plaintext settings, the algebraic relation $M_i = B_i \star K_i$ may reveal structural information if the same session parameters are reused. These limitations are discussed separately in the following subsections.

6.3. Chosen Point-Block Reconstruction Limitation

Proposition 3. Let $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$ be a cyclic subgroup of prime order q , and let $\mathcal{O}$ denote the point at infinity. Assume that the same session parameters a , b , and m are reused without nonce-dependent randomization or point masking. For a fixed block index i , let

$$K_i = \tilde{P}_{m+i-1}(a, b) = \begin{pmatrix} k_{11}^{(i)} & k_{12}^{(i)} \\ k_{21}^{(i)} & k_{22}^{(i)} \end{pmatrix} \in GL_2(\mathbb{F}_q).$$

Suppose that an adversary is allowed to submit chosen point blocks in $G^{2 \times 2}$ at the same block index i . If the adversary submits the special block

$$B_i^* = \begin{pmatrix} G_0 & \mathcal{O} \\ \mathcal{O} & G_0 \end{pmatrix},$$

then the corresponding ciphertext block

$$M_i^* = B_i^* \star K_i$$

is

$$M_i^* = \begin{pmatrix} [k_{11}^{(i)}]G_0 & [k_{12}^{(i)}]G_0 \\ [k_{21}^{(i)}]G_0 & [k_{22}^{(i)}]G_0 \end{pmatrix}.$$

Therefore, the chosen point block reveals the elliptic-curve point images of all scalar entries of K_i . If the elliptic-curve discrete logarithm problem in G is efficiently solvable, then the scalar entries

$$k_{11}^{(i)}, k_{12}^{(i)}, k_{21}^{(i)}, k_{22}^{(i)}$$

can be recovered modulo q . Hence, under deterministic parameter reuse and a chosen point-block model, recovery of the scalar key matrix K_i reduces to solving discrete logarithms in G .

Proof. By the definition of the elliptic-curve scalar matrix action, for

$$B = \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix}$$

and

$$K_i = \begin{pmatrix} k_{11}^{(i)} & k_{12}^{(i)} \\ k_{21}^{(i)} & k_{22}^{(i)} \end{pmatrix},$$

we have

$$B \star K_i = \begin{pmatrix} [k_{11}^{(i)}]P_1 \oplus [k_{21}^{(i)}]P_2 & [k_{12}^{(i)}]P_1 \oplus [k_{22}^{(i)}]P_2 \\ [k_{11}^{(i)}]P_3 \oplus [k_{21}^{(i)}]P_4 & [k_{12}^{(i)}]P_3 \oplus [k_{22}^{(i)}]P_4 \end{pmatrix}.$$

Now choose

$$B_i^* = \begin{pmatrix} G_0 & \mathcal{O} \\ \mathcal{O} & G_0 \end{pmatrix}.$$

Then $P_1 = G_0$, $P_2 = \mathcal{O}$, $P_3 = \mathcal{O}$, and $P_4 = G_0$. Since

$$[k]\mathcal{O} = \mathcal{O}$$

and

$$P \oplus \mathcal{O} = P$$

for every $P \in G$, we obtain

$$B_i^* \star K_i = \begin{pmatrix} [k_{11}^{(i)}]G_0 & [k_{12}^{(i)}]G_0 \\ [k_{21}^{(i)}]G_0 & [k_{22}^{(i)}]G_0 \end{pmatrix}.$$

Thus, the ciphertext block directly exposes the point images of the scalar entries of K_i . If ECDLP in G is efficiently solvable, then each scalar $k_{uv}^{(i)}$ can be recovered from the corresponding point $[k_{uv}^{(i)}]G_0$. Therefore, the scalar key matrix K_i can be reconstructed. $\square$

However, this result should be interpreted as a limitation of deterministic parameter reuse under a chosen point-block model. It does not imply that K_i can be recovered efficiently in a properly chosen elliptic-curve subgroup where ECDLP is computationally infeasible. It also does not represent a standard IND-CPA proof or a complete security reduction. Rather, it shows that without nonce-dependent randomization or point masking, the deterministic scalar matrix action can expose algebraic information when special plaintext point blocks are allowed.

In a standard symbol-level chosen-plaintext setting, the adversary may be restricted by the public symbol-to-point encoding map. If the encoding map does not allow the adversary to construct arbitrary point blocks in $G^{2 \times 2}$, especially blocks containing G_0 and the point at infinity $\mathcal{O}$ in prescribed positions, then the exact reconstruction mechanism described above may not be directly applicable. Nevertheless, the proposition demonstrates that deterministic parameter reuse should not be regarded as sufficient for modern chosen-plaintext security.

Similarly, in a known-plaintext setting, an adversary may observe plaintext-ciphertext block pairs (B_i, M_i) satisfying $M_i = B_i \star K_i$. Although this relation is not an ordinary linear equation over $\mathbb{F}_q$, because B_i and M_i are point-valued matrices, it still represents a deterministic algebraic relation between the plaintext block, the ciphertext block, and the scalar key matrix. Therefore, repeated use of the same session parameters a , b , and m may reveal equality patterns or structural relations between blocks.

Consequently, the deterministic version of the proposed construction is not claimed to provide semantic security or IND-CPA security as a standalone encryption scheme. The result above is included to clarify the algebraic limitation of the construction under chosen point-block access. To obtain resistance against standard chosen-plaintext attacks, the construction would need to be extended with nonce-dependent randomization, point masking, fresh session parameters, or integration into an authenticated-encryption framework.

6.4. Effective Key-State Size and Brute-Force Considerations

The effective secret material of the deterministic construction is determined by the shared scalar parameter b and the pre-agreed initial index m . Since the scalar parameter a is transmitted as part of the ciphertext pair (a, M) , it is regarded as public information and is not counted as an independent secret component. For fixed a, b and m , the block-dependent key matrices are determined by $K_i = \tilde{P}_{m+i-1}(a, b)$, $1 \leq i \leq n^2$. Thus, once b

and m are known, the full key-matrix sequence $K_1, K_2, \dots, K_{n^2}$ can be reconstructed. Let $\mathcal{B}$ denote the admissible set of values for b , and let $\mathcal{M}$ denote the admissible set of initial indices m . A direct brute-force adversary may test candidate pairs $(b', m') \in \mathcal{B} \times \mathcal{M}$ by computing $K'_i = \tilde{P}_{m'+i-1}(a, b')$ and checking whether $M_i = B_i \star K'_i$ holds for known plaintext-ciphertext block pairs. The nominal brute-force search space is therefore

$$|\mathcal{B}| \cdot |\mathcal{M}|,$$

and the nominal key-state size is

$$\log_2 |\mathcal{B}| + \log_2 |\mathcal{M}|.$$

By Proposition 2, the contribution of m must be interpreted with respect to the finite period T_k of the bi-periodic Pell key-matrix sequence over $\mathbb{F}_q$. If

$$m \equiv m' \pmod{T_{\text{eff}}},$$

then m and m' generate the same shifted key-matrix sequence. Therefore, the effective contribution of m is bounded by

$$\min(|\mathcal{M}|, T_{\text{eff}}).$$

Hence, the effective brute-force key-state space is more accurately expressed as

$$|\mathcal{B}| \cdot \min(|\mathcal{M}|, T_{\text{eff}}),$$

with effective key-state size

$$\log_2 |\mathcal{B}| + \log_2 (\min(|\mathcal{M}|, T_{\text{eff}})).$$

This estimate should not be confused with the size of the full matrix group $GL_2(\mathbb{F}_q)$. The key matrices are not independently sampled from $GL_2(\mathbb{F}_q)$; rather, they are consecutive terms of the structured sequence $K_i = \tilde{P}_{m+i-1}(a, b)$. Therefore, the effective key space is governed by the hidden state (b, m) , not by

$$|GL_2(\mathbb{F}_q)|^{n^2}.$$

The scalar parameter b is derived from a Diffie-Hellman-type agreement over a subgroup

$$H = \langle \beta \rangle \subseteq \mathbb{F}_q^\times.$$

If $b \in H$, then

$$|\mathcal{B}| \leq |H|.$$

Recovering $b = \beta^{re}$ from the public values β, β^r, β^e corresponds to a computational Diffie-Hellman-type problem in H . Therefore, the difficulty of recovering b depends on the size of H and the hardness of the discrete logarithm and computational Diffie-Hellman problems in that subgroup. Known plaintext-ciphertext pairs may help an adversary verify candidate hidden states. For example, if (B_i, M_i) is known for a fixed block index i , then a candidate pair (b', m') can be tested by checking whether

$$M_i = B_i \star \tilde{P}_{m'+i-1}(a, b')$$

holds. Thus, known-plaintext information can reduce the cost of verifying brute-force guesses, even if it does not directly determine the correct hidden state.

The small numerical example over $\mathbb{F}_{47}$ has a very limited search space and is not cryptographically secure. It is included only to illustrate the algebraic operation of the construction. For practical security, the subgroup H , the elliptic-curve subgroup G , and the effective period T_{eff} must be sufficiently large. Moreover, deterministic reuse of the same parameters a , b , and m should be avoided; nonce-dependent key derivation,

point masking, or fresh session parameters are required for a deployable cryptographic scheme.

6.5. Computational Complexity Analysis

In this section, the computational cost of the proposed deterministic algebraic block transformation is analyzed in terms of finite-field operations over $\mathbb{F}_q$ and elliptic-curve group operations in G . Let the plaintext matrix be $B \in G^{2n \times 2n}$. Since B is divided into 2×2 point blocks, the total number of plaintext blocks is $N_b = n^2$. For each block index i , the block-dependent key matrix is $K_i = \tilde{P}_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q)$. The encryption of the i -th block is given by $M_i = B_i \star K_i$. Here, $\star$ denotes the elliptic-curve scalar matrix action. This action consists of scalar multiplication of elliptic-curve points by entries of K_i , followed by point addition in the subgroup G . Let C_{SM} denote the cost of one elliptic-curve scalar multiplication in G , let C_{PA} denote the cost of one elliptic-curve point addition in G , and let $C_{\mathbb{F}_q}$ denote the cost of one finite-field operation over $\mathbb{F}_q$ [3,4,20,24–26]

6.5.1. Cost of Parameter Agreement

The scalar parameters a and b are generated by a Diffie-Hellman-type agreement over a multiplicative subgroup $H = \langle \beta \rangle \subseteq \mathbb{F}_q^\times$. Alice chooses a secret exponent r and computes $E_1 \equiv \beta^r \pmod{q}$. Bob chooses a secret exponent e and computes $a \equiv \beta^e \pmod{q}$, and $b \equiv E_1^e \pmod{q}$. Alice can recover the same shared scalar parameter by computing $b \equiv a^r \pmod{q}$. Thus, the parameter agreement phase consists of modular exponentiations in $\mathbb{F}_q^\times$. Using the square-and-multiply method, each exponentiation requires $O(\log q)$ finite-field multiplications over $\mathbb{F}_q$. Since this phase is performed once per session, its cost is independent of the number of message blocks. Therefore, it does not dominate the total computational cost for messages containing many 2×2 blocks.

6.5.2. Cost of Generating the Key Matrices

For each block index i , the corresponding block-dependent key matrix is $K_i = \tilde{P}_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q)$. The key matrices are generated over the scalar field $\mathbb{F}_q$. Since they are consecutive terms of the bi-periodic Pell matrix sequence, they can be computed sequentially from the recurrence relation. After the initial matrix $\tilde{P}_m(a, b)$ is obtained, each subsequent matrix $K_{i+1} = \tilde{P}_{m+i}(a, b)$ requires only a constant number of finite-field operations over $\mathbb{F}_q$. Therefore, if the total number of plaintext blocks is $N_b = n^2$, then generating the required sequence $K_1, K_2, \dots, K_{N_b}$ requires $O(N_b)$ finite-field operations after initialization. If the initial index m is large, computing $\tilde{P}_m(a, b)$ by naive iteration from the beginning of the sequence would require $O(m)$ finite-field operations. For practical implementation, the initial state should instead be computed using a fast recurrence method or a matrix-exponentiation method, reducing the initialization cost to $O(\log m)$ finite-field operations. Thus, the total cost of generating all required key matrices can be expressed as $O(\log m + N_b)$ finite-field operations over $\mathbb{F}_q$. Since each K_i is a 2×2 scalar matrix, the per-block key-generation cost is constant. Moreover, this finite-field cost is typically much smaller than the elliptic-curve scalar multiplication cost required by the $\star$ -action.

6.5.3. Encryption Complexity per Block

Let $B_i \in G^{2 \times 2}$ and $K_i \in GL_2(\mathbb{F}_q)$. The encryption of the i -th block is given by the elliptic-curve scalar matrix action $M_i = B_i \star K_i$. Writing

$$M_i = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix},$$

we obtain

$$\begin{aligned} Q_{11} &= [k_{11}^{(i)}]P_{11} \oplus [k_{21}^{(i)}]P_{12}, \\ Q_{12} &= [k_{12}^{(i)}]P_{11} \oplus [k_{22}^{(i)}]P_{12}, \\ Q_{21} &= [k_{11}^{(i)}]P_{21} \oplus [k_{21}^{(i)}]P_{22}, \\ Q_{22} &= [k_{12}^{(i)}]P_{21} \oplus [k_{22}^{(i)}]P_{22}. \end{aligned}$$

Each output point requires two elliptic-curve scalar multiplications and one elliptic-curve point addition. Since each ciphertext block contains four output points, the encryption of one 2×2 block requires 8 elliptic-curve scalar multiplications and 4 elliptic-curve point additions. Therefore, the encryption cost for one block is

$$8C_{SM} + 4C_{PA} + O(1)C_{\mathbb{F}_q}.$$

Here, C_{SM} denotes the cost of one elliptic-curve scalar multiplication in G , C_{PA} denotes the cost of one elliptic-curve point addition in G , and $C_{\mathbb{F}_q}$ denotes the cost of one finite-field operation over $\mathbb{F}_q$. Since the total number of 2×2 blocks is $N_b = n^2$, the total encryption cost is

$$N_b(8C_{SM} + 4C_{PA}) + O(N_b)C_{\mathbb{F}_q}.$$

Equivalently, since $N_b = n^2$, the dominant encryption cost is $O(n^2C_{SM})$. This is because elliptic-curve scalar multiplication is typically much more expensive than finite-field additions and multiplications used in the generation of the scalar key matrices.

6.5.4. Decryption Complexity per Block

For decryption, the receiver first reconstructs the same block-dependent key matrix K_i . Then the inverse matrix K_i^{-1} is computed over the scalar field $\mathbb{F}_q$. Since K_i is a 2×2 matrix, its inverse can be computed as

$$K_i^{-1} = \det(K_i)^{-1} \begin{pmatrix} k_{22}^{(i)} & -k_{12}^{(i)} \\ -k_{21}^{(i)} & k_{11}^{(i)} \end{pmatrix}.$$

Thus, computing K_i^{-1} requires one finite-field inversion in $\mathbb{F}_q$ and a constant number of finite-field multiplications and additions, provided that $\det(K_i) \neq 0$. By Lemma 2, this condition is satisfied because $K_i \in GL_2(\mathbb{F}_q)$. After K_i^{-1} is obtained, decryption is performed by $B_i = M_i \star K_i^{-1}$. The $\star$ -action has the same elliptic-curve operation structure as in encryption. Therefore, for each 2×2 ciphertext block, decryption requires 8 elliptic-curve scalar multiplications and 4 elliptic-curve point additions. Let C_{Inv} denote the cost of computing one 2×2 matrix inverse over $\mathbb{F}_q$. Then the decryption cost for one block is $8C_{SM} + 4C_{PA} + C_{Inv}$. Since the total number of blocks is $N_b = n^2$, the total decryption cost is $N_b(8C_{SM} + 4C_{PA} + C_{Inv})$. Equivalently, since $N_b = n^2$, the dominant decryption cost is $O(n^2C_{SM})$. This is because C_{Inv} is a finite-field cost over $\mathbb{F}_q$, whereas elliptic-curve scalar multiplication is typically the dominant operation.

6.5.5. Overall Complexity

The total encryption cost consists of three main components:

parameter agreement cost + key-matrix generation cost + block encryption cost.

The parameter agreement phase requires modular exponentiations over $\mathbb{F}_q^\times$, and its cost is $O(\log q)$ finite-field multiplications per exponentiation. The generation of the key-matrix sequence requires $O(\log m + N_b)$ finite-field operations over $\mathbb{F}_q$, where $N_b = n^2$ is the number of 2×2 blocks. The block encryption cost for all blocks is $N_b(8C_{SM} + 4C_{PA})$.

Therefore, the total encryption cost can be written as

$$C_{\text{enc}} = O(\log q) + O(\log m + N_b) + N_b(8C_{\text{SM}} + 4C_{\text{PA}}).$$

Since $N_b = n^2$, this becomes

$$C_{\text{enc}} = O(\log q) + O(\log m + n^2) + n^2(8C_{\text{SM}} + 4C_{\text{PA}}).$$

As elliptic-curve scalar multiplication is typically the dominant operation, the leading term of the encryption cost is $C_{\text{enc}} = O(n^2 C_{\text{SM}})$. Similarly, decryption consists of parameter reconstruction, key-matrix generation, inverse matrix computation over $\mathbb{F}_q$, and block decryption through the $\star$ -action. Hence,

$$C_{\text{dec}} = O(\log q) + O(\log m + N_b) + N_b(8C_{\text{SM}} + 4C_{\text{PA}} + C_{\text{INV}}).$$

Equivalently,

$$C_{\text{dec}} = O(\log q) + O(\log m + n^2) + n^2(8C_{\text{SM}} + 4C_{\text{PA}} + C_{\text{INV}}).$$

Since C_{INV} is a finite-field matrix inversion cost over $\mathbb{F}_q$, while elliptic-curve scalar multiplication is the dominant operation, the leading term of the decryption cost is also $C_{\text{dec}} = O(n^2 C_{\text{SM}})$. The main computational costs of the proposed construction are summarized in Table 2.

Table 2. Computational complexity summary.

Operation	Complexity
Parameter agreement	$O(\log q)$
Initial key-sequence generation	$O(\log m)$
Generation of all K_i matrices	$O(N_b) = O(n^2)$
Encryption per block	$8 C_{\text{SM}} + 4 C_{\text{PA}}$
Encryption of all blocks	$n^2 (8 C_{\text{SM}} + 4 C_{\text{PA}})$
Decryption per block	$8 C_{\text{SM}} + 4 C_{\text{PA}} + C_{\text{INV}}$
Decryption of all blocks	$n^2 (8 C_{\text{SM}} + 4 C_{\text{PA}} + C_{\text{INV}})$

Here, $N_b = n^2$ denotes the total number of 2×2 blocks. The dominant cost is elliptic-curve scalar multiplication, because each block transformation is performed through the elliptic-curve scalar matrix action $M_i = B_i \star K_i$ or, in decryption, $B_i = M_i \star K_i^{-1}$. Compared with a purely finite-field matrix transformation, the proposed construction has a higher computational cost because each scalar action on an elliptic-curve point requires elliptic-curve scalar multiplication. This additional cost arises from representing plaintext and ciphertext entries as elliptic-curve group elements and applying scalar matrix actions on point-valued blocks. Therefore, the proposed construction trades computational efficiency for a richer algebraic point-valued block structure. For practical implementations, the dominant cost remains the elliptic-curve scalar multiplication in the block transformation. The symbols and cost expressions used in the computational complexity analysis are listed in Table 3.

Table 3. Symbols used in the computational complexity analysis.

Symbol/Expression	Description
C_{SM}	Cost of one elliptic-curve scalar multiplication in G
C_{PA}	Cost of one elliptic-curve point addition in G .
C_{INV}	Cost of one 2×2 finite-field matrix inversion over $\mathbb{F}_q$.
$O(\log q)$	Cost order of one modular exponentiation in $\mathbb{F}_q^\times$ using square-and-multiply.
$O(\log m)$	Cost order of computing the initial term of the key sequence using a fast recurrence or matrix-exponentiation method.

$O(n^2)$	Cost order of sequentially generating all block-dependent key matrices for $N_b = n^2$ blocks.
$8 C_{SM} + 4 C_{PA}$	Cost of encrypting one 2×2 block.
$8 C_{SM} + 4 C_{PA} + C_{INV}$	Cost of decrypting one 2×2 block, including inverse matrix computation over $\mathbb{F}_q$.
$n^2 (8 C_{SM} + 4 C_{PA})$	Total encryption cost for all $N_b = n^2$ blocks.
$n^2 (8 C_{SM} + 4 C_{PA} + C_{INV})$	Total decryption cost for all $N_b = n^2$ blocks.

6.6. Qualitative Comparison with Related Frameworks

This subsection provides a qualitative and structural comparison between the proposed construction and related encryption frameworks. The purpose is not to claim practical efficiency, standardized security, or security equivalence with AES or established cryptographic schemes. Rather, the comparison is intended to position the proposed construction with respect to its algebraic domain, transformation structure, and dominant operations. AES is included only as a standardized reference point for symmetric block encryption. The classical Hill cipher and recursive matrix-based schemes are included to position the proposed construction within matrix-based cryptography. Ref. [14] is included because it is a Pell-based matrix encryption scheme closely related to the algebraic motivation of this work. ECC-based hybrid schemes are included to clarify the difference between standard hybrid approaches, where elliptic-curves are mainly used for key establishment, and the proposed construction, where elliptic-curve points are used directly as plaintext and ciphertext entries.

The comparison in Table 4 is qualitative and operation-level. It is not based on implementation-level runtime benchmarking, and it does not imply that the proposed construction achieves the same practical efficiency or formal security level as AES or other established cryptographic standards. In its present deterministic form, the proposed construction should be interpreted as a theoretical algebraic block transformation rather than as a deployable authenticated-encryption scheme.

Table 4. Qualitative Comparison of Related Algebraic and Cryptographic Frameworks.

Scheme	Main Algebraic Domain	Key and Transformation Structure	Operation	Role in Comparison/Limitation
AES	Byte-oriented finite field operations over $\mathbb{F}_{2^8}$.	Substitution-permutation network with fixed round structure.	S-box evaluations, XOR operations, and finite-field multiplications.	Standardized and highly optimized symmetric block cipher. Included only as a reference framework; direct runtime or security comparison requires implementation-level benchmarking and formal security analysis [27].
Classical Hill cipher	Modular arithmetic over $\mathbb{Z}_m$ or $\mathbb{F}_p$.	Fixed invertible matrix used for encryption.	Modular / finite-field matrix multiplication.	Efficient matrix-based transformation, but vulnerable to known-plaintext attacks when enough plaintext-ciphertext pairs are available to reconstruct the key matrix by linear algebra.
Fibonacci/Pell matrix-based schemes	Finite-fields or modular rings.	Key matrices generated from recursive sequences such as Fibonacci, Lucas, Pell, or related matrix sequences.	Recursive matrix generation and finite-field matrix multiplication.	Recursive matrices provide structured key generation, but encryption generally remains a finite-field or modular matrix transformation.

Pell-based scheme in Ref. [14]	Finite-field or modular matrix setting.	Pell or Pell-type matrix used as an encryption matrix.	Finite-field matrix operations.	Provides a Pell-matrix-based algebraic construction; however, plaintext and ciphertext entries are not generally represented as elliptic-curve points.
ECC-based hybrid schemes	Elliptic-curve group for key establishment and symmetric encryption domain for data encryption.	ECC is mainly used for key exchange, key encapsulation, or public-key encryption; data encryption is usually performed by a separate symmetric cipher.	EC scalar multiplication for key establishment; symmetric operations for data encryption.	ECC and data encryption are usually separated. The elliptic-curve component is not typically used as the direct algebraic domain for every plaintext block.
Proposed construction	Elliptic-curve group $G \subseteq E(\mathbb{F}_p)$ for point-valued plaintext and ciphertext blocks; scalar key matrices over $\mathbb{F}_q$	Block-dependent matrices $K_i = P_{m+i-1}(a, b) \in GL_2(\mathbb{F}_q)$ act on EC point-valued plaintext blocks through the right scalar matrix action.	Elliptic-curve scalar multiplication and point addition in G .	Theoretical algebraic framework combining bi-periodic Pell matrices with elliptic-curve point-valued block transformations. Computationally more expensive than purely finite-field matrix schemes and deterministic unless additional randomization or masking is added.

Note. This table provides a qualitative and operation-level comparison only. It is not based on implementation-level runtime benchmarks or formal security reductions. AES and established hybrid encryption schemes are included as reference frameworks, not as schemes claimed to be practically or security-wise equivalent to the proposed construction. In its deterministic form, the proposed construction should be interpreted as a theoretical algebraic block transformation unless additional randomization, authentication, and formal security proofs are provided.

7. Conclusions

In this paper, a theoretical algebraic block transformation based on bi-periodic Pell matrices and elliptic-curve point-valued blocks was presented. The proposed construction uses block-dependent scalar matrices $K_i \in GL_2(\mathbb{F}_q)$ to transform plaintext blocks whose entries lie in the elliptic-curve subgroup $G = \langle G_0 \rangle \subseteq E(\mathbb{F}_p)$. Thus, the encryption operation is not ordinary finite-field matrix multiplication. Instead, it is defined by the elliptic-curve scalar matrix action $M_i = B_i \star K_i$.

The main algebraic contribution of the construction is the combination of bi-periodic Pell matrix sequences over $\mathbb{F}_q$ with elliptic-curve point-valued block transformations over G . The construction distinguishes between the elliptic-curve base field $\mathbb{F}_p$, the subgroup order q , and the scalar field $\mathbb{F}_q$, thereby making the role of scalar multiplication on elliptic-curve points explicit.

The analysis shows that the block-dependent matrices are invertible over $\mathbb{F}_q$, so the decryption operation $B_i = M_i \star K_i^{-1}$ is algebraically well-defined. The scalar parameter b is derived through a Diffie-Hellman-type agreement over a multiplicative subgroup of $\mathbb{F}_q^\times$, while the initial index m determines the starting position of the bi-periodic Pell key-matrix sequence.

At the same time, the deterministic version of the construction has important limitations. Once the parameters a , b , and m are fixed, the key-matrix sequence is deterministic and consists of consecutive terms of a recursive matrix sequence rather than independent random matrices. Therefore, deterministic reuse of the same session parameters may reveal equality patterns or algebraic relations under known-plaintext or chosen-plaintext settings. In particular, under a chosen point-block model, specially structured inputs may reveal elliptic-curve point images of the scalar entries of K_i ; recovering the

corresponding scalar coefficients from these point images is related to the elliptic-curve discrete logarithm problem in G .

Consequently, the deterministic form of the proposed construction is not claimed to provide semantic security, IND-CPA security, or authenticated-encryption security as a standalone cryptographic scheme. The numerical example over $\mathbb{F}_{47}$ is included only to illustrate the algebraic steps of the construction and does not represent secure cryptographic parameters.

Overall, the proposed framework should be viewed as a theoretical algebraic construction rather than as an implementation-optimized competitor to AES or established authenticated-encryption schemes. Its main value lies in formalizing how bi-periodic Pell key matrices can act on elliptic-curve point-valued plaintext blocks through a scalar matrix action.

Future Work

Future work will focus on extending the deterministic algebraic construction with mechanisms required for modern cryptographic deployment. In particular, nonce-dependent key derivation, randomized point masking, fresh session parameters, and message authentication should be incorporated to reduce deterministic reuse and to support formal security analysis.

Further research may also determine the exact periods of bi-periodic Pell matrix sequences over finite fields, analyze possible collisions in the map from hidden states to key-matrix sequences, and study alternative recursive matrix families. Another important direction is to develop a randomized version of the construction and study its security under standard indistinguishability models such as IND-CPA or authenticated-encryption security. The implementation-level performance of such an enhanced version may then be compared with established cryptographic schemes through proper benchmarking.

Author Contributions: Conceptualization, E.A. and S.U.; methodology, E.A.; validation, E.A. and S.U.; formal analysis, E.A.; investigation, E.A.; resources, E.A. and S.U.; data curation, E.A.; writing—original draft preparation, E.A.; writing—review and editing, E.A. and S.U.; visualization, E.A.; supervision, S.U.; project administration, S.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. The numerical data used in the illustrative example are included in the article and its Appendix.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Points on $E(\mathbb{F}_{47})$ and Symbol-to-Point Mapping

This appendix collects supplementary numerical material used in the illustrative example of Section 5. The purpose of this appendix is to ensure reproducibility and transparency of the proposed construction, while keeping the main text focused on the algorithmic structure, security arguments, and complexity analysis. This subsection lists the affine points of the elliptic-curve $y^2 = x^3 + 3x + 41(\text{mod}47)$ as follows:

Table A1. Points on $E(\mathbb{F}_{47})$.

(2,14)	(2,33)	(10,15)	(10,32)	(11,18)	(11,29)	(13,16)	(13,31)	(14,17)	(14,30)
(16,7)	(16,40)	(19,1)	(19,46)	(20,8)	(20,39)	(21,23)	(21,24)	(24,5)	(24,42)
(27,21)	(27,26)	(28,9)	(28,38)	(30,23)	(30,24)	(33,13)	(33,34)	(34,22)	(34,25)

(35,4)	(35,43)	(38,15)	(38,32)	(40,10)	(40,37)	(41,18)	(41,29)	(42,18)	(42,29)
(43,23)	(43,24)	(45,11)	(45,36)	(46,15)	(46,32)	$\mathcal{O}$			

The point set was obtained by exhaustive enumeration over $\mathbb{F}_{47}$, verifying the curve equation for each candidate pair (x, y) . A total of 46 affine points were identified; together with the point at infinity $\mathcal{O}$, this yields $\#E(\mathbb{F}_{47}) = 47$. Since the group order is prime, the elliptic curve group is cyclic, and every non-identity point is a generator. In particular, the point $G_0 = (2, 14)$ generates the whole group. Hence, in this numerical example, we take $G = \langle G_0 \rangle = E(\mathbb{F}_{47})$. For the purpose of encrypting textual data in Section 5, each character (letters, digits, and selected symbols) is mapped to a unique point on $E(\mathbb{F}_{47})$. The mapping is fixed and publicly known, serving only as an encoding mechanism rather than a source of cryptographic secrecy. The assignment was constructed by selecting distinct points from the elliptic-curve group and associating them with characters in a deterministic manner. This mapping ensures a one-to-one correspondence between symbols and elliptic-curve points, enabling block-wise matrix construction over encoded plaintext.

Table A2. Character Table.

A	B	C	D	E	F	G	H	I	J
(2,14)	(28,9)	(21,24)	(33,34)	(40,10)	(11,29)	(42,29)	(45,11)	(27,26)	(35,4)
K	L	M	N	O	P	Q	R	S	T
(46,15)	(20,39)	(41,18)	(16,40)	(43,24)	(10,15)	(24,42)	(30,23)	(19,46)	(38,15)
U	V	W	X	Y	Z	0	1	2	3
(14,17)	(34,25)	(13,16)	(13,31)	(34,22)	(14,30)	(38,32)	(19,1)	(30,24)	(24,5)
4	5	6	7	8	9	-	!	@	#
(10,32)	(43,23)	(16,7)	(41,29)	(20,8)	(46,32)	(35,43)	(27,21)	(45,36)	(42,18)
\$	%	&	*	_	^				
(11,18)	(40,37)	(21,23)	(28,38)	(2,33)	(33,13)				

Note. Each table entry consists of a plaintext character and its corresponding elliptic-curve point. The special symbols -, !, @, #, \$, %, &, *, _ and ^ denote the hyphen, exclamation mark, at sign, number sign, dollar sign, percent sign, ampersand, asterisk, underscore, and caret, respectively. These symbols are included only as plaintext alphabet characters in the symbol-to-point encoding map and should not be interpreted as mathematical operations.

Appendix A.2. Complete Block Computations

This subsection provides the complete Pell matrix computations used in the numerical example of Section 5. The plaintext matrix is given as follows:

$$B = \begin{bmatrix} (2,14) & (2,33) & (28,9) & (27,26) & (35,43) & (10,15) & (40,10) & (30,23) \\ (27,26) & (43,24) & (33,34) & (27,26) & (21,24) & (2,33) & (10,15) & (40,10) \\ (20,39) & (20,39) & (2,33) & (41,18) & (2,14) & (38,15) & (30,23) & (27,26) \\ (13,31) & (35,43) & (28,9) & (2,14) & (19,46) & (40,10) & (33,34) & (2,33) \\ (40,10) & (16,40) & (21,24) & (30,23) & (34,22) & (10,15) & (38,15) & (27,26) \\ (43,24) & (16,40) & (2,33) & (41,18) & (40,10) & (38,15) & (45,11) & (43,24) \\ (33,34) & (2,33) & (43,24) & (34,25) & (40,10) & (30,23) & (2,33) & (40,10) \\ (21,24) & (21,24) & (2,33) & (2,33) & (2,33) & (2,33) & (2,33) & (2,33) \end{bmatrix}$$

Block matrices B_i are listed as follows:

$$B_1 = \begin{bmatrix} A & \bar{0} \\ I & \bar{0} \end{bmatrix} = \begin{bmatrix} (2,14) & (2,33) \\ (27,26) & (43,24) \end{bmatrix} \quad B_2 = \begin{bmatrix} B & I \\ D & I \end{bmatrix} = \begin{bmatrix} (28,9) & (27,26) \\ (33,34) & (27,26) \end{bmatrix}$$

$$\begin{aligned}
B_3 &= \begin{bmatrix} - & P \\ C & - \end{bmatrix} = \begin{bmatrix} (35,43) & (10,15) \\ (21,24) & (2,33) \end{bmatrix} \quad B_4 = \begin{bmatrix} E & R \\ P & E \end{bmatrix} = \begin{bmatrix} (40,10) & (30,23) \\ (10,15) & (40,10) \end{bmatrix} \\
B_5 &= \begin{bmatrix} L & L \\ X & - \end{bmatrix} = \begin{bmatrix} (20,39) & (20,39) \\ (13,31) & (35,43) \end{bmatrix} \quad B_6 = \begin{bmatrix} - & M \\ B & A \end{bmatrix} = \begin{bmatrix} (2,33) & (41,18) \\ (28,9) & (2,14) \end{bmatrix} \\
B_7 &= \begin{bmatrix} A & T \\ S & E \end{bmatrix} = \begin{bmatrix} (2,14) & (38,15) \\ (19,46) & (40,10) \end{bmatrix} \quad B_8 = \begin{bmatrix} R & I \\ D & - \end{bmatrix} = \begin{bmatrix} (30,23) & (27,26) \\ (33,34) & (2,33) \end{bmatrix} \\
B_9 &= \begin{bmatrix} E & N \\ O & N \end{bmatrix} = \begin{bmatrix} (40,10) & (16,40) \\ (43,24) & (16,40) \end{bmatrix} \quad B_{10} = \begin{bmatrix} C & R \\ - & M \end{bmatrix} = \begin{bmatrix} (21,24) & (30,23) \\ (2,33) & (41,18) \end{bmatrix} \\
B_{11} &= \begin{bmatrix} Y & P \\ E & T \end{bmatrix} = \begin{bmatrix} (34,22) & (10,15) \\ (40,10) & (38,15) \end{bmatrix} \quad B_{12} = \begin{bmatrix} T & I \\ H & O \end{bmatrix} = \begin{bmatrix} (38,15) & (27,26) \\ (45,11) & (43,24) \end{bmatrix} \\
B_{13} &= \begin{bmatrix} D & - \\ C & C \end{bmatrix} = \begin{bmatrix} (33,34) & (2,33) \\ (21,24) & (21,24) \end{bmatrix} \quad B_{14} = \begin{bmatrix} O & V \\ - & - \end{bmatrix} = \begin{bmatrix} (43,24) & (34,25) \\ (2,33) & (2,33) \end{bmatrix} \\
B_{15} &= \begin{bmatrix} E & R \\ - & - \end{bmatrix} = \begin{bmatrix} (40,10) & (30,23) \\ (2,33) & (2,33) \end{bmatrix} \quad B_{16} = \begin{bmatrix} - & E \\ - & - \end{bmatrix} = \begin{bmatrix} (2,33) & (40,10) \\ (2,33) & (2,33) \end{bmatrix}
\end{aligned}$$

Block matrices M_i are listed as follows:

$$\begin{aligned}
M_1 &= B_1 \star \tilde{P}_{29}(23,43) = \begin{bmatrix} (2,14) & (2,33) \\ (27,26) & (43,24) \end{bmatrix} \star \begin{bmatrix} 18 & 46 \\ 41 & 20 \end{bmatrix} \pmod{47} = \begin{bmatrix} (13,31) & (14,30) \\ (34,22) & (27,26) \end{bmatrix} \\
M_2 &= B_2 \star \tilde{P}_{30}(23,43) = \begin{bmatrix} (28,9) & (27,26) \\ (33,34) & (27,26) \end{bmatrix} \star \begin{bmatrix} 22 & 18 \\ 14 & 41 \end{bmatrix} \pmod{47} = \begin{bmatrix} (30,24) & (30,24) \\ (14,30) & (30,23) \end{bmatrix} \\
M_3 &= B_3 \star \tilde{P}_{31}(23,43) = \begin{bmatrix} (35,43) & (10,15) \\ (21,24) & (2,33) \end{bmatrix} \star \begin{bmatrix} 24 & 35 \\ 22 & 18 \end{bmatrix} \pmod{47} = \begin{bmatrix} (30,23) & (43,23) \\ (21,24) & (42,18) \end{bmatrix} \\
M_4 &= B_4 \star \tilde{P}_{32}(23,43) = \begin{bmatrix} (40,10) & (30,23) \\ (10,15) & (40,10) \end{bmatrix} \star \begin{bmatrix} 40 & 24 \\ 3 & 22 \end{bmatrix} \pmod{47} = \begin{bmatrix} (19,46) & (2,33) \\ (21,23) & (13,31) \end{bmatrix} \\
M_5 &= B_5 \star \tilde{P}_{33}(23,43) = \begin{bmatrix} (20,39) & (20,39) \\ (13,31) & (35,43) \end{bmatrix} \star \begin{bmatrix} 6 & 38 \\ 40 & 24 \end{bmatrix} \pmod{47} = \begin{bmatrix} (20,8) & (45,36) \\ (14,30) & (16,40) \end{bmatrix} \\
M_6 &= B_6 \star \tilde{P}_{34}(23,43) = \begin{bmatrix} (2,33) & (41,18) \\ (28,9) & (2,14) \end{bmatrix} \star \begin{bmatrix} 18 & 6 \\ 36 & 40 \end{bmatrix} \pmod{47} = \begin{bmatrix} (38,32) & (21,23) \\ (34,22) & (40,10) \end{bmatrix} \\
M_7 &= B_7 \star \tilde{P}_{35}(23,43) = \begin{bmatrix} (2,14) & (38,15) \\ (19,46) & (40,10) \end{bmatrix} \star \begin{bmatrix} 12 & 3 \\ 18 & 6 \end{bmatrix} \pmod{47} = \begin{bmatrix} (33,13) & (30,24) \\ (46,32) & (42,18) \end{bmatrix} \\
M_8 &= B_8 \star \tilde{P}_{36}(23,43) = \begin{bmatrix} (30,23) & (27,26) \\ (33,34) & (2,33) \end{bmatrix} \star \begin{bmatrix} 21 & 12 \\ 25 & 18 \end{bmatrix} \pmod{47} = \begin{bmatrix} (45,36) & (28,9) \\ (20,39) & (24,5) \end{bmatrix} \\
M_9 &= B_9 \star \tilde{P}_{37}(23,43) = \begin{bmatrix} (40,10) & (16,40) \\ (43,24) & (16,40) \end{bmatrix} \star \begin{bmatrix} 19 & 27 \\ 21 & 12 \end{bmatrix} \pmod{47} = \begin{bmatrix} (41,18) & (14,17) \\ (43,24) & (27,26) \end{bmatrix} \\
M_{10} &= B_{10} \star \tilde{P}_{38}(23,43) = \begin{bmatrix} (21,24) & (30,23) \\ (2,33) & (41,18) \end{bmatrix} \star \begin{bmatrix} 14 & 19 \\ 20 & 21 \end{bmatrix} \pmod{47} = \begin{bmatrix} (14,30) & (20,39) \\ (46,15) & (19,46) \end{bmatrix} \\
M_{11} &= B_{11} \star \tilde{P}_{39}(23,43) = \begin{bmatrix} (34,22) & (10,15) \\ (40,10) & (38,15) \end{bmatrix} \star \begin{bmatrix} 8 & 18 \\ 14 & 19 \end{bmatrix} \pmod{47} = \begin{bmatrix} (2,14) & (28,9) \\ (27,21) & \emptyset \end{bmatrix}
\end{aligned}$$

$$M_{12} = B_{12} \star \tilde{P}_{40}(23,43) = \begin{bmatrix} (38,15) & (27,26) \\ (45,11) & (43,24) \end{bmatrix} \star \begin{bmatrix} 29 & 8 \\ 1 & 14 \end{bmatrix} (mod 47) = \begin{bmatrix} (34,22) & (33,34) \\ (20,39) & (45,36) \end{bmatrix}$$

$$M_{13} = B_{13} \star \tilde{P}_{41}(23,43) = \begin{bmatrix} (33,34) & (2,33) \\ (21,24) & (21,24) \end{bmatrix} \star \begin{bmatrix} 13 & 44 \\ 29 & 8 \end{bmatrix} (mod 47) = \begin{bmatrix} (13,16) & (38,32) \\ (43,23) & (43,24) \end{bmatrix}$$

$$M_{14} = B_{14} \star \tilde{P}_{42}(23,43) = \begin{bmatrix} (43,24) & (34,25) \\ (2,33) & (2,33) \end{bmatrix} \star \begin{bmatrix} 14 & 13 \\ 31 & 29 \end{bmatrix} (mod 47) = \begin{bmatrix} (2,33) & (41,29) \\ (28,9) & (40,10) \end{bmatrix}$$

$$M_{15} = B_{15} \star \tilde{P}_{43}(23,43) = \begin{bmatrix} (40,10) & (30,23) \\ (2,33) & (2,33) \end{bmatrix} \star \begin{bmatrix} 2 & 18 \\ 14 & 13 \end{bmatrix} (mod 47) = \begin{bmatrix} (38,32) & (40,37) \\ (10,32) & (10,15) \end{bmatrix}$$

$$M_{16} = B_{16} \star \tilde{P}_{44}(23,43) = \begin{bmatrix} (2,33) & (40,10) \\ (2,33) & (2,33) \end{bmatrix} \star \begin{bmatrix} 40 & 2 \\ 12 & 14 \end{bmatrix} (mod 47) = \begin{bmatrix} (38,15) & (14,17) \\ (40,37) & (10,32) \end{bmatrix}$$

The ciphertext matrix M is assembled from the ciphertext blocks $M_1, M_2, \dots, M_{16}$ as follows:

$$M = \begin{bmatrix} (13,31) & (14,30) & (30,24) & (30,24) & (30,23) & (43,23) & (19,46) & (2,33) \\ (34,22) & (27,26) & (14,30) & (30,23) & (21,24) & (42,18) & (21,23) & (13,31) \\ (20,8) & (45,36) & (38,32) & (21,23) & (33,13) & (30,24) & (45,36) & (28,9) \\ (14,30) & (16,40) & (34,22) & (40,10) & (46,32) & (42,18) & (20,39) & (24,5) \\ (41,18) & (14,17) & (14,30) & (20,39) & (2,14) & (28,9) & (34,22) & (33,34) \\ (43,24) & (27,26) & (46,15) & (19,46) & (27,21) & \emptyset & (20,39) & (45,36) \\ (13,16) & (38,32) & (2,33) & (41,29) & (38,32) & (40,37) & (38,15) & (14,17) \\ (43,23) & (43,24) & (28,9) & (40,10) & (10,32) & (10,15) & (40,37) & (10,32) \end{bmatrix}$$

The recovered plaintext coincides with the original message, illustrating the correctness of the proposed algebraic construction for the numerical example.

$M =$

$$\begin{bmatrix} (13,31) & (14,30) & (30,24) & (30,24) & (30,23) & (43,23) & (19,46) & (2,33) \\ (34,22) & (27,26) & (14,30) & (30,23) & (21,24) & (42,18) & (21,23) & (13,31) \\ (20,8) & (45,36) & (38,32) & (21,23) & (33,13) & (30,24) & (45,36) & (28,9) \\ (14,30) & (16,40) & (34,22) & (40,10) & (46,32) & (42,18) & (20,39) & (24,5) \\ (41,18) & (14,17) & (14,30) & (20,39) & (2,14) & (28,9) & (34,22) & (33,34) \\ (43,24) & (27,26) & (46,15) & (19,46) & (27,21) & \emptyset & (20,39) & (45,36) \\ (13,16) & (38,32) & (2,33) & (41,29) & (38,32) & (40,37) & (38,15) & (14,17) \\ (43,23) & (43,24) & (28,9) & (40,10) & (10,32) & (10,15) & (40,37) & (10,32) \end{bmatrix}$$

Also, the ciphertext blocks M_i are listed as follows:

$$M_1 = \begin{bmatrix} (13,31) & (14,30) \\ (34,22) & (27,26) \end{bmatrix} \quad M_2 = \begin{bmatrix} (30,24) & (30,24) \\ (14,30) & (30,23) \end{bmatrix} \quad M_3 = \begin{bmatrix} (30,23) & (43,23) \\ (21,24) & (42,18) \end{bmatrix}$$

$$M_4 = \begin{bmatrix} (19,46) & (2,33) \\ (21,23) & (13,31) \end{bmatrix} \quad M_5 = \begin{bmatrix} (20,8) & (45,36) \\ (14,30) & (16,40) \end{bmatrix} \quad M_6 = \begin{bmatrix} (38,32) & (21,23) \\ (34,22) & (40,10) \end{bmatrix}$$

$$M_7 = \begin{bmatrix} (33,13) & (30,24) \\ (46,32) & (42,18) \end{bmatrix} \quad M_8 = \begin{bmatrix} (45,36) & (28,9) \\ (20,39) & (24,5) \end{bmatrix} \quad M_9 = \begin{bmatrix} (41,18) & (14,17) \\ (43,24) & (27,26) \end{bmatrix}$$

$$M_{10} = \begin{bmatrix} (14,30) & (20,39) \\ (46,15) & (19,46) \end{bmatrix} \quad M_{11} = \begin{bmatrix} (2,14) & (28,9) \\ (27,21) & \emptyset \end{bmatrix} \quad M_{12} = \begin{bmatrix} (34,22) & (33,34) \\ (20,39) & (45,36) \end{bmatrix}$$

$$M_{13} = \begin{bmatrix} (13,16) & (38,32) \\ (43,23) & (43,24) \end{bmatrix} \quad M_{14} = \begin{bmatrix} (2,33) & (41,29) \\ (28,9) & (40,10) \end{bmatrix} \quad M_{15} = \begin{bmatrix} (38,32) & (40,37) \\ (10,32) & (10,15) \end{bmatrix}$$

$$M_{16} = \begin{bmatrix} (38,15) & (14,17) \\ (40,37) & (10,32) \end{bmatrix}$$

$$\tilde{P}_{29}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 29 & 32 \\ 4 & 12 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{30}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 36 & 14 \\ 37 & 9 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{31}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 45 & 30 \\ 39 & 13 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{32}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 39 & 13 \\ 31 & 41 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{33}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 9 & 21 \\ 32 & 14 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{34}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 15 & 33 \\ 10 & 42 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{35}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 16 & 39 \\ 46 & 32 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{36}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 40 & 36 \\ 28 & 31 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{37}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 27 & 45 \\ 35 & 31 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{38}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 32 & 27 \\ 21 & 37 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{39}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 36 & 3 \\ 18 & 30 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{40}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 22 & 21 \\ 32 & 12 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{41}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 34 & 1 \\ 6 & 20 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{42}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 41 & 27 \\ 21 & 36 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{43}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 38 & 45 \\ 35 & 42 \end{bmatrix}(\text{mod}47)$$

$$\tilde{P}_{44}^{-1}(23,43)(\text{mod}47) \equiv \begin{bmatrix} 23 & 37 \\ 34 & 12 \end{bmatrix}(\text{mod}47)$$

Combining the recovered blocks gives the following plaintext matrix:

$$B_1 = \begin{bmatrix} (13,31) & (14,30) \\ (34,22) & (27,26) \end{bmatrix} \star \begin{bmatrix} 29 & 32 \\ 4 & 12 \end{bmatrix} = \begin{bmatrix} (2,14) & (2,33) \\ (27,26) & (43,24) \end{bmatrix}$$

$$B_2 = \begin{bmatrix} (30,24) & (30,24) \\ (14,30) & (30,23) \end{bmatrix} \star \begin{bmatrix} 36 & 14 \\ 37 & 9 \end{bmatrix} = \begin{bmatrix} (28,9) & (27,26) \\ (33,34) & (27,26) \end{bmatrix}$$

$$B_3 = \begin{bmatrix} (30,23) & (43,23) \\ (21,24) & (42,18) \end{bmatrix} \star \begin{bmatrix} 45 & 30 \\ 39 & 13 \end{bmatrix} = \begin{bmatrix} (35,43) & (10,15) \\ (21,24) & (2,33) \end{bmatrix}$$

$$B_4 = \begin{bmatrix} (19,46) & (2,33) \\ (21,23) & (13,31) \end{bmatrix} \star \begin{bmatrix} 39 & 13 \\ 31 & 41 \end{bmatrix} = \begin{bmatrix} (40,10) & (30,23) \\ (10,15) & (40,10) \end{bmatrix}$$

$$B_5 = \begin{bmatrix} (20,8) & (45,36) \\ (14,30) & (16,40) \end{bmatrix} \star \begin{bmatrix} 9 & 21 \\ 32 & 14 \end{bmatrix} = \begin{bmatrix} (20,39) & (20,39) \\ (13,31) & (35,43) \end{bmatrix}$$

$$B_6 = \begin{bmatrix} (38,32) & (21,23) \\ (34,22) & (40,10) \end{bmatrix} \star \begin{bmatrix} 15 & 33 \\ 10 & 42 \end{bmatrix} = \begin{bmatrix} (2,33) & (41,18) \\ (28,9) & (2,14) \end{bmatrix}$$

$$B_7 = \begin{bmatrix} (33,13) & (30,24) \\ (46,32) & (42,18) \end{bmatrix} \star \begin{bmatrix} 16 & 39 \\ 46 & 32 \end{bmatrix} = \begin{bmatrix} (2,14) & (38,15) \\ (19,46) & (40,10) \end{bmatrix}$$

$$B_8 = \begin{bmatrix} (45,36) & (28,9) \\ (20,39) & (24,5) \end{bmatrix} \star \begin{bmatrix} 40 & 36 \\ 28 & 31 \end{bmatrix} = \begin{bmatrix} (30,23) & (27,26) \\ (33,34) & (2,33) \end{bmatrix}$$

$$B_9 = \begin{bmatrix} (41,18) & (14,17) \\ (43,24) & (27,26) \end{bmatrix} \star \begin{bmatrix} 27 & 45 \\ 35 & 31 \end{bmatrix} = \begin{bmatrix} (40,10) & (16,40) \\ (43,24) & (16,40) \end{bmatrix}$$

$$B_{10} = \begin{bmatrix} (14,30) & (20,39) \\ (46,15) & (19,46) \end{bmatrix} \star \begin{bmatrix} 32 & 27 \\ 21 & 37 \end{bmatrix} = \begin{bmatrix} (21,24) & (30,23) \\ (2,33) & (41,18) \end{bmatrix}$$

$$B_{11} = \begin{bmatrix} (2,14) & (28,9) \\ (27,21) & \emptyset \end{bmatrix} \star \begin{bmatrix} 36 & 3 \\ 18 & 30 \end{bmatrix} = \begin{bmatrix} (34,22) & (10,15) \\ (40,10) & (38,15) \end{bmatrix}$$

$$B_{12} = \begin{bmatrix} (34,22) & (33,34) \\ (20,39) & (45,36) \end{bmatrix} \star \begin{bmatrix} 22 & 21 \\ 32 & 12 \end{bmatrix} = \begin{bmatrix} (38,15) & (27,26) \\ (45,11) & (43,24) \end{bmatrix}$$

$$B_{13} = \begin{bmatrix} (13,16) & (38,32) \\ (43,23) & (43,24) \end{bmatrix} \star \begin{bmatrix} 34 & 1 \\ 6 & 20 \end{bmatrix} = \begin{bmatrix} (33,34) & (2,33) \\ (21,24) & (21,24) \end{bmatrix}$$

$$B_{14} = \begin{bmatrix} (2,33) & (41,29) \\ (28,9) & (40,10) \end{bmatrix} \star \begin{bmatrix} 41 & 27 \\ 21 & 36 \end{bmatrix} = \begin{bmatrix} (43,24) & (34,25) \\ (2,33) & (2,33) \end{bmatrix}$$

$$B_{15} = \begin{bmatrix} (38,32) & (40,37) \\ (10,32) & (10,15) \end{bmatrix} \star \begin{bmatrix} 38 & 45 \\ 35 & 42 \end{bmatrix} = \begin{bmatrix} (40,10) & (30,23) \\ (2,33) & (2,33) \end{bmatrix}$$

$$B_{16} = \begin{bmatrix} (38,15) & (14,17) \\ (40,37) & (10,32) \end{bmatrix} \star \begin{bmatrix} 23 & 32 \\ 34 & 12 \end{bmatrix} = \begin{bmatrix} (2,33) & (40,10) \\ (2,33) & (2,33) \end{bmatrix}$$

The reconstructed plaintext matrix is obtained as follows:

$$B = \begin{bmatrix} (2,14) & (2,33) & (28,9) & (27,26) & (35,43) & (10,15) & (40,10) & (30,23) \\ (27,26) & (43,24) & (33,34) & (27,26) & (21,24) & (2,33) & (10,15) & (40,10) \\ (20,39) & (20,39) & (2,33) & (41,18) & (2,14) & (38,15) & (30,23) & (27,26) \\ (13,31) & (35,43) & (28,9) & (2,14) & (19,46) & (40,10) & (33,34) & (2,33) \\ (40,10) & (16,40) & (21,24) & (30,23) & (34,22) & (10,15) & (38,15) & (27,26) \\ (43,24) & (16,40) & (2,33) & (41,18) & (40,10) & (38,15) & (45,11) & (43,24) \\ (33,34) & (2,33) & (43,24) & (34,25) & (40,10) & (30,23) & (2,33) & (40,10) \\ (21,24) & (21,24) & (2,33) & (2,33) & (2,33) & (2,33) & (2,33) & (2,33) \end{bmatrix}$$

References

1. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209.
2. Miller, V.S. Use of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO '85, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 1986; Volume 218, pp. 417–426.

3. Silverman, J.H. *The Arithmetic of Elliptic Curves*, 2nd ed.; Springer: New York, NY, USA, 2009.
4. Hankerson, D.; Menezes, A.; Vanstone, S. *Guide to Elliptic Curve Cryptography*; Springer: New York, NY, USA, 2004.
5. Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
6. Barker, E.; Chen, L.; Keller, S.; Roginsky, A.; Vassilev, A.; Davis, R. *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography*, NIST Special Publication 800-56A Revision 3; NIST: Gaithersburg, MD, USA, 2018.
7. Hill, L.S. Cryptography in an algebraic alphabet. *Am. Math. Mon.* **1929**, *36*, 306–312.
8. Coskun, A.; Taskara, N. The matrix sequence in terms of the bi-periodic Fibonacci sequence. *Commun. Fac. Sci. Univ. Ank. Ser. A1 Math. Stat.* **2019**, *68*, 1939–1949.
9. Stakhov, A.; Massingue, V.; Sluchenko, A. *Introduction into Fibonacci Coding and Cryptography*; Osnova: Kharkov, Ukraine, 1999.
10. Basu, M.; Prasad, B. The generalized relations among the code elements for Fibonacci coding theory. *Chaos Solitons Fractals* **2009**, *41*, 2517–2525.
11. Wang, F.; Ding, J.; Dai, Z.; Peng, Y. An application of mobile phone encryption based on Fibonacci structure of chaos. In *Proceedings of the World Congress on Software Engineering*; IEEE: Piscataway, NJ, USA, 2010; Volume 2, pp. 97–100.
12. Prajapat, S.; Jain, A.; Thakur, R.S. A novel approach for information security with automatic variable key using Fibonacci Q-matrix. *Int. J. Comput. Commun. Technol.* **2012**, *3*, 54–57.
13. Prasad, K.; Mahato, H. Cryptography using generalized Fibonacci matrices with affine Hill cipher. *arXiv* **2020**, arXiv:2003.11936.
14. Kumari, M.; Tanti, J. On the role of the Fibonacci matrix as key in modified ECC. *arXiv* **2021**, arXiv:2112.11013.
15. Uçar, S.; Taş, N.; Özgür, N.Y. A new application to coding theory via Fibonacci and Lucas numbers. *Math. Sci. Appl. E-Notes* **2019**, *7*, 62–70.
16. Karatas, Z.; Gonen, B.; Luy, E. A public key cryptosystem based on matrices. *Int. J. Comput. Appl.* **2019**, *182*, 47–50.
17. Uygun, S.; Akinci, E. Matrix representation of bi-periodic Pell sequence. *J. Mahani Math. Res.* **2023**, *12*, 565–583.
18. Kaymak, Ö. Coding theory for $h(x)$ -Fibonacci polynomials. *BAUN Fen. Bilim. Enstitüsü Derg.* **2024**, *26*, 226–236.
19. Kannan, J.; Somanath, M.; Mahalakshmi, M.; Raja, K. Encryption decryption algorithm using solutions of Pell equation. *Int. J. Math. Its Appl.* **2022**, *10*, 1–8.
20. Washington, L.C. *Elliptic Curves: Number Theory and Cryptography*, 2nd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2008.
21. Kerckhoffs, A. La cryptographie militaire. *J. Des. Sci. Mil.* **1883**, *9*, 5–38.
22. Katz, J.; Lindell, Y. *Introduction to Modern Cryptography*; CRC Press: Boca Raton, FL, USA, 2015.
23. Goldreich, O. *Foundations of Cryptography*, Vols. 1–2; Cambridge University Press: Cambridge, UK, 2001–2004.
24. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 7th ed.; Pearson: New York, NY, USA, 2020.
25. Paar, C.; Pelzl, J. *Understanding Cryptography*; Springer: Berlin/Heidelberg, Germany, 2010.
26. Boneh, D.; Shoup, V. *A Graduate Course in Applied Cryptography*; Version 0.6; 2023. Available online: <https://toc.cryptobook.us/> (accessed on 19 May 2026).
27. National Institute of Standards and Technology. *Advanced Encryption Standard (AES), FIPS PUB 197, Updated 2023*; NIST: Gaithersburg, MD, USA, 2023.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.