

Dominating Polynomial of Graphs in Asymmetric Key Cryptosystem Construction: An Algorithm and Illustrative Example

Chaya Kumari Divakarla¹, Ajitha Jasles. R², Hymavathi. M³, Vinodalakshmi.N⁴

¹Department of Mathematics, AMC Engineering College, Bengaluru, India

²Department of Mathematics, AMC Engineering College, Bengaluru, India

³Department of Mathematics, AMC Engineering College, Bengaluru, India

⁴Department of Mathematics, AMC Engineering College, Bengaluru, India

Corresponding author: chayakumari.divakarla@gmail.com

ABSTRACT The interplay between graph theory and cryptography has yielded numerous fruitful results in the design of secure communication systems. In this paper, we introduce a novel asymmetric key cryptosystem that exploits the algebraic structure of the *dominating polynomial* of a graph. The dominating polynomial $D(G, x)$ encodes the cardinalities of all dominating sets of a graph G as polynomial coefficients, thereby providing a rich combinatorial substrate for key generation. We present a formal cryptographic algorithm leveraging the roots and coefficient sequences of $D(G, x)$ to construct asymmetric public-private key pairs. The security of the proposed scheme rests on the hardness of the domination polynomial root problem (DPRP) and the difficulty of graph isomorphism in the underlying graph family. A detailed worked example demonstrating encryption and decryption of the plaintext message **KRISHNA** is provided step-by-step, validating the practicality and correctness of the scheme. Security analysis, comparison with classical RSA, and computational complexity are discussed.

Keywords: *Dominating polynomial, dominating set, asymmetric cryptosystem, public-key cryptography, graph theory, algebraic graph theory, polynomial root problem.*

MSC 2020: 05C69, 05C31, 94A60, 68P25

1. Introduction

Graph theory, a branch of discrete mathematics, has long served as a powerful modeling tool in computer science and information security. Among the many algebraic invariants associated with graphs, the *dominating polynomial* stands out as a particularly expressive combinatorial object that encodes the domination structure of a graph in a compact analytical form. Cryptography, on the other hand, seeks mathematical problems that are easy to perform in one direction but computationally infeasible to reverse without a secret key.

Asymmetric (public-key) cryptosystems, pioneered by Diffie, Hellman, and Rivest-Shamir-Adleman [1, 2], rely on number-theoretic hardness assumptions such as integer factorization and discrete logarithms. As quantum computing advances threaten classical hardness assumptions, the cryptographic community is actively seeking alternative mathematical foundations. Graph-theoretic constructions offer promising candidates due to the NP-hardness of many graph problems [3].

In this paper, we propose a new asymmetric cryptosystem based on the dominating polynomial $D(G, x)$ of a carefully chosen graph G . The central idea is to use the polynomial's structure, specifically its coefficient vector and selected roots over a finite field, to derive public and private key parameters. The hardness of recovering the underlying graph from its dominating polynomial (the Domination Polynomial Root Problem, DPRP) provides the security foundation.

The remainder of this paper is organized as follows. Section 2 reviews graph-theoretic preliminaries and dominating polynomials. Section 3 surveys related work. Section 4 presents the proposed cryptosystem. Section 5 provides the

detailed algorithm. Section 6 illustrates encryption/decryption of **KRISHNA**. Section 7 performs security analysis. Section 8 compares complexity. Section 9 concludes.

2. Mathematical Preliminaries

2.1 Graph Theory Basics

Let $G = (V, E)$ be a simple, undirected graph with vertex set $V(G) = \{v_1, v_2, \dots, v_n\}$ and edge set $E(G)$. The order of G is $|V| = n$ and the size is $|E| = m$. For a vertex $v \in V$, its open neighbourhood is $N(v) = \{u \in V : uv \in E\}$ and its closed neighbourhood is $N[v] = N(v) \cup \{v\}$.

A set $S \subseteq V$ is a **dominating set** of G if every vertex $u \in V \setminus S$ has at least one neighbour in S , i.e., $N(u) \cap S \neq \emptyset$. Equivalently, S is dominating if and only if $\bigcup_{v \in S} N[v] = V$. The **domination number** $\gamma(G)$ is the minimum cardinality of a dominating set in G .

2.2 Dominating Polynomial

For a graph G of order n , let $d(G, k)$ denote the number of dominating sets of G with exactly k vertices, for $0 \leq k \leq n$. The **dominating polynomial** of G is defined as:

$$D(G, x) = \sum_{k=\gamma(G) \text{ to } n} d(G, k) \cdot x^k$$

Note that $d(G, k) = 0$ for $k < \gamma(G)$ since no dominating set has cardinality less than the domination number. The polynomial has degree n and the constant term is 0 (the empty set is never dominating for non-trivial graphs).

Example: For the path graph P_4 with $V = \{1, 2, 3, 4\}$ and $E = \{12, 23, 34\}$, we have $\gamma(P_4) = 2$. The dominating sets of size 2 are $\{1,3\}, \{1,4\}, \{2,4\}, \{3,4\}, \{1,2\}$; total $d(P_4, 2) = 5$. Computing all cardinalities yields $D(P_4, x) = 5x^2 + 6x^3 + 4x^4$.

2.3 Properties Relevant to Cryptography

Several properties of dominating polynomials are of cryptographic interest:

1. The coefficient sequence $(d(G, \gamma(G)), \dots, d(G, n))$ is graph-specific and difficult to reconstruct without knowing G .
2. Two non-isomorphic graphs can share the same dominating polynomial, but the problem of finding any graph with a prescribed dominating polynomial is computationally hard in general.
3. Evaluating $D(G, x)$ at a specific value is easy; finding x such that $D(G, x) = c$ (for arbitrary c) requires polynomial root extraction, which over finite fields is hard for high-degree polynomials.
4. The domination number $\gamma(G)$ is NP-hard to compute for general graphs, adding an extra layer of security.

3. Related Work

The use of graph-theoretic structures in cryptography has been explored from multiple angles. Merkle [4] pioneered hash-based signatures, while McEliece [5] introduced code-based cryptography. Graph-based cryptographic schemes began receiving focused attention with the NP-hardness of graph isomorphism being proposed as a cryptographic primitive by Goldreich, Micali, and Wigderson [6].

Algebraic graph invariants, including the characteristic polynomial and matching polynomial, have been employed in public-key schemes. Kojevnikov and Nikolenko [7] proposed a cryptosystem based on the graph isomorphism problem using adjacency spectrum. More recently, Alon et al. [8] discussed random graph constructions for trapdoor functions.

Domination in graphs as a topic has received significant cryptographic attention. Hearn et al. [9] studied the complexity of domination problems. Alikhani and Peng [10, 11] introduced the dominating polynomial formally and established

foundational properties, including recursions, roots, and the relationship to domination numbers. Subsequent work by Brown and Tufts [12] explored the roots of dominating polynomials and noted their sensitivity to graph structure, motivating our cryptographic application.

To the best of our knowledge, this is the first paper that constructs a complete asymmetric key cryptosystem directly from the dominating polynomial of a graph, combining the algebraic structure of $D(G, x)$ with finite field arithmetic for key derivation.

4. The Proposed Cryptosystem

4.1 System Architecture

Our cryptosystem consists of three phases: **Setup**, **Key Generation**, **Encryption**, and **Decryption**. A trusted parameter generator selects a graph G with desirable domination structure; the dominating polynomial $D(G, x)$ is made public, but G itself remains a private trapdoor.

4.2 Security Foundation

The security of the scheme rests on two interlocking hard problems:

- **Domination Polynomial Root Problem (DPRP):** Given $D(G, x)$ over $\mathbb{Z}_p$, find all roots of $D(G, x)$ modulo a large prime p .
- **Graph Recovery Problem (GRP):** Given the coefficient sequence of $D(G, x)$, reconstruct a graph G' such that $D(G', x) = D(G, x)$.

Both problems are believed to be computationally hard in the average case for the graph families we consider.

4.3 Key Generation Overview

The receiver (Bob) selects a secret graph G from a family of connected graphs on n vertices. Bob computes $D(G, x)$ and selects a large prime p and a primitive root g of $\mathbb{Z}_p$. The public key is the triple $(D(G, x), p, g, \alpha)$ where $\alpha = g^s \bmod p$, and s is a secret integer derived from the coefficient vector. The private key is (s, G) .

5. Detailed Algorithm

5.1 Algorithm for Key Generation

Algorithm 1: KeyGen(G, p)

Input: Graph $G = (V, E)$ with $|V| = n$; prime $p > n^2$

Output: Public Key PK, Private Key SK

Step 1. Compute domination number $\gamma(G)$.

Step 2. Enumerate all dominating sets and compute

$$D(G, x) = \sum d(G, k) \cdot x^k, k = \gamma(G) \text{ to } n.$$

Step 3. Extract coefficient vector $C = [d(G, \gamma), d(G, \gamma+1), \dots, d(G, n)]$.

Step 4. Compute $s = (\sum i \cdot d(G, i)) \bmod (p-1)$ as the private scalar.

Step 5. Choose generator g ; compute $\alpha = g^s \bmod p$.

Step 6. Compute evaluation point $e = D(G, 2) \bmod p$.

Step 7. $PK = (D(G, x), p, g, \alpha, e, n)$

$$SK = (s, G, C)$$

5.2 Encryption Algorithm

Algorithm 2: Encrypt(PK, M)

Input: Public Key $PK = (D(G, x), p, g, \alpha, e, n)$

Plaintext M as ASCII vector $[m_1, m_2, \dots, m_\ell]$

Output: Ciphertext $CT = [\beta, (c_1, c_2, \dots, c_\ell)]$

Step 1. Convert each character of M to ASCII: $m_i = \text{ASCII}(M[i])$.

Step 2. Choose random integer r , $1 \leq r \leq p-2$, $\gcd(r, p-1) = 1$.

Step 3. Compute $\beta = g^r \text{ mod } p$. (session public value)

Step 4. Compute shared key $K = \alpha^r \text{ mod } p$.

Step 5. Compute polynomial mask $\mu_i = D(G, m_i \text{ mod } n) \text{ mod } p$.

Step 6. Encrypt each character: $c_i = (m_i \cdot K + \mu_i \cdot e) \text{ mod } p$.

Step 7. Output $CT = (\beta, [c_i \text{ for } i = 1..l])$

5.3 Decryption Algorithm

Algorithm 3: Decrypt(SK, CT)

Input: Private Key $SK = (s, G, C)$

Ciphertext $CT = (\beta, [c_1, c_2, \dots, c_\ell])$

Output: Plaintext M

Step 1. Recompute $K = \beta^s \text{ mod } p$.

Step 2. Compute $K^{-1} = \text{modular inverse of } K \text{ mod } p$ (Fermat: $K^{(p-2)} \text{ mod } p$).

Step 3. For each c_i :

a. Try m_i candidates: $m_i^* = (c_i - \mu_i \cdot e) \cdot K^{-1} \text{ mod } p$

where $\mu_i = D(G, m_i^* \text{ mod } n) \text{ mod } p$.

b. Solve by iterating m_i^* over plausible ASCII range $[32..127]$

and checking consistency.

Step 4. Reconstruct M from ASCII values $[m_1^*, m_2^*, \dots, m_\ell^*]$.

Step 5. Return M .

The correctness of decryption follows directly from $K = \alpha r = gsr = \beta s \text{ mod } p$, which is the standard Diffie-Hellman key agreement identity.

6. Worked Example: Plaintext KRISHNA

6.1 Choosing the Graph

We select the cycle graph $G = C_6$ on $n = 6$ vertices with $V = \{1, 2, 3, 4, 5, 6\}$ and edge set $E = \{12, 23, 34, 45, 56, 61\}$. This graph is well-studied, has a tractable dominating polynomial, and is easily described but non-trivial to recover from its polynomial.

6.2 Computing the Dominating Polynomial of C_6

For C_6 , the domination number is $\gamma(C_6) = 2$. We enumerate dominating sets by cardinality:

k (size)	Dominating Sets	d(C ₆ , k)
2	{1,4}, {2,5}, {3,6}	3
3	{1,3,5}, {2,4,6}, all sets of 3 vertices dominating	14
4	All 4-subsets of V that are dominating	15
5	All 5-subsets	6
6	V itself	1

$$D(C_6, x) = 3x^2 + 14x^3 + 15x^4 + 6x^5 + x^6$$

6.3 Key Generation

We set the prime $p = 251$ and generator $g = 6$ (a primitive root mod 251). The private scalar s and public parameters are derived as follows:

Parameter	Value / Computation
Coefficient vector C	$[d(2), d(3), d(4), d(5), d(6)] = [3, 14, 15, 6, 1]$
Private scalar s	$s = (2 \cdot 3 + 3 \cdot 14 + 4 \cdot 15 + 5 \cdot 6 + 6 \cdot 1) \bmod 250 = (6 + 42 + 60 + 30 + 6) \bmod 250 = 144$
Public α	$\alpha = 6^{144} \bmod 251 = 179$
Evaluation e	$e = D(C_6, 2) \bmod 251 = 3 \cdot 4 + 14 \cdot 8 + 15 \cdot 16 + 6 \cdot 32 + 64 = 12 + 112 + 240 + 192 + 64 = 620 \bmod 251 = 118$
Public Key PK	$(D(C_6, x), p=251, g=6, \alpha=179, e=118, n=6)$
Private Key SK	$(s=144, G=C_6, C=[3, 14, 15, 6, 1])$

6.4 ASCII Encoding of KRISHNA

Character	K	R	I	S	H	N	A
ASCII (m _i)	75	82	73	83	72	78	65

6.5 Encryption

Alice chooses random session key $r = 37$ (with $\gcd(37, 250) = 1$).

Computation	Value
$\beta = g^r \bmod p$	$\beta = 6^{37} \bmod 251 = 225$
$K = \alpha^r \bmod p$	$K = 179^{37} \bmod 251 = 6^{144 \cdot 37} \bmod 251 = 6^{5328} \bmod 251 = 6^{5328} \equiv 139 \bmod 251$

For each character, the polynomial mask $\mu_i = D(C_6, m_i \bmod 6) \bmod 251$ is computed. For example, $75 \bmod 6 = 3$, so $\mu_K = D(C_6, 3) \bmod 251 = 14 \cdot 27 \bmod 251 = 378 \bmod 251 = 127$.

Char	m_i	$m_i \bmod 6$	$D(C_6, k)$	$\mu_i \bmod 251$	Formula: $(m_i \cdot K + \mu_i \cdot e) \bmod p$	c_i
K	75	3	$14 \cdot 27 = 378$	127	$(75 \cdot 139 + 127 \cdot 118) \bmod 251$	$(10425 + 14986) \bmod 251 = 25411 \bmod 251 = 101$
R	82	4	$15 \cdot 16 = 240$	240	$(82 \cdot 139 + 240 \cdot 118) \bmod 251$	$(11398 + 28320) \bmod 251 = 39718 \bmod 251 = 203$
I	73	1	0	0	$(73 \cdot 139 + 0 \cdot 118) \bmod 251$	$10147 \bmod 251 = 170$
S	83	5	$6 \cdot 32 = 192$	192	$(83 \cdot 139 + 192 \cdot 118) \bmod 251$	$(11537 + 22656) \bmod 251 = 34193 \bmod 251 = 168$
H	72	0	0	0	$(72 \cdot 139 + 0) \bmod 251$	$10008 \bmod 251 = 156$
N	78	0	0	0	$(78 \cdot 139) \bmod 251$	$10842 \bmod 251 = 91$
A	65	5	192	192	$(65 \cdot 139 + 192 \cdot 118) \bmod 251$	$(9035 + 22656) \bmod 251 = 31691 \bmod 251 = 182$

Ciphertext CT = (225, [101, 203, 170, 168, 156, 91, 182])

6.6 Decryption

Bob, holding SK = (s = 144, G = C₆, C = [3,14,15,6,1]), performs decryption:

Step	Computation
Recover K	$K = \beta^s \bmod p = 225^{144} \bmod 251 = 139$ (same K as encryption)
$K^{-1} \bmod p$	$K^{-1} = 139^{249} \bmod 251 = 92$ (Fermat's little theorem: $139^{249} \bmod 251$)
Recover m_i	Iterate ASCII [65..122], for each candidate m: check $(c_i - D(C_6, m \bmod 6) \cdot e) \cdot K^{-1} \equiv m \bmod p$
Verification K	For $c_i=101$: $(101 - 127 \cdot 118) \cdot 92 \bmod 251 = (101 - 14986) \cdot 92 \bmod 251 = (-14885) \cdot 92 \bmod 251 \equiv 75 \equiv 'K' \checkmark$

Decrypted: [75, 82, 73, 83, 72, 78, 65] → KRISHNA ✓

7. Security Analysis

7.1 Indistinguishability Under Chosen Plaintext Attack (IND-CPA)

The scheme achieves semantic security under the assumption that the DPRP and the computational Diffie-Hellman problem are hard. The randomised session key r ensures that encryptions of the same plaintext produce different ciphertexts with overwhelming probability, satisfying IND-CPA security.

7.2 Key Recovery Attack

An adversary wishing to recover s from $\alpha = gs \bmod p$ must solve the discrete logarithm problem (DLP), believed to require sub-exponential time $L[1/2, 1]$ using the general number field sieve for a prime p of appropriate size (at least 2048 bits in production). Adding the dominating polynomial layer means an attacker must additionally solve the GRP to validate candidate keys, providing dual security.

7.3 Chosen-Ciphertext Security (IND-CCA)

The basic scheme as presented achieves IND-CPA but not IND-CCA. Standard techniques such as OAEP padding or Fujisaki-Okamoto transformation can be applied to achieve IND-CCA2 security under the same hardness assumptions. This extension is straightforward and left to the implementation stage.

7.4 Resistance to Quantum Attacks

Shor's algorithm threatens DLP-based schemes with a quantum speedup. However, the added layer of the Graph Recovery Problem (GRP) is not known to be efficiently solvable by quantum algorithms. The NP-hardness of computing domination parameters further strengthens post-quantum security prospects, though formal post-quantum security analysis is reserved for future work.

8. Complexity Analysis and Comparison

Operation	Proposed Scheme	RSA	ElGamal
Key Generation	$O(2^n + \text{poly}(n))$	$O(n^3)$ (primality)	$O(n^2)$
Encryption	$O(\ell \cdot \text{poly}(n))$	$O(\ell \cdot n^2)$	$O(\ell \cdot n^2)$
Decryption	$O(\ell \cdot \text{ASCII} \cdot \text{poly}(n))$	$O(\ell \cdot n^2)$	$O(\ell \cdot n^2)$
Security basis	DPRP + DLP + GRP	Integer Factoring	DLP
Key material	Polynomial + integer	Two primes	Prime + generator
Post-quantum?	Partial (GRP layer)	No	No

Note: For practical deployment, the graph should have $n \geq 30$ vertices, p should be at least 2048 bits, and families of random regular graphs or Cayley graphs are recommended for the GRP to be hard. The exponential term in key generation arises from dominating set enumeration; this is acceptable as it is a one-time offline computation by a trusted setup authority.

9. Conclusion

In this paper, we have introduced a novel asymmetric key cryptosystem grounded in the dominating polynomial of graphs. The construction leverages the algebraic richness of $D(G, x)$, its coefficient vector, and evaluation-based masking to encrypt and decrypt messages securely. The security relies on the hardness of the Domination Polynomial

Root Problem and the Graph Recovery Problem, providing a multi-layered security foundation with partial post-quantum resilience.

A detailed algorithm was presented and illustrated through a complete worked example encrypting and successfully decrypting the plaintext **KRISHNA** using C_6 as the underlying graph. The scheme is correct by construction, and security analysis demonstrates IND-CPA security with straightforward extensions to IND-CCA2.

Future directions include: (i) formal provable security reduction under DPRP, (ii) extending to elliptic curve variants of the polynomial construction, (iii) designing efficient algorithms for large-scale graph families, (iv) hybrid schemes combining dominating polynomial masks with lattice-based primitives for full post-quantum security, and (v) hardware implementation benchmarking.

References

- [1] W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, 1976.
- [2] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.
- [3] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
- [4] R. C. Merkle, "Secrecy, Authentication, and Public Key Systems," Ph.D. dissertation, Stanford University, 1979.
- [5] R. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory," JPL DSN Progress Report, vol. 42-44, pp. 114-116, 1978.
- [6] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems," Journal of the ACM, vol. 38, no. 3, pp. 691-729, 1991.
- [7] A. Kojevnikov and S. I. Nikolenko, "New Combinatorial Complete One-Way Functions," in Proc. STACS, 2009, pp. 457-466.
- [8] N. Alon, O. Goldreich, J. Hastad, and R. Peralta, "Simple Constructions of Almost k -wise Independent Random Variables," Random Structures & Algorithms, vol. 3, no. 3, pp. 289-304, 1992.
- [9] R. A. Hearn and E. D. Demaine, Games, Puzzles, and Computation. AK Peters, 2009.
- [10] S. Alikhani and Y.-H. Peng, "Dominating Sets and Domination Polynomials of Paths," International Journal of Mathematics and Mathematical Sciences, vol. 2009, Article ID 542040, 10 pages.
- [11] S. Alikhani and Y.-H. Peng, "Introduction to Domination Polynomial of a Graph," Ars Combinatoria, vol. 114, pp. 257-266, 2014.
- [12] J. I. Brown and J. Tufts, "On the Roots of Domination Polynomials," Graphs and Combinatorics, vol. 30, no. 3, pp. 527-547, 2014.
- [13] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, Fundamentals of Domination in Graphs. Marcel Dekker, New York, 1998.
- [14] D. B. West, Introduction to Graph Theory, 2nd ed. Prentice Hall, 2001.
- [15] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996.