

Algebraic Construction of Post-Quantum Cryptographic Schemes over Semirings and Semimodules: Security and Computational Analysis

Joyce. Z¹, R. Stella Maragatham²

¹Research Scholar, Department of Mathematics, Queen Mary's College, Mylapore, Chennai, Tamil Nadu, 600004, India

²Associate professor, Department of Mathematics Queen Mary's College, Mylapore, Chennai, Tamil Nadu, 600004, India

Abstract

The advent of quantum computing necessitates the study of algebraic cryptographic schemes which would not rely explicitly on the problem of integer factorization or discrete logarithms. In this paper, we explore a framework for post-quantum cryptography with use of finite semirings, semimodules, and matrix actions with commutativity. A problem of recovering an action of a semimodule (SARP) is set up as the main computational problem, and cryptographic protocols of key exchange and hybrid public-key cryptosystems with algebraic correctness are constructed. The framework is studied in terms of its efficiency and security against attacks from the perspective of effective key spaces, collisions, orbit and stabilizer structures, algebraic recovery, exhaustive search, and generic quantum search algorithms. We derive the computational complexity of the operations in the main semiring and complement our theoretical results by an experimental implementation of those operations in Python on finite modular semiring. Our experiments of computations in semimodules of dimension $n \in \{16, 32, 64, 128, 256\}$ show the expected increase of cost of action of matrices on vectors and are consistent with the theoretical complexity of quadratic growth.

Keywords: Post-Quantum Cryptography, Semiring, Semimodule, Semigroup Action, SARP, Public-Key Encryption, Algebraic Cryptography.

1. Introduction

Public key cryptography plays a very important role in secure communications, secure authentication and establishment of keys. Popular cryptographic protocols like RSA, Diffie-Hellman, ECC have security guarantees based mainly on integer factorization or discrete logarithm problem. The emergence of quantum algorithms has changed the security scenario since it proved that both the integer factorization problem and discrete logarithm problem can be solved in polynomial time using a powerful enough quantum computer [1]. Semigroup and semiring actions are examples of such an algebraic approach. Maze et al. provided constructions of public key cryptography relying on semigroup actions, which show how action-based computational problems can extend the classical discrete-logarithm problems outside groups

[2]. Nivetha et al. then examined semiring actions in the context of public key cryptography, showing how to use them for key distribution and encryption [3]. This approach is appealing since semirings do not necessarily have additive inverses and hence allow for algebraic manipulations quite distinct from traditional group or field-based systems. Tropical algebra has extended this line even further. The earliest cryptographic schemes based on tropical algebra were constructed by Grigoriev and Shpilrain [4] and subsequently extended using homomorphisms [5]. Further work done in tropical circular matrices [6], double-key-exchange schemes [7], and protocols with lower circulant and anti-circulant matrices [8]. Other recent work in this area has included studies on specialized semiring-based structures and their possible cryptographic applications. These include the design of a semiring structure with applications to cryptography [9] and a semiring-based public key cryptography scheme that focuses on computational efficiency [10].

Nevertheless, presence of an unconventional algebraic structure does not necessarily mean security beyond quantum attacks. Security of an action-based construction is determined by the hardness of recovering private transformations from observed action values and the properties of the orbits, stabilizers and collision sets formed due to these transformations. As a result, crypto-systems over semiring structures need specific computational assumptions along with analysis of structural, algebraic, classical and quantum attacks. Driven by such a requirement, the current paper develops a cryptographic framework using the notions of finite semirings, semimodules, and commuting transformations. The main computational problem is stated within the Semimodule Action Recovery Problem (SARP) which allows to design a key exchange protocol and a public-key cryptography scheme which correctness holds under the commutativity of an admissible family of transformations. Also, the effective key space, collisions, properties of orbits and stabilizers, algebraic reconstruction, generic classical and quantum attacks, complexity, and memory usage are investigated in this paper. The primary contribution here is not the statement that all semiring constructions are quantum safe, but the development of a mathematical framework where one can investigate the security and computational features of semiring-semimodule cryptographic constructions. This difference is crucial since the security of the suggested framework strongly depends on the choice of the semiring and semimodule, family of transformations and hardness of the corresponding SARP problem.

2. Mathematical Preliminaries And Problem Formulation

This section introduces the algebraic concepts required for the proposed cryptographic construction.

Definition 2.1. Semiring

A nonempty set S with two binary operations $+$ and $\cdot$ is a semiring when $(S, +)$ is a commutative monoid, $(S, \cdot)$ is a monoid and multiplication distributes over addition

$$a(b + c) = ab + ac, (a + b)c = ac + bc, \quad (1)$$

for all $a, b, c \in S$. Unlike rings, additive inverses are not necessarily required.

Definition 2.2. Semimodule

Let S be a semiring. A left S -semimodule M is a commutative monoid $(M, +)$ together with scalar multiplication $S \times M \rightarrow M$ satisfying

$$a(x + y) = ax + ay, (a + b)x = ax + bx, \quad (2)$$

$$(ab)x = a(bx), 1_S x = x, \quad (3)$$

for $a, b \in S$ and $x, y \in M$.

Definition 2.3. Semigroup Action

For a semigroup G and a nonempty set X , an action is a mapping

$$\phi: G \times X \rightarrow X \quad (4)$$

satisfying

$$\phi(g_1, g_2 x) = \phi(g_1, \phi(g_2, x)). \quad (5)$$

For $x \in X$, its orbit and stabilizer are respectively

$$\text{Orb}(x) = \{\phi(g, x) : g \in G\}, \quad (6)$$

$$\text{Stab}(x) = \{g \in G : \phi(g, x) = x\}. \quad (7)$$

This is important to cryptography, since small orbits and large stabilizers considerably reduce the private action space.

Definition 2.4. Matrix Semiring Action

For a finite semiring S , the set $M_n(S)$ of $n \times n$ matrices form a matrix semiring. Taking the semimodule $M = S^n$, a matrix $A \in M_n(S)$ acts on $x \in M$ through

$$\phi_A(x) = Ax. \quad (8)$$

Let $\mathcal{F} \subseteq M_n(S)$ be a finite family of admissible transformations satisfying

$$AB = BA, \forall A, B \in \mathcal{F}. \quad (9)$$

The commutativity condition is central to the shared-secret construction developed subsequently.

Definition 2.5. Semimodule Action Recovery Problem

Given a public base vector $x \in M$ and

$$y = Ax, A \in \mathcal{F}, \quad (10)$$

the Semimodule Action Recovery Problem (SARP) is to recover A , or an equivalent admissible transformation A' satisfying

$$A'x = Ax. \quad (11)$$

SARP constitutes the principal computational recovery problem of the proposed framework.

The Decisional Semimodule Action Problem (DSAP) is such that an adversary must be able to distinguish between a correct action tuple and a sample action tuple taken from a given reference distribution. This distinction is useful in the analysis of key exchange and encryption schemes.

3. Proposed Semiring–Semimodule Cryptographic Framework

Let the public system parameters be

$$\mathcal{P} = (S, M, \mathcal{F}, x, H), \quad (12)$$

where S is a finite semiring, $M = S^n$ is an S -semimodule, $\mathcal{F}$ is a finite commuting family of admissible transformations, $x \in M$ is a public base vector, and H denotes a cryptographic hash or key-derivation function. For $A \in \mathcal{F}$, the corresponding public image is

$$P_A = Ax. \quad (13)$$

Consequently, public key generation involves the process of mapping $A \mapsto Ax$, while deriving an equivalent private transformation (x, P_A) is in line with the SARP concept.

3.1 Commuting Transformation Family

We can create a convenient class of transformations using a public matrix $G \in M_n(S)$. Given a coefficient vector

$$\alpha = (\alpha_0, \alpha_1, \dots, \alpha_d), \quad (14)$$

Define

$$A_\alpha = \alpha_0 I \oplus \alpha_1 G \oplus \alpha_2 G^2 \oplus \dots \oplus \alpha_d G^d. \quad (15)$$

Since transformation functions constructed from polynomials using the same matrix commute based on the semiring operations assumed,

$$A_\alpha A_\beta = A_\beta A_\alpha. \quad (16)$$

Small coefficient vectors thus become private descriptions. Nevertheless, cryptographic appropriateness implies that it be intractable to derive from the public operation its corresponding description or transformation.

3.2 Key Establishment

3.3

Alice independently selects $A \in \mathcal{F}$, while Bob selects $B \in \mathcal{F}$. Their public values are

$$P_A = Ax, P_B = Bx. \quad (17)$$

After exchanging these values, Alice computes

$$Z_A = AP_B = A(Bx), \quad (18)$$

whereas Bob computes

$$Z_B = BP_A = B(Ax). \quad (19)$$

Because $AB = BA$,

$$Z_A = (AB)x = (BA)x = Z_B. \quad (20)$$

Both parties consequently obtain the common secret

$$Z = (AB)x, \quad (21)$$

and derive the session key as

$$K = H(Z \parallel P_A \parallel P_B). \quad (22)$$

Theorem 1 (Correctness).

For any $A, B \in \mathcal{F}$ satisfying $AB = BA$, the two communicating parties derive the same session key.

Proof. From associativity of the matrix action and commutativity of A and B ,

$$A(Bx) = (AB)x = (BA)x = B(Ax). \quad (23)$$

Hence $Z_A = Z_B$. Application of the same deterministic key-derivation function to identical inputs produces the same session key K . Current realizations of Stickel's key-exchange problem over max-min and max- T semirings show the importance of the proper choice of semiring in terms of both the implementation and the cryptanalytic behavior of the scheme [11]. Analogously, block-matrix schemes in the context of tropical algebra show how the structure of a matrix can be used when designing key exchange [12]. The suggested construction thus provides the algebraic requirement needed for generating a shared secret separately from the security requirement: correctness follows from (22), but security is dependent on the hardness of the SARP problem for a particular S, M, F , and the associated parameters.

4. Public-Key Encryption Construction

The shared secret approach in Section 3 can be further generalized into a public key-based hybrid scheme. The approach uses the semiring-semimodule algebraic operation and the symmetric encryption to obtain the shared secret and use the latter to derive the symmetric encryption key.

4.1 Encryption Procedure

Let the receiver's private transformation be $A \in \mathcal{F}$, with corresponding public key

$$P_A = Ax. \quad (24)$$

For a plaintext m , the sender independently selects a random ephemeral transformation $R \in \mathcal{F}$ and computes the corresponding ephemeral public value

$$P_R = Rx. \quad (25)$$

Using the receiver's public value P_A , the sender computes the shared action value

$$Z = RP_A = R(Ax). \quad (26)$$

A symmetric session key is then derived from the shared value using the key-derivation function H :

$$K = H(Z). \quad (27)$$

The plaintext is protected using an authenticated-encryption algorithm under the derived key:

$$c = \text{AEAD.Enc}_K(m), \quad (28)$$

where c denotes the authenticated ciphertext. The complete ciphertext transmitted to the receiver is therefore

$$C = (P_R, c). \quad (29)$$

The transient public value P_R enables the recipient to regenerate the same shared value A through the private value A , while the message is protected by authenticated symmetric encryption.

4.2 Decryption and Correctness

After receiving the ciphertext $C = (P_R, c)$, the receiver applies the private transformation A to the ephemeral public value and computes

$$Z' = AP_R = A(Rx). \quad (30)$$

Since $A, R \in \mathcal{F}$ and the admissible transformation family is commuting,

$$AR = RA. \quad (31)$$

Consequently,

$$Z' = A(Rx) = (AR)x = (RA)x = R(Ax) = Z. \quad (32)$$

The receiver therefore derives the same symmetric key $K = H(Z') = H(Z)$ and recovers the plaintext as

$$m = \text{AEAD.Dec}_K(c). \quad (33)$$

Theorem 2 (Functional Correctness).

For honest key and ciphertext generation, the proposed encryption scheme correctly reconstructs the plaintext.

Proof. From (29)–(31), the commutativity of the admissible transformations implies

$$Z' = A(Rx) = R(Ax) = Z. \quad (34)$$

Since both parties apply the same deterministic key-derivation function to identical shared values,

$$H(Z') = H(Z) = K. \quad (35)$$

Correctness of the authenticated-encryption algorithm then gives

$$\text{AEAD.Dec}_K(\text{AEAD.Enc}_K(m)) = m, \quad (36)$$

which establishes functional correctness.

It is critical to differentiate between functional correctness and formal ciphertext security. Recent research into action-based cryptography proves that more advanced security models like CCA security require decisional assumptions and corresponding reductions in security [13]. Hence, Theorem 2 proves the functional correctness of the presented scheme; however, it cannot be seen as an IND-CCA security proof by itself.

5. Security And Computational Analysis

The security of the suggested construction relies on the difficulty of semimodule action chosen rather than the structure of semirings only. It means that the efficient action space, collision properties, algebraic representation, and the ability to withstand classical and quantum attacks should be analyzed together.

5.1 SARP, Collisions, and Effective Key Space

For the public action map

$$\psi_x: \mathcal{F} \rightarrow M, \psi_x(A) = Ax, \quad (37)$$

two distinct transformations collide whenever

$$A \neq B, Ax = Bx. \quad (38)$$

Define the maximum collision multiplicity as

$$\mu_x = \max_{y \in M} |\{A \in \mathcal{F}: Ax = y\}|. \quad (39)$$

A conservative lower bound on the effective number of distinguishable private actions is therefore

$$N_{\text{eff}} \geq \frac{|\mathcal{F}|}{\mu_x}. \quad (40)$$

In other words, a big nominal key space does not guarantee a big effective one. It may be reduced because of small orbits, big stabilizers, or collisions. Linear algebraic structures on semirings which are additively idempotent can have some properties that are useful for revealing transformations from their public algebraic expressions [14]. Similarly, studying tropical circulant protocols allows concluding that special families of matrices need special cryptanalysis rather than an argumentation about the size of the nominal key space [15].

5.2 Algebraic and Structural Attacks

For a polynomial transformation

$$A_\alpha = \bigoplus_{i=0}^d \alpha_i G^i, \quad (41)$$

the public relation

$$P_A = A_\alpha x = \bigoplus_{i=0}^d \alpha_i (G^i x) \quad (42)$$

may reveal formulas involving the private coefficients α_i . Consequently, an attacker may try to recover either the coefficients, equivalent transformation, or perform decomposition, linearization, or orbit enumeration. Such problem becomes especially relevant in the field of tropical cryptography. The study of the security of the tropical two-sided discrete logarithm problem reveals that the security of such protocols is strongly related to the algebraic problem and its representation [16]. Previous analysis of tropical key-exchange protocols revealed that large parameter spaces do not necessarily protect from structural attacks [17].

5.3 Classical and Quantum Search

If no structural attack performs better than exhaustive search, classical recovery over the effective action space requires approximately

$$T_{cl} = O(N_{eff}) \quad (43)$$

action evaluations. Quantum amplitude amplification can reduce generic unstructured search to approximately

$$T_q = O(\sqrt{N_{eff}}). \quad (44)$$

Therefore, targeting λ -bit resistance against generic quantum search suggests the conservative sizing condition

$$N_{eff} \geq 2^{2\lambda}. \quad (45)$$

This is merely a requirement for generic search, not a demonstration of post-quantum security. The significance of this is highlighted even more by the quantum algorithms for semidirect discrete logarithm problems related to the construction in question, where quantum attacks subexponential in difficulty can take advantage of the algebraic structure above and beyond generic search [18]. Alternative cryptographic group actions have also been researched, showing how important the particular action itself is to security [19], as well as QROM analysis of action-based constructions [20].

5.4 Computational Complexity

Let C_+ and $C_\times$ denote the costs of one semiring addition and multiplication. For $A \in M_n(S)$ and $x \in S^n$, a matrix–vector action requires n^2 semiring multiplications and $n(n-1)$ semiring additions. Hence,

$$T_{act} = n^2 C_\times + n(n-1) C_+ = O(n^2). \quad (46)$$

On the other hand, traditional matrix multiplication of two $n \times n$ matrices using the semiring takes $O(n^3)$ number of semiring operations. If each semiring operation takes b bits, then the space requirements for dense matrices and vectors are

$$S_M = n^2 b, \quad S_V = nb. \quad (47)$$

Table 1. Computational Complexity of Principal Operations

Operation	Multiplications	Additions	Complexity
Matrix–vector action	n^2	$n(n-1)$	$O(n^2)$
Shared-secret action	n^2	$n(n-1)$	$O(n^2)$
Dense matrix multiplication	n^3	$n^2(n-1)$	$O(n^3)$

Degree- d polynomial action	$\approx dn^2$	$\approx dn(n-1)$	$O(dn^2)$
-------------------------------	----------------	-------------------	-----------

Source: Authors' derivation based on the computational complexity analysis of the proposed framework. As seen in Table 1, direct matrix-vector evaluation is the better approach to take where the process only requires actions of transformations and not actual multiplications of the matrices. However, these asymptotic results indicate scalability but are not supposed to be considered as cryptographic hardness.

6. Numerical And Computational Results

This proposal was analyzed using three different approaches which included algebraic soundness, computational analysis of complexity, and scalability at the implementation level. The numerical analysis is meant to verify the computational behavior of the proposal and not to prove the cryptographic strength of the SARP parameters.

6.1 Numerical Verification

Let $A, B \in \mathcal{F}$ be admissible commuting transformations and $x \in M$ the public base vector. The corresponding public values are

$$P_A = Ax, P_B = Bx. \quad (48)$$

The two parties independently calculate

$$Z_A = A(Bx), Z_B = B(Ax). \quad (49)$$

Since $AB = BA$,

$$Z_A = (AB)x = (BA)x = Z_B. \quad (50)$$

Equation (46) confirms the basic condition of correctness that is essential to the key establishment procedure. But correctness does not mean security by itself. In case two different transforms $A \neq A'$ satisfy

$$Ax = A'x, \quad (51)$$

They create the same public image and hence are part of the same effective action class. The choice of cryptographic parameters should not be based solely on the cardinality of F , but should also take into account distinguishable action classes and collisions.

6.2 Analytical Scalability

For an n -dimensional semimodule, a dense matrix-vector action requires

$$N_{\times} = n^2, N_{+} = n(n-1), \quad (52)$$

for multiplication and addition operations in semirings, respectively. Considering 8 bits representation for elements of the semiring, the analytical requirements are shown in Table 2.

Table 2. Analytical Scalability of the Proposed Construction

n	Multiplications/Action	Additions/Action	Dense Matrix Storage
16	256	240	256 B
32	1,024	992	1,024 B
64	4,096	4,032	4,096 B
128	16,384	16,256	16,384 B
256	65,536	65,280	65,536 B

Source: Authors' calculations based on the derived computational complexity.

This operation cost grows as the square of n . If the semimodule is increased in dimension from 16 to 256, then the number of multiplications grow from 256 to 65,536, and the number of bytes from 256 B to 65,536 B, assuming the one-byte representation. The obtained numbers match the complexity $O(n^2)$ of direct matrix-vector computations and are still lower asymptotically than direct dense matrix multiplications, that need $O(n^3)$ operations.

6.3 Experimental Computational Evaluation

To complement the analytical complexity assessment, a Python-based computational implementation was developed to examine the computational behaviour of the principal semiring matrix–vector action. The experiment was designed to evaluate implementation-level scalability and to determine whether the observed computational trend is consistent with the analytical operation count derived in Section 6.2. It was not intended to establish a cryptographically secure parameter set.

For benchmarking purposes, computations were performed over the finite modular semiring

$$S = \mathbb{Z}_{251}, \quad (53)$$

with the corresponding semimodule

$$M = S^n, n \in \{16, 32, 64, 128, 256\}. \quad (54)$$

For a matrix $A \in S^{n \times n}$ and a vector $x \in S^n$, the implemented action was

$$y = Ax \pmod{251}. \quad (55)$$

Pseudorandom seeds were utilized to generate random matrices and vectors to allow for reproducibility. Warm-up was performed before timing measurements. Each operation involving matrix-vector product was then measured for 30 runs. The average time of execution was computed for each semimodule dimension. Wall-clock measurements depend on implementation details and the environment used, which makes us consider our timings as an observation of scalability of the implementation rather than some universal performance metrics. The selection of $\mathbb{Z}_{251}$ can be seen as a concrete computational experiment. Indeed, since $\mathbb{Z}_{251}$ is a field and thus also satisfies the axioms of semirings, this semiring can be used to check the implementation and scaling of the action under consideration. Nevertheless, the usage of $\mathbb{Z}_{251}$ does not imply that this semiring forms a cryptographically safe SARP or post-quantum parameter set.

Table 3. Experimental Computational Performance

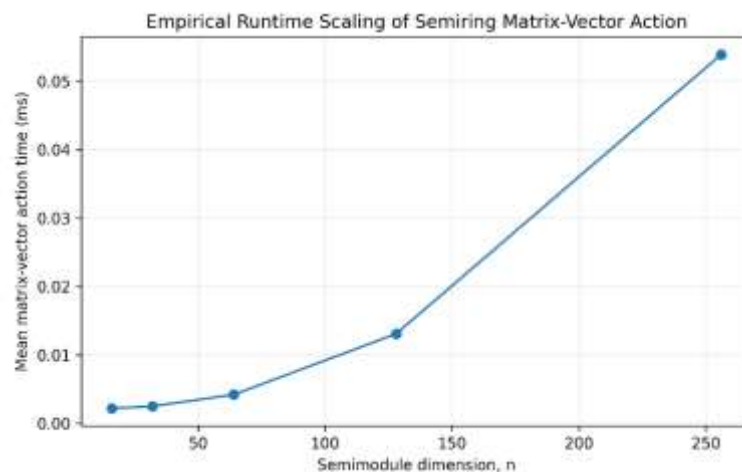
n	Multiplications	Additions	Mean Action Time (ms)	Matrix Storage*
16	256	240	0.0020	256 B
32	1,024	992	0.0024	1,024 B
64	4,096	4,032	0.0042	4,096 B
128	16,384	16,256	0.0130	16,384 B
256	65,536	65,280	0.0539	65,536 B

*Theoretical packed storage assuming one byte per semiring element; this value does not represent the in-memory storage occupied by the Python/NumPy implementation.

Source: Authors' computation using the Python implementation.

As can be seen from Table 3, the average execution time rises from $0.0020ms$ when $n = 16$ up to $0.0539ms$ when $n = 256$. It is becoming even more evident at higher dimensions, since it reflects the quadratic dependence on the number of elementary operations necessary for dense matrix-vector multiplication. When n increases from 16 to 256, then the number of semiring multiplications rises from 256 to 65,536 while the number of additions rises from 240 to 65,280. Thus, the empirical timings support the theoretical $O(n^2)$ estimation for the number of operations obtained in Section 6.2, although the asymptotic complexity arises out of the mathematical operation count.

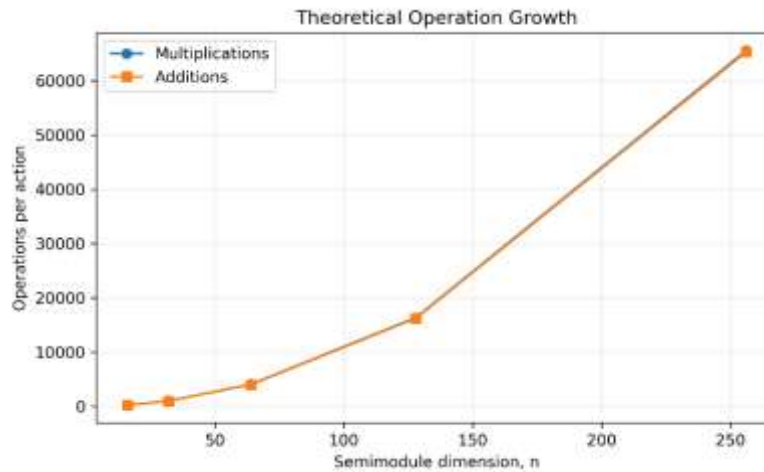
Figure 1. Empirical Runtime Scaling of the Semiring Matrix–Vector Action



Source: Authors' computation and visualization using Python.

The gradual rise of the time needed to compute matrix-vector products depending on the semimodule size is shown in Figure 1. This phenomenon especially comes out clear for $n = 128$ and $n = 256$. Such behavior correlates with the theoretically obtained quadratic growth in operations. However, the time values depend on the computation platform and thus should be considered relatively.

Figure 2. Theoretical Growth of Semiring Operations



Source: Authors' calculations and visualization based on the derived operation counts.

Figure 2 represents the theoretical scaling of semiring multiplications and additions needed for a dense matrix-vector multiplication. Both these quantities scale with the square of n and stay coordinated with each other as the semimodule dimension grows. For instance, at $n=256$, one multiplication operation takes 65,536 semiring multiplications and 65,280 semiring additions. The correlation of the operation scaling with the runtime scaling demonstrates the practical basis of the scalability analysis.

6.4 Comparative Security and Computational Discussion

The main difference between traditional public key systems and the one being suggested is that of the computational problem. Unlike RSA, Diffie-Hellman, and Elliptic Curve Cryptography that are based on factoring or traditional discrete logarithm problems, the current system is built on recovery and decisional problems related to semimodule actions.

Table 4. Structural Comparison of Cryptographic Approaches

Approach	Algebraic Structure	Principal Hardness Basis	Main Security Concern
RSA	Modular arithmetic	Integer factorization	Quantum factorization
Diffie–Hellman	Finite cyclic group	DLP/CDH	Quantum discrete logarithm
ECC	Elliptic-curve group	ECDLP	Quantum discrete logarithm
Tropical schemes	Tropical semiring/matrices	Structure-specific problems	Algebraic/structural attacks
Proposed framework	Semiring–semimodule action	SARP/DSAP	Parameter- and structure-dependent hardness

Source: Authors' synthesis based on the literature reviewed in this study.

The presented design does not rely explicitly on the factorization or discrete logarithm assumption but maintains computationally efficient matrix-vector product. The obtained analytical and empirical results show the reasonable scalability of computations for the considered dimension range. However, the computational efficiency and post-quantum security are two different features. To construct a secure

SARP, it should additionally exhibit sufficiently large action classes, controllable number of collisions, and protection against algebraic, structural, classical, and quantum attacks. Thus, the results provide the proof of algebraic correctness, analytical complexity, and computational viability at the implementation level, but not of cryptographically hard post-quantum parameters set. This feature precludes computational performance benchmarks from being viewed as a cryptographic hardness indicator.

7. Conclusion

This paper presented a post-quantum cryptography scheme built upon finite semirings, semimodules and commuting actions of matrices. The Semimodule Action Recovery Problem (SARP) and decisional versions of the problem were proposed as the core computational problems; key establishment and hybrid public-key encryption schemes were built with functionally correct algebraically constructions. The security analysis showed that the security level of a particular construction is determined by effective action space size, number of collisions, behavior of orbits and stabilizers, immunity to algebraic recovery and impossibility of recovery algorithms for classical and quantum computers. The computational analysis revealed that the complexity of performing matrix-vector actions is $O(n^2)$, while the complexity of explicit multiplication of dense matrices is $O(n^3)$. The experimental analysis conducted at implementation level showed that the time of an action increases with respect to dimension $n = \{16, 32, 64, 128, 256\}$. The findings thus provide a basis for exploring SARP as an option for post-quantum cryptographic construction. However, it is important to note that the modular experimentally defined semiring should not be considered a parameter suite tested for cryptographic security, while the construction itself is not claimed to be quantum secure. It would be necessary in the future to define specific difficult SARP instances, develop security reductions, conduct classical and quantum cryptanalysis, and validate implementations in comparison with post-quantum cryptography schemes.

References

1. Shor, P. W., "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM Review, 1999, 41 (2), 303–332.
2. Maze, G., Monico, C., Rosenthal, J., "Public key cryptography based on semigroup actions," Advances in Mathematics of Communications, 2007, 1 (4), 489–507.
3. Nivetha, S., Thiruvani, V., Chandramouleeswaran, M., "Semiring actions for public key cryptography," Journal of Computer and Mathematical Sciences, 2019, 10 (1), 238–244.
4. Grigoriev, D., Shpilrain, V., "Tropical cryptography," Communications in Algebra, 2014, 42 (6), 2624–2632.
5. Grigoriev, D., Shpilrain, V., "Tropical cryptography II: Extensions by homomorphisms," Communications in Algebra, 2019, 47 (10), 4224–4229.
6. Huang, H., Li, C., Deng, L., "Public-key cryptography based on tropical circular matrices," Applied Sciences, 2022, 12 (15), 7401.
7. Durcheva, M. I., "TrES: Tropical encryption scheme based on double key exchange," European Journal of Information Technologies and Computer Science, 2022, 2 (4), 11–17.
8. Amutha, B., Perumal, R., "Public key exchange protocols based on tropical lower circulant and anti circulant matrices," AIMS Mathematics, 2023, 8 (7), 17307–17334.
9. Huang, H., Jiang, X., Peng, C., Pan, G., "A new semiring and its cryptographic applications," AIMS Mathematics, 2024, 9 (8), 20677–20691.

10. Nassr, D. I., Bahig, H. M., Hazber, M. A., Alseadoon, I. M., Bahig, H. M., “A fast semiring-based public-key encryption,” *AIMS Mathematics*, 2025, 10 (4), 8569–8586.
11. Alhussaini, S. Sergeev, S., “On implementation of Stickel’s key exchange protocol over max-min and max-T semirings,” *Journal of Mathematical Cryptology*, 2024, 18 (1), 20240014.
12. Durcheva, M., Danilchenko, K., “Secure key exchange in tropical cryptography: Leveraging efficiency with advanced block matrix protocols,” *Mathematics*, 2024, 12 (10), 1429.
13. Meers, J. Riepel, D. “CCA secure updatable encryption from non-mappable group actions,” in *Post-Quantum Cryptography: 15th International Workshop, PQCrypto 2024, LNCS*, 2024, 14771, 137–169.
14. Otero Sánchez, Á., Camazón Portela, D., López-Ramos, J. A., “On the solutions of linear systems over additively idempotent semirings,” *Mathematics*, 2024, 12 (18), 2904.
15. Buchinskiy, I., Kotov, M., Treier, A., “Analysis of four protocols based on tropical circulant matrices,” *Indian Journal of Pure and Applied Mathematics*, 2024.
16. Alhussaini, S., Collett, C., Sergeev, S., “On the tropical two-sided discrete logarithm and a key exchange protocol based on the tropical algebra of pairs,” *Communications in Algebra*, 2024, 52 (10), 4115–4134.
17. Rudy, D., Monico, C., “Remarks on a tropical key exchange system,” *Journal of Mathematical Cryptology*, 2021, 15 (1), 280–283.
18. Battarbee, C., Kahrobaei, D., Perret, L., Shahandashti, S. F., “A subexponential quantum algorithm for the semidirect discrete logarithm problem,” in *Post-Quantum Cryptography: 15th International Workshop, PQCrypto 2024, LNCS*, 2024, 14771, 202–226.
19. Benčina, B., Budroni, A., Chi-Domínguez, J.-J., Kulkarni, M., “Properties of lattice isomorphism as a cryptographic group action,” in *Post-Quantum Cryptography: 15th International Workshop, PQCrypto 2024, LNCS*, 2024, 14771, 170–201.
20. Bläser, M., Chen, Z., Duong, D. H., Joux, A., Nguyen, T., Plantard, T., Qiao, Y., Susilo, W., Tang, G., “On digital signatures based on group actions: QROM security and ring signatures,” in *Post-Quantum Cryptography: 15th International Workshop, PQCrypto 2024, LNCS*, 2024, 14771, 227–261.