WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 8, Number 3, Year 2026
DOI: https://doi.org/10.62051/ijcsit.v8n3.08
Journal homepage: https://wepub.org/index.php/IJCSIT/index

A Computational-Space-Oriented Reconstruction of Abstract Algebra Teaching in Foundations of Information Security Mathematics

Wei Gao, Mei Li

Department of Computer Science, North China Electric Power University(Baoding), Baoding, China

ABSTRACT

Foundations of Information Security Mathematics supports subsequent cryptography courses, yet teaching abstract algebra as an isolated theory often prevents students from relating algebraic structures to cryptographic mechanisms. This paper interprets algebra as the computational spaces of cryptography and reorganizes the course through a requirement-driven structure, where cyclic groups, finite fields, and quotient polynomial rings are introduced as progressively extended environments. A computational-space-oriented pathway is implemented by introducing concepts through cryptographic operations and reinterpreting prior discrete mathematics knowledge. Classroom practice shows a shift from procedural to structural understanding and a unified view of different cryptographic schemes, turning the course from a set of prerequisites into the structural foundation for later study.

KEYWORDS

Abstract Algebra; Algebraic structures; Computational space; Information security mathematics; Requirement-driven teaching

1. INTRODUCTION

1.1. Research Background

Foundations of Information Security Mathematics is a core compulsory course for undergraduates majoring in Information Security and Cyberspace Security. It provides the mathematical basis for understanding and designing cryptographic systems and is typically composed of two parts: elementary number theory and abstract algebra, the latter focusing on groups, rings, and fields.

Abstract algebra forms the structural language of cryptography. Classical schemes such as RSA and Diffie-Hellman (DH) are based on group-theoretic settings, elliptic curve cryptography (ECC) is defined over finite fields, and recent developments including lattice-based cryptography and homomorphic encryption rely on polynomial rings and their quotient structures [1]. In this sense, algebra is not only a mathematical tool but also the computational environment in which cryptographic mechanisms are defined and executed.

Before taking this course, students have encountered groups, rings, and fields in discrete mathematics. However, these topics are usually presented at a formal level without application context. After a time gap of about one year, most students retain only fragmentary knowledge. The course therefore becomes both a process of knowledge reconstruction and a transition from abstract structures to application-oriented mathematics, which is consistent with the theory that prior knowledge must be reorganized in a new conceptual framework for effective learning [2].

1.2. Current Teaching Challenges

Despite its importance, several difficulties remain in the teaching of abstract algebra in this course.

(1) The level of abstraction is high. Algebraic structures lack intuitive interpretations, and students often fail to see their operational meaning.

(2) The available teaching hours are limited. Within 40–50 hours, number theory and abstract algebra must be covered, which makes content selection a central issue.

(3) Theory and application are often disconnected. Traditional teaching emphasizes definitions and proofs, while the role of algebra in concrete cryptographic schemes is insufficiently highlighted.

(4) Recent cryptographic developments are not adequately reflected. Polynomial rings, which are fundamental to lattice-based cryptography and homomorphic encryption [3], are usually treated as marginal topics and are rarely presented as unified computational environments.

1.3. Research Significance and Objectives

To address these problems, this paper interprets abstract algebra as the computational spaces of cryptographic systems and reconstructs the course accordingly. A requirement-driven organization of content is proposed, in which cyclic groups, finite fields, and quotient polynomial rings are introduced as progressively extended environments. On this basis, a computational-space-oriented teaching pathway is designed and implemented.

The aim is to transform algebra from a collection of formal definitions into a structural framework for understanding cryptographic computation, to support the transition from discrete mathematics to professional cryptography courses, and to provide a coherent curriculum model under limited teaching hours.

2. ALGEBRA AS THE CRYPTOGRAPHIC WORKSPACE: AN APPLICATION-ORIENTED PERSPECTIVE

2.1. Algebraic Structures as Computational Spaces

Cryptographic algorithms are not only collections of symbolic operations but computations carried out in specific algebraic environments. The correctness, efficiency, and security of a scheme are determined by the structural properties of the space in which these operations are defined [1].

In classical public-key cryptography, modular exponentiation is performed in cyclic groups, and the hardness of the discrete logarithm problem (DLP) depends on the group structure. In ECC, point operations are defined over finite fields [4], and their algebraic properties ensure both correctness and security. In symmetric cryptography, the operations of AES are realized in $\mathrm{GF}(2^8)$ [5], where invertibility and closure are guaranteed by field construction. In more recent schemes, such as lattice-based and homomorphic encryption, encryption and evaluation are carried out in quotient polynomial rings [6].

From this perspective, algebraic structures are the computational spaces of cryptographic systems. Definitions and theorems describe the properties of these spaces, and cryptographic algorithms can be interpreted as structured computations within them.

2.2. From Modular Arithmetic to Abstract Algebraic Cryptographic Spaces

The algebraic environments used in cryptography form a natural progression. Modular integer arithmetic provides the initial setting for number-theoretic computation. Cyclic groups extend this

setting to support discrete-logarithm-based (DLP-based) protocols. Finite fields offer a closed and invertible environment for symmetric and ECC. Quotient polynomial rings further generalize these structures and serve as the computational space for lattice-based and homomorphic schemes.

This progression is not only a mathematical classification but also a path of increasing computational capability. Each new structure preserves the operational features of the previous one while enabling new cryptographic mechanisms. As a result, modern cryptographic systems can be interpreted as computations carried out in progressively extended algebraic spaces rather than as applications of isolated mathematical topics.

2.3. Implications for Teaching Abstract Algebra in Information Security

If algebraic structures are viewed as computational spaces, the focus of teaching shifts from formal completeness to structural relevance. The key question is no longer how to present an axiomatic system in full generality, but how each structure supports a class of cryptographic mechanisms.

This perspective provides a criterion for content organization under limited teaching hours. Topics that directly contribute to the understanding of cryptographic computation become central, while highly abstract structural classifications can be correspondingly reduced. It also explains why polynomial rings should not be treated as marginal topics, since they form the computational environment for major contemporary cryptographic schemes.

More importantly, this interpretation establishes a conceptual bridge between discrete mathematics and professional cryptography courses. Algebra is no longer learned as an isolated theoretical component but as the structural model underlying cryptographic computation. This theoretical perspective forms the basis for the course reconstruction described in the following sections.

3. KNOWLEDGE RECONSTRUCTION: WHAT SHOULD BE TAUGHT

3.1. From Mathematical System to Cryptographic Requirement

In traditional abstract algebra courses, the content is usually organized according to the internal logical structure of mathematics. Topics are introduced in a linear sequence—groups, subgroups, quotient groups, homomorphisms, rings, ideals, fields, and field extensions—with the primary goal of establishing a complete and rigorous theoretical system. While such an arrangement is appropriate for mathematics majors, it does not fully meet the needs of students in information security, whose main concern is not the structural completeness of algebra but its role in cryptographic constructions.

From the perspective established in the previous section, abstract algebra in this course should be understood as the description of cryptographic workspaces. Cryptographic algorithms do not operate on abstract symbols in isolation; they are carried out within specific algebraic environments. The properties of these environments—such as closure, invertibility, element order, and quotient structure—determine the correctness, efficiency, and security of the algorithms. Therefore, the question of "what should be taught" is essentially the question of "which algebraic knowledge is necessary for understanding these computational spaces".

This leads to a shift from a structure-oriented curriculum to a requirement-driven reconstruction. Instead of introducing algebraic concepts according to their position in the mathematical hierarchy, the course content is reorganized by tracing back from cryptographic systems to the algebraic structures that support them. In this framework, cryptographic schemes serve as cognitive anchors, and the corresponding algebraic notions are introduced as tools for explaining how and why these schemes work. More specifically, the required knowledge can be analyzed at three interrelated levels.

The first level is the understanding of the computational space. Students need to know in which mathematical domain a cryptographic algorithm operates. For DLP-based cryptosystems, this space

is a cyclic group; for AES and ECC, it is a finite field; for lattice-based cryptography and homomorphic encryption, it is typically a quotient polynomial ring. Recognizing the computational space allows students to see that the choice of algebraic structure is not arbitrary but is directly related to both functionality and security.

The second level is the understanding of structural properties that ensure the correctness of the algorithm. Concepts such as the order of an element, the existence of multiplicative inverses, and the formation of quotient structures explain why exponent reduction works in RSA, why inversion is possible in AES, and why polynomial reduction defines a closed and efficient computational environment in lattice-based schemes. At this level, algebra is no longer a collection of definitions but a mechanism that guarantees the validity of cryptographic operations.

The third level is the understanding of the origin of security. The hardness of the DLP is a consequence of the structure of a finite cyclic group; the security of lattice-based cryptography is rooted in the geometry of high-dimensional modules over polynomial rings. By relating security assumptions to algebraic structures, students can move beyond a formula-based view of cryptography and develop a structural understanding of why these systems are believed to be secure [7].

Under this requirement-driven framework, algebraic topics are no longer taught as isolated theoretical components. Instead, they are introduced in response to concrete explanatory needs. For example, the notion of a generator arises naturally when discussing the feasibility of DH key exchange; the construction of finite fields becomes necessary when explaining the byte-level operations in AES; quotient polynomial rings are motivated by the need for efficient convolution in lattice-based cryptography. In this way, the learning sequence follows a path from cryptographic phenomena to algebraic abstraction, rather than the reverse.

Such a reconstruction also provides a natural way to reactivate students' prior knowledge from discrete mathematics. Concepts that were previously learned as formal definitions are revisited in a new context where their computational meaning becomes explicit. This process not only reinforces existing knowledge but also transforms it into a usable cognitive tool.

In summary, the transition from a mathematical system–oriented curriculum to a cryptographic requirement–driven curriculum is the key step in redefining the role of abstract algebra in Foundations of Information Security Mathematics. It establishes a clear principle for content selection and organization, and it enables students to understand algebra as the structural foundation of cryptographic computation rather than as an independent theoretical discipline.

3.2. Algebra for DLP-based Cryptosystems

Based on the requirement-driven framework, group theory is introduced through DLP-based cryptosystems, such as DH, ElGamal, and ECC. The objective is not to present a complete theory of groups, but to identify the concepts needed to explain how these schemes work and why they are secure.

The starting point is the computational space. Students need to recognize that these cryptographic algorithms are performed in a cyclic group rather than in the set of integers itself. The notion of a generator is introduced in this context: it explains how all elements used in the protocol are produced and why public parameters can be represented in exponential form. This shifts the focus from symbolic manipulation to the structure of the underlying space.

The next step is to understand the correctness of the algorithm. The concept of the order of an element provides the explanation for exponent reduction and for the periodicity of group operations. At an intuitive level, Lagrange's theorem clarifies why certain exponentiations return to the identity element and why key agreement in DH produces the same result on both sides. In this way, the correctness of the protocol is seen as a direct consequence of the group structure.

More importantly, group theory provides the structural source of security. The hardness of the DLP is not a property of a particular formula but a feature of the cyclic group in which the operation is performed. By emphasizing the relationship between the group structure, the existence of generators, and the infeasibility of reversing exponentiation, students can understand why these systems are believed to be secure.

Under this organization, the teaching focus is placed on cyclic groups, generators, and element order, while highly abstract classifications are not taken as primary objectives. With this foundation, ECC can be introduced as a change of computational space that provides higher efficiency while preserving the same group-based security principle.

In this way, group theory is learned through three closely related questions: where the computation takes place, why the protocol works, and why it is secure.

3.3. Algebra for Finite-Field-Based Cryptography

Finite fields form the main computational environment for block ciphers and ECC. In this course, field theory is therefore introduced through the need to explain concrete cryptographic operations rather than as an abstract axiomatic system.

The key idea is a closed and invertible computational space. Cryptographic algorithms require addition, multiplication, and inversion to be consistently defined, which naturally leads to the concept of a field. Students thus understand a field as the environment in which encryption and decryption are actually performed.

Polynomial representation becomes essential when analyzing AES. Each byte is interpreted as an element of $GF(2^8)$, and both the S-box and the MixColumns transformation are realized through polynomial operations modulo an irreducible polynomial. In this way, field construction becomes a method for implementing efficient and reversible computation.

Field extensions are further introduced through ECC. The coordinates of curve points are elements of a finite field, and the group law is implemented through field operations. Once students recognize that elliptic curve operations are carried out over a finite field, the transition from modular arithmetic to elliptic curves becomes a change of computational space rather than a new theoretical system.

Under this organization, the teaching focus is placed on multiplicative inverses, polynomial representation, and field operations as the basis of higher-level cryptographic structures. AES and ECC are thus interpreted within the same algebraic framework: both are algorithms defined over carefully constructed finite fields.

In summary, field theory is taught as the algebraic foundation for reliable and efficient computation in cryptography, enabling students to understand where the operations take place and why such a structure is necessary.

3.4. Algebra for Polynomial-Ring-Based Cryptography

Polynomial rings are introduced as the computational space for modern cryptographic schemes beyond classical number-theoretic settings. They underpin both lattice-based cryptography and homomorphic encryption and therefore extend the course toward contemporary applications.

The starting point is to view polynomial rings as a natural generalization of modular integer arithmetic. Just as modular reduction defines a closed space for integer computation, reduction modulo a polynomial defines a closed space for polynomial operations. This allows students to connect earlier knowledge of congruence with a new algebraic environment.

Operations are performed in quotient rings of the form $R_q = Z_q[x] / f(x)$, which serve as the space for encryption, decryption, and key generation in schemes such as NTRU and Ring-LWE–based

constructions. The algebraic structure is thus the environment in which the algorithm operates. This setting also explains efficiency. Polynomial multiplication becomes convolution and can be accelerated by NTT-based algorithms, showing how algebraic design determines computational performance.

For homomorphic encryption, the same quotient ring functions as the ciphertext space. The homomorphic property is interpreted as the preservation of addition and multiplication under encryption, while the Chinese Remainder Theorem over polynomial rings provides the mechanism for ciphertext packing and parallel computation.

Under this organization, the teaching focus is placed on quotient structures, polynomial multiplication, and algebraic decomposition, rather than on abstract ring classifications. This creates a continuous path from modular integers to finite fields and further to lattice-based and homomorphic cryptography.

In summary, polynomial rings are presented as the algebraic foundation for efficient and advanced cryptographic computation, linking classical techniques with current developments in both post-quantum security and privacy-preserving computation.

3.5. Reactivating Prior Knowledge and Forming a Cryptography-Oriented Knowledge Map

An important role of this course is to reactivate students' prior exposure to groups, rings, and fields in discrete mathematics and turn it into a usable framework for understanding cryptography. Previously learned concepts are revisited in a concrete computational context, so that algebraic structures are no longer seen as formal definitions but as the spaces in which cryptographic algorithms operate.

This leads to a layered knowledge map for cryptography-oriented learning. The first layer consists of computational spaces, including cyclic groups, finite fields, and quotient polynomial rings. The second layer contains the structural properties that ensure correct and efficient operations, such as element order, multiplicative inverses, and quotient structures. The third layer corresponds to cryptographic instantiations, including DLP-based schemes, AES and ECC, and lattice-based and homomorphic encryption. Through this organization, individual topics are connected within a coherent framework. This framework is visualized in Table 1.

In this sense, Foundations of Information Security Mathematics functions as a transition from the formal study of algebraic structures to their role as computational models in cryptography, providing the conceptual basis for subsequent professional courses.

Table 1. Cryptography-oriented algebraic knowledge reconstruction

Layer 3	Cryptographic instantiations		
	DLP-based schemes	AES and ECC	lattice-based and homomorphic encryption
Layer 2	Structural properties		
	element order	multiplicative inverses	quotient structures
Layer 1	Computational spaces		
	cyclic groups	finite fields	quotient polynomial rings

4. FROM ABSTRACT STRUCTURES TO CRYPTOGRAPHIC COGNITION: A TEACHING DESIGN

4.1. Introducing Algebraic Concepts from Cryptographic Phenomena

In this course, algebraic concepts are introduced from concrete cryptographic operations rather than from formal definitions. Students first observe how a scheme works, and the corresponding concept is then used to explain the underlying structure.

In DH, the question of why a single element can produce all required values leads naturally to the notion of a generator and the structure of a cyclic group. In AES, the requirement that every nonzero byte be invertible motivates the introduction of finite fields and their polynomial construction. In lattice-based schemes, polynomial modular reduction appears as an operational rule, and the quotient ring is identified as the space in which the computation is closed.

Definitions are therefore presented as concise descriptions of structures that students have already encountered through cryptographic processes. Algebra becomes a tool for explaining algorithmic behavior rather than an isolated system of symbols.

By linking each concept to a cryptographic mechanism, students learn to ask structural questions: where the computation takes place and which properties make it valid. This marks the transition from symbolic manipulation to computational-space-oriented understanding.

4.2. A Computational-Space-First Teaching Sequence

With concepts introduced from cryptographic phenomena, the overall teaching sequence is reorganized according to the evolution of computational spaces rather than the traditional algebraic hierarchy.

The course begins with modular integer arithmetic, which serves as the most familiar computational environment. On this basis, cyclic groups are introduced to explain DLP-based protocols. Finite fields are then constructed to support the analysis of AES and elliptic curve operations. Finally, quotient polynomial rings are presented as the space for lattice-based and homomorphic cryptographic schemes.

In this sequence, each new structure is a natural extension of the previous one. Students do not encounter algebraic systems as disconnected topics but as progressively enriched computational environments. Content selection is guided by their relevance to subsequent cryptography courses, with emphasis on cyclic groups, finite field construction, and polynomial quotient rings.

Through this sequence, students form a continuous conceptual path from classical number-theoretic computation to abstract algebraic cryptographic spaces, which prepares them for later study without introducing additional theoretical layers.

4.3. A Classroom Mechanism for Reactivating Discrete Mathematics Knowledge

Previously learned notions of groups, rings, and fields are reintroduced as computational environments rather than as formal definitions. The classroom emphasis is not on repeating their axiomatic descriptions, but on reinterpreting them through cryptographic operations.

A direct comparison is made between the two courses. In discrete mathematics, a group is defined by its axioms; in this course, it appears as the space in which DH is executed. A finite field is no longer an abstract structure but the environment that guarantees invertibility in AES. In this way, earlier knowledge is transformed into an explanatory tool.

Students are encouraged to describe cryptographic procedures in structural terms, identifying the space of computation and the properties that make the operations valid. This shifts their use of algebra from recalling definitions to interpreting algorithms.

4.4. Competence Development for Subsequent Cryptography Courses

This teaching design aims to develop the ability to interpret cryptographic systems in structural terms. Instead of focusing on symbolic procedures, students learn to identify the algebraic space in which an algorithm operates and to explain its correctness and security through the properties of that structure.

With this training, ECC is understood as computation in a finite field–based group, lattice-based schemes as operations in quotient polynomial rings, and homomorphic encryption as structure-preserving computation over the same type of space. New topics in later courses therefore appear as extensions of an existing framework rather than as independent theories.

As a result, the transition from mathematical foundations to professional cryptography courses becomes continuous. The role of this course shifts from providing isolated prerequisite knowledge to establishing a unified structural perspective for subsequent study.

5. TEACHING PRACTICE AND REFLECTION

5.1. Implementation in the Classroom

The proposed design was implemented in the Foundations of Information Security Mathematics course for undergraduate students majoring in Information Security and Cyberspace Security. The course is positioned between discrete mathematics and the subsequent cryptography courses and therefore serves as a conceptual transition from formal algebraic definitions to their use as computational models.

In classroom teaching, algebraic notions were consistently introduced through cryptographic operations. The teaching sequence followed the evolution of computational spaces, beginning with modular integer arithmetic, moving to cyclic groups and finite fields, and ending with quotient polynomial rings. Previously learned concepts from discrete mathematics were revisited in this new context and reinterpreted as environments in which concrete algorithms are executed.

Instead of emphasizing formal derivations, the classroom focused on structural explanation. Students were guided to describe cryptographic procedures by identifying the underlying algebraic space and the properties that ensure the validity of the operations. In this way, definitions were used as explanatory tools rather than as isolated learning objects.

This implementation provided the practical basis for observing how a computational-space-oriented approach influences students' understanding of algebra and its role in cryptography.

5.2. Observed Changes in Students' Understanding

After the implementation of this approach, students increasingly interpreted cryptographic schemes in structural terms. Instead of listing computational steps, they identified the algebraic space in which an algorithm operates and used its properties to explain why the operations are valid. In discussions of AES and ECC, finite fields were referred to as the computational environment rather than as abstract definitions.

Previously separated topics were also described within a unified framework. Cyclic groups, finite fields, and quotient polynomial rings were recognized as different spaces for different classes of cryptographic schemes, which enabled structural comparisons between classical and lattice-based

constructions. At the same time, the dependence on memorized definitions was reduced. Definitions were more often used as concise descriptions of mechanisms that had already appeared in cryptographic processes.

These observations indicate a shift from procedural learning to a computational-space-oriented understanding, which supports the transition to subsequent cryptography courses.

5.3. Discussion: The Role of the Course in the Curriculum

The implementation of this approach redefines the role of Foundations of Information Security Mathematics in the curriculum. The course no longer functions as a collection of prerequisite topics, but as the structural foundation for subsequent cryptography courses.

With a computational-space-oriented perspective, later subjects appear as extensions of an existing framework rather than as new mathematical systems. This reduces the conceptual gap between mathematical foundations and professional study. Content selection is guided by its relevance to cryptographic mechanisms, which allows limited teaching hours to be used more effectively while maintaining coherence.

Overall, the course forms a transition from formal algebraic learning to the structural interpretation of cryptographic computation and supports the continuity of the curriculum.

6. CONCLUSION

This paper treats abstract algebra in Foundations of Information Security Mathematics as the computational spaces of cryptographic systems and reorganizes the course accordingly. Algebraic concepts are introduced to explain concrete mechanisms, and cyclic groups, finite fields, and quotient polynomial rings are presented as progressively extended environments. Classroom practice shows a shift from procedural learning to structural understanding, with later cryptography courses perceived as continuations of the same framework. The course thus moves from a set of prerequisites to the structural foundation of the curriculum, providing a continuous path from modular arithmetic to modern cryptographic computation under limited teaching hours.

ACKNOWLEDGEMENTS

Supported by 'the Fundamental Research Funds for the Central Universities (2023MS136)'

REFERENCES

[1] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press.

[2] Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). How people learn: Brain, mind, experience, and school. National Academy Press.

[3] Lyubashevsky, V., Peikert, C., & Regev, O. (2010). On ideal lattices and learning with errors over rings. In Advances in Cryptology – EUROCRYPT 2010 (pp. 1–23). Springer.

[4] Washington, L. C. (2008). Elliptic curves: Number theory and cryptography (2nd ed.). Chapman & Hall/CRC.

[5] Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES — The advanced encryption standard. Springer.

[6] Fan, J., & Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, 2012, 144.

[7] Harel, G., & Tall, D. (1991). The general, the abstract, and the generic in advanced mathematics. For the Learning of Mathematics, 11(1), 38–42.