Hindawi
Security and Communication Networks
Volume 2017, Article ID 9036382, 21 pages
https://doi.org/10.1155/2017/9036382

Research Article

Novel Noncommutative Cryptography Scheme Using Extra Special Group

Gautam Kumar and Hemraj Saini

Department of Computer Science & Engineering, Jaypee University of Information Technology, Solan 173234, India

Correspondence should be addressed to Gautam Kumar; gautam.kumar@mail.juit.ac.in

Received 19 July 2016; Revised 19 October 2016; Accepted 24 October 2016; Published 12 January 2017

Academic Editor: Pino Caballero-Gil

Noncommutative cryptography (NCC) is truly a fascinating area with great hope of advancing performance and security for high end applications. It provides a high level of safety measures. The basis of this group is established on the hidden subgroup or subfield problem (HSP). The major focus in this manuscript is to establish the cryptographic schemes on the extra special group (ESG). ESG is showing one of the most appropriate noncommutative platforms for the solution of an open problem. The working principle is based on the random polynomials chosen by the communicating parties to secure key exchange, encryption-decryption, and authentication schemes. This group supports Heisenberg, dihedral order, and quaternion group. Further, this is enhanced from the general group elements to equivalent ring elements, known by the monomials generations for the cryptographic schemes. In this regard, special or peculiar matrices show the potential advantages. The projected approach is exclusively based on the typical sparse matrices, and an analysis report is presented fulfilling the central cryptographic requirements. The order of this group is more challenging to assail like length based, automorphism, and brute-force attacks.

1. Introduction

Cryptography is a discipline of computer science, where algorithms and security practices are acting as a central tool. This is traditionally based on the mathematical foundation. The practical applications contain the assurance of legitimacy, protection of information from confessing, and protected message communication systems for essential requirements. To enforce security, the cryptographic schemes are concerning the vital role responsiveness in the field of security for numerous relevant applications all over the world. The absolute measure of cryptographic approaches shows the full-fledged appropriateness. But the serenities fondness with an assortment of more arbitrariness and impulsiveness with statistical responses is the motivational issue.

Public key cryptography (PKC) thought was first proposed by Diffie and Hellman [1]. Since then there are varieties of PKC algorithms that have been proposed, where Elliptic Curve Cryptography (ECC) [2, 3] in all of them has attracted the most attention in the cryptographic area. ECC has played a crucial role that made a big impact on the lower computational and communicational cost. Today ECC considered being tenable, but researchers are looking for alternative approaches for future security by not putting all the security protocols in one group only, that is, commutative group. On behalf of the open opinion, a brief analysis is presented below.

Peter's [4], in 1994, proposed a competent quantum algorithm for solving the discrete logarithm problem (DLP) and integer factorization problem (IFP). A Kitaev framework in 1996 [5] considered as a special case on DLP, called hidden subgroup or subfield problem (HSP). Stinson sensibly observed, in 2002, the most eternal PKCs belonging to a commutative or abelian group only, whose intention is susceptible in the forthcoming future. According to the same cryptographers Goldreich and Lee advised that we do not put all cryptographic protocols in one group. The reason was clear to introduce a new field of cryptography; this was only the opening of noncommutative cryptography [6]. Then afterwards for key exchange, encryption-decryption

(ED) and authentication schemes for cryptographic protocols on noncommutative cryptography were developed for various problems. Those were analogous protocols like the commutative cases. The elliptic curve over the HSP [7] comprehensively resolved on DLP, as recognized by ECC-DLP. The random HSP over noncommutative groups are well organized on quantum algorithms, which are well responsive. Further, the evidences are recommending that HSP over noncommutative groups are much harder [8].

The earlier structure of noncommutative cryptography was based on the braid based cryptography for the generalizations of the protocols. Afterward several other structures like Thompsons, polycyclic, Grigorchuk, or matrix groups/ring elements were proposed. The cryptographic primitives, methods, and systems of the noncommutative cryptography are based on algebraic structures of group, ring, and semiring elements. But in all of them matrix group of elements has shown the prospective advantages. In contrast, implementation in recent applications (protocols) using public key cryptographic approaches on Diffie-Hellman, RSA, and ECC is based on number theory. They are solving the various problems like session key establishments, encryption-decryption, and authentication schemes.

The basis of noncommutative cryptography is based on $*$ (contains reflection and/or rotation) operation on the noncommutative group G of $(G, *)$ that consists of group, ring, semiring, or some algebraic structural elements, in which two group elements a and b of G such that $a * b \neq b * a$ are known by noncommutative or nonabelian group. The group of these problems is broadly encompassed from mathematics and physics.

1.1. Background. The generation of noncommutative cryptographic approach has a solid backbone for security enhancements and performances. A course of numerous attempts has been made available for the same. A brief analysis is described below.

(i) Wagner and Magyarik in 1985 [9] proposed undecidable word problems on semigroup elements for public key cryptography (PKC). But Birget et al. [10] pointed out that it is not based on word problem and proposed a new system on finitely generated groups with a hard problem.

(ii) On braid based cryptography, there is a compact key established protocol proposed by Anshel et al. [11] in 1999. The basis was difficulty in solving equations over algebraic structures. In their research paper, they also recommended that braid groups may subsist to be a good alternate platform for PKC.

(iii) Afterwards, Ko et al. in 2000 [12] anticipated a new PKC by using braid groups. The Conjugacy Search Problem (CSP) is the intractability security foundation, such as effective canonical lengths and braid index when they are chosen suitably. Further, the area under consideration met with immediate successes by Dehornoy in 2004 [13], Anshel et al. in 2003 [14], Anshel et al. in 2006 [15], and Cha et al. in 2001 [16]. Despite the fact, 2001 to 2003, recurring cryptanalytic sensation, Ko et al. in 2002 [17] and Cheon and Jun in 2003 [18] diminished the initial buoyancy on the noteworthy theme, in Hughes and Tannenbaum in 2000 [19]. Many numbers of authors even proclaimed the impetuous death of the braid based PKC, Bohli et al. in 2006 [20] and Dehornoy in 2004 [21]. Dehornoy's paper conducted a survey on the state of the subject with significant research, but still being desirable for accomplishing a definite final conclusion on the cryptographic prospective of braid groups.

(iv) In 2001, Paeng et al. [22] also published a new PKC built on finite nonabelian groups. Their method is based on the DLP in the inner automorphism group passing through the conjugation accomplishment. These were further improved, named MOR systems.

(v) In the meantime, one-way function and trapdoors generated on the finite fields were remarkable in group theory by Magliveras et al. in 2002 [23]. Later on, in 2002 Vasco et al. [24] confirmed an appropriate generality on factorization and a uniform description of several cryptographic primitives on convincing homomorphic cryptosystem; those were constructed in the first time for nonabelian groups. Meanwhile, Magliveras et al. in [25] proposed a new approach to designing public key cryptosystems using one-way functions and trapdoors in finite groups. Grigoriev and Ponomarenko [26] and Grigoriev and Ponomarenko [27], consequently, extended the difficulty of membership problems on integer matrices for a finitely generated random group of elements.

(vi) The arithmetic key exchange is enlightened by Eick and Kahrobaei 2004 [28] and an innovative cryptosystem on polycyclic groups is proposed by them. The structures of polycyclic groups are a complex of their own cyclic group. The algorithmic theory and investigation properties are more difficult which seems to have a more open proposal. The progression tenure is a succession of subgroups of a group $G = G_1 \rhd G_2 \rhd \cdots \rhd G_{n+1} = \{1\}$. For each term of the series, succession not only is in the entire group but also is not contained in the former term. A group G is called polycyclic series with cyclic aspects; that is, G_i/G_{i+1} is recurring for $i = 1, \ldots, n$.

(vii) Shpilrain and Ushakov in 2005 [29] recommended that Thompson's group is a good proposal for building PKCs. The assumption under the decomposing problem is intractable, ancillary to the Conjugacy Search Problem, described over R.

(viii) In 2005, Mahalanobis [30] did not discriminate the D-H key exchange protocol from a cyclic group to a finitely presented nonabelian nilpotent group of class 2. The nilpotent group is a normal series to each quotient H_i/H_{i+1} lying in the center of G/H_{i+1} and supposed to be a central succession. A class of nilpotent group is the shortest series length with its shortest nilpotency degree. Engendered nilpotent finite groups are polycyclic groups and moreover

have a central series with cyclic factors. Also in 2006 Dehornoy proposed an authentication scheme based on the left self-distributive (LD) systems. This idea was further developed by introducing the concept of the one-way LD system structured by Wang et al. in 2010 [31]. The algebraic association $(A, *)$ is called left self-distributive for all elements $a, b, c \in A$, $a * (b * c) = (a * b)(a * c)$.

(ix) To extract from a given $a * b$ and b, this system is said to be one way, if it is intractable. LD system, in general, is much different from groups or semigroups or semirings. Even the regarding facts are not associative, so solitarily describing a nontrivial LD system over any noncommutative group G via the mapping $a * b \triangleq ab\bar{a}$.

(x) Moreover, if the Conjugacy Search Problem (CSP) over G is intractable, the derivative of LD system is treated as being one way.

(xi) In 2007, Cao et al. [32] proposed a method to use polynomials over noncommutative rings or semigroups to build cryptographic scheme. This method is referred to as the $\mathbb{Z}$-modular method. Further, the protocol application was based on nonabelian dihedral order 6 by Kubo in 2008 [33] which is the initial order for this group and construction is based on three-dimensional revolutions.

(xii) In 2008, Reddy et al. [34], $\mathbb{Z}$-modular method was used to build signature schemes over noncommutative groups and division rings.

(xiii) The cryptographic protocol implementation was constructed on four-dimensional considerations by D. N. Moldovyan and N. A. Moldovyan in 2010 [35]. The perspectives were the generalizations for security enhancement on the basis of noncommutative groups.

(xiv) In 2014, Myasnikov and Ushakov [36] cryptanalyzed the authentication scheme proposed by Shpilrain and public key encryption to use the hardness of the Conjugacy Search Problem in noncommutative monoids. A heuristic algorithm, was devised by those to solve these problems and declared that these protocols are anxious.

(xv) Svozil in 2014 [37] proposed the metaphorical recognized hidden variable on noncontextual indecisiveness that cannot be comprehended by quantum systems. The cryptanalytic attacks are not accompanied and aligned by quasi configurations, and the theorems do not subsist assembled proofs reclining over the same.

1.2. Motivation and Our Contribution. The issues related to the ring structure of the group elements are one of the most motivational concerns. A typical semiring structure, such as sparse matrices, shows the potential advantages towards a possible way to avoid the various attacks. The initial order for general and monomials [original parameters are hidden, and monomials structures provably equivalent consideration takes part in computation process] structure on polynomial $\mathbb{Z}$-modular noncommutative cryptography is the foundation.

Our contribution is in multidisciplinary scenario on extra special group on the cryptographic protocol regarding the key exchange, encryption-decryption, and authentication in four-dimensional perspective. The key idea is based on a special case of prime order with more resistance to attacks, and proposed approach works on the bigger range of probabilistic theories.

1.3. Manuscript Organization. The content of manuscript is organized into its subsequent sections. The next section presents cryptographic preliminaries for modular polynomial assumptions on general scalar multiplication and monomials like scalar multiplication on group, ring, and semiring elements. Section 3 presents the fundamental of the proposed work on the extra special group and its elementary analysis is elaborated. Sections 4 and 5 are our core parts, where our considerations are perfectly set aside on the general protocol schemes for the session key establishment, encryption-decryption, and authentication schemes; further similar schemes on monomials are presented on a combinational congruence on group, ring, or semiring elements. In Section 6, a brief idea is presented to achieve the bigger search space for the length based attack, which gives its security guarantees. Finally, the work concludes, along with references.

2. Preliminaries

2.1. $\mathbb{Z}$-Modular Assumptions on Noncommutative Cryptography. The scalar multiplication is the basis for all cryptographic computations. The major goal of scalar multiplication is to generate the discrete logarithmic value. A new public key cryptography on polynomials scalar multiplication over the noncommutative ring R is proposed by Cao et al. in 2007 [32]. The developed scheme is based on modulo prime integers, named $\mathbb{Z}$-modular method. The derived $\mathbb{Z}$-modular structure on ring is $\mathbb{Z}(r)$, and this structure applies to positive $\mathbb{Z}^+[r]$ and/or negative $\mathbb{Z}^-[r]$ on noncommutative ring R elements, where $r \in R$ is undetermined. Also, group and semiring are comprehensively applicable on $\mathbb{Z}$-modular assumptions.

2.1.1. Noncommutative Rings on $\mathbb{Z}$-Modular Method. The integral coefficient polynomial on additive noncommutative characterization is defined on ring $(R, +, 0)$ and for multiplicative noncommutative $(R, \cdot, 1)$; the scalar multiplication over R is well defined for $k \in \mathbb{Z}^+$ and $r \in R$:

$$(k)\, r \cong \underbrace{r + \cdots + r}_{k \text{ times}}. \tag{1}$$

Further, for $k \in \mathbb{Z}^-$,

$$(k)\, r \cong \underbrace{(-r) + \cdots + (-r)}_{-k \text{ times}}. \tag{2}$$

Finally, if it is to be defined on scalar $k = 0$, it is likely to be $(k)r = 0$.

Proposition 1. *In general, scalar multiplication on noncommutative property follows* $(a)r \cdot (b)s \neq (b)s \cdot (a)r$, *when* $r \neq s$. *Recall a polynomial with positive integral coefficient* $f(x) = a_0 + a_1x + \cdots + a_nx^n \in \mathbb{Z}^+$, *for all* x. *To assign the component* x *as an element* $r \in R$, *then attain a precise element in ring* R *as*

$$f(r) = \sum_{i=0}^{n} (a_i)^{r^i} = a_0 \cdot 1 + a_1 \cdot x + \cdots + a_n \cdot x^n. \quad (3)$$

In addition, suppose r *to be undetermined; then polynomial over* $f(r)$ *is univariable polynomial lying on* R. *The univariable polynomial over* R *as a whole set is denoted as* $\mathbb{Z}^+[r]$, *and it is defined as follows for the respective functions on two different ring elements:*

$$f(r) = \sum_{i=0}^{n} (a)_i^{r^i} \in \mathbb{Z}^+[r],$$
$$h(r) = \sum_{j=0}^{m} (b)_j^{r^j} \in \mathbb{Z}^+[r]. \quad (4)$$

Again, if $n \geq m$, *then*

$$\left(\sum_{i=0}^{n} (a_i)^{r^i}\right) + \left(\sum_{j=0}^{m} (b_j)^{r^j}\right) = \sum_{i=0}^{m} (a_i + b_i) r^i + \sum_{i=m+1}^{n} (a_i)^{r^i} \quad (5)$$

and according to the property of distributive law it generalizes the above equation as

$$\left(\sum_{i=0}^{n} (a_i)^{r^i}\right) + \left(\sum_{j=0}^{m} (b_j)^{r^j}\right) = \left(\sum_{i=0}^{n+m} (p_i)^{r^j}\right), \quad (6)$$

where,

$$p_i = \left(\sum_{j=0}^{i} (a_j b_{i-j})^{r^i}\right) = \sum_{j+k=i} (a_j)(b_k). \quad (7)$$

Theorem 2. $f(r) \cdot h(r) = h(r) \cdot f(r)$, $\forall f(r) \in \mathbb{Z}^+[r]$ *and* $\forall h(r) \in \mathbb{Z}^+[r]$, *where* $\forall$ *signify for all elements.*

Proof. Here ring r is a subset of ring R applying to polynomial functions of $f(r)$ and $h(r)$, for all positive integers of $\mathbb{Z}^+[r]$. A ring is a set of elements with two binary operations of addition and multiplication satisfying the following case properties on commutative law, associative law, identity, inverse, and closure. In addition to the same some more properties are also satisfying for all ring elements as follows:

(i) Closure under multiplication: if a and b belong to ring element, then ab is also in ring.

(ii) Associative law of multiplication: $a(bc) = (ab)c$ for all a, b, c.

(iii) Distributive laws: $a(b+c) = ab+ac$ or $(a+b)c = ac+bc$ for all a, b, c.

(iv) Commutative multiplication: $ab = ba$ for a, b.

(v) Multiplicative identity: $a \cdot 1 = 1 \cdot a = a$ for all a.

(vi) No zero divisors: for all a and b in R and $ab = 0$, then, either $a = 0$ or $b = 0$ and this does not follow on dividing by zero.

Therefore, this theorem proofs itself on the above properties of (i), (iii), and (iv). □

2.2. Two Well-Known Cryptographic Assumptions. The assumptions of security strength are due to the difficulty of the following two problems:

(i) *Conjugacy Decisional Problem (CDP)*. It is, on the given two group elements a and b, to determine a random x to produce the value of other group elements, such that $b = a^x$ or to produce the same using the Conjugacy multiplicative inverse as $b = x^{-1}ax$.

(ii) *Conjugacy Search Problem (CSP)*. It is, for the two group elements of a and b in a group G, to find whether there exists x in G, such that $b = a^x$ or Conjugacy multiplicative inverse $b = x^{-1}ax$.

If no algorithm exists to solve the CSP, by applying x on one group of elements to determine the other group of elements, that is, $a \rightarrow b^x$, then this is considered to be a one-way function. In the contemporary computation, both problems on general noncommutative group G are too complicated enough to determine the assumptions on cryptographic primitives. The CSP assumptions are difficult enough to solve this problem on probabilistic polynomial time, whereas CDP assumptions are a unique representation for any group, ring, or semiring elements for cryptographic use. The CDP transition on all these assumptions finishing efficiently over each other is one of the major advantages.

2.3. Using Monomials in $\mathbb{Z}$-Modular Method. The polynomials used in $\mathbb{Z}$-modular method are constrained to be monomials; that is, if the original information of group elements is hidden with its equivalents ring or semiring elements, such participation in computation is viewed as a special case. Under these considerations new creations of public key encryption schemes from Conjugacy Search Problems are proposed.

2.3.1. Conjugacy Search Problem. Let $(G, \cdot, 1)$ be a noncommutative monomial for an element $a \in G$, the other group element being $b \in G$, such that $a \cdot b = b \cdot a$; then it assumed that group a is invertible, and call b an inverse of a. Not all elements in G are invertible. If the inverse of a exists, it is unique and denoted by a^{-1}. In monomials, the positive power of a group element for n integer is described as follows: $a^n = \underbrace{a + \cdots + a}_{n \text{ times}}$ for $n > 1$. If b is the inverse of a, one can also define the negative power of a by setting $a^{-1} = \underbrace{b + \cdots + b}_{n \text{ times}}$ for $n > 1$. The Conjugacy Search Problem can be extended to

monomials G, for $\forall a \in G$ and $\forall x \in G^{-1}$, xax^{-1} is a conjugate to a, and call x as conjugator of the pair (a, xax^{-1}).

Definition 3 (Conjugacy Search Problem (CSP)). Let G be a noncommutative monomial; the two elements $a, b \in G$ so that $b = xax^{-1}$ for some unknown element $x \in G^{-1}$, and the objective of the CSP in G is to find $x \in G^{-1}$ such that $b = x'ax'^{-1}$.

Definition 4 (left self-distributive system). Suppose that S is a nonempty set, $F : S \times S \to S$ is a well-defined function, and let one denote $F(a, b)$ by $F_a(b)$. If the rewritten formula $F_r(F_s(p)) = F_{F_r(s)}(F_r(p))$, $(\forall p, r, s \in S)$ holds then call $F_\cdot(\cdot)$ as a left self-distributive system (LD).

Theorem 5. *Let G be noncommutative monomial, the function F on conjugate follows as $F : G^{-1} \times G \to G$, $(a, b) \mapsto aba^{-1}$ and is known by LD system, also abbreviated as Conj-LD.*

Proof. According to Definition 4, the term LD is an analogical observation from $F_r(s)$ as a binary operation in $r * s$; then this is observed as $r * (s * p) = (r * s) * (r * p)$, where "$*$" is left distributive with respect to itself. The proof of these observations is following from left-hand side to right-hand side as follows: $F_r(F_s(p)) = F_r(s * p) = r * (s * p) = (r * s) * (r * p) = F_r(s) * F_r(p) = F_{F_r(s)}(F_r(p))$. Here, it is satisfying the function F on $F_r(F_s(p)) = F_{F_r(s)}(F_r(p))$ as an LD system. Thus, Theorem 5 proof is based on these observations. □

Proposition 6. *Let F be a Conj-LD system defined over a noncommutative monomial G; then for the given $a \in G^{-1}$ and $b, c \in G$, the following proposals are well-defined, according to [38]:*

(i) $F_a(a) = a$.

(ii) $F_a(b) = c \Leftrightarrow F_{a^{-1}}(c) = b$.

(iii) $F_a(bc) = F_a(b)F_a(c)$.

Proof of (i). Since $aaa^{-1} = a$, so $F_a(a) = a$. □

Proof of (ii). $F_a(b) = c \xrightarrow{\text{yields}} aba^{-1} = c \xrightarrow{\text{yields}} a^{-1}ca = b \xrightarrow{\text{yields}} F_{a^{-1}}(c) = b$. □

Proof of (iii). $F_a(bc) = a(bc)a^{-1} = (aba^{-1})(aca^{-1}) = F_a(b)F_a(c)$. □

2.4. Symmetry and Generalization Assumptions over Noncommutative Groups. To explain the symmetries, the generalizations on the noncommutative cryptography are the following problems on group G:

(i) *Symmetrical Decomposition Problem (SDP).* Given $(a, b) \in G$ and $m, n \in Z$, find $x \in G$ such that $b = x^m \cdot a \cdot x^n$.

(ii) *Generalized Symmetrical Decomposition Problem (GSDP).* Given $(a, b) \in G$, $S \subseteq G$, and $m, n \in Z$, find $x \in G$ such that $b = x^m \cdot a \cdot x^n$.

The GSDP is evidently a sort of constrained SDP, and if subset S is large enough, then membership information in general does not help one to extract x from $x^m \cdot a \cdot x^n$. Now, it is understood that GSDP is at least as rigid as SDP. The following GSDP hypothesis says that it is not flexible to solve the same in probabilistic polynomial time with nonnegligible precision with respect to problem scale. In this regard these works are like discrete logarithm problem (DLP) over group G.

2.5. Computational Diffie-Hellman (CDH) Problem over Noncommutative Group G. The CDH problem with respect to its subset S on noncommutative group determines $a^{x_1x_2}$ or $a^{x_2x_1}$ for known a, a^{x_1}, and a^{x_2}, where $x \in G$, $x_1, x_2 \in S$. The commutative law keeps an extraction property from $x_1 \in G(x_2)$, if it holds the relation for $a^{x_1x_2} = a^{x_2x_1}$. It is noticeable that DLP in GSDP over G is tractable. But the converse of the same is not true. At present, no clue is available to resolve this problem without extracting x_1 (or x_2) from a and a^{x_1} (or a^{x_2}). Then, the CDH hypothesis over G says that problem over G is intractable. In this regard, no such probabilistic polynomial time algorithm exists to solve the dilemma with significant accuracy to problem extent. The same definition is also well distinct for a noncommutative semiring. Hence, DLP of GSDP and CDH assumptions over noncommutative semigroup are well appropriate.

3. Extra Special Group

The definition says that any prime p to the power $1 + 2n$, that is, p^{1+2n}, sustains the twofold properties: (i) Heisenberg group and semidirect product of cyclic group order and/or (ii) dihedral order 8 and quaternion group. The two belonging group elements revolve around fixed center, known by extra special group [39]. These are based on finite size fields on modulo primes and analogues to group elements following sparse matrices properties. Due to this reason, group contains the dual identity, which meets the requirement for perfect cryptography. The quotients or remainders belong to *nontrivial (*nontrivial refers to terms or variables that are not equal to zero or identity after resultant) element, whose center is cyclic. Since its size is prime, so its classification is based on either prime $p = 2$ or $p =$ odd. The reason is clearly that any prime starts from 2, and the remaining primes only belong to odd numbers.

At $p = odd$. The extra special group, for p odd, is given below:

(i) The group of triangular 3×3 matrices is over the field with p elements, with 1s on the diagonal. The group is exponent p for p odd. These are known by Heisenberg group elements.

(ii) There is the semidirect product of a cyclic group of order p^2 by a cyclic group of order p acting nontrivially on it.

Again, if n is a positive integer, then for p odd the following is given.

(i) The central product of n extra special group of order p^3, all of exponent p: this extra special group also has exponent p.

(ii) The central product of order p^3: at least one of the exponents should be p^2.

Now, consider *prime* $p = 2$; the minimum order starts from $n = 1$, so extra special group order $8 = 2^3$ is described as follows:

(i) The dihedral group D_8 in order 8: this group has 4 elements of order 4.

(ii) The quaternion group of order 8: it has six elements of order 4, for example,

$$\begin{bmatrix} 1 & a & b & c \\ 0 & 1 & d & e \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Again, if we consider n as a positive integer for quaternion groups then,

(i) for an odd integer, the central product is in the quaternion group;

(ii) for an even integer, the central product is in the quaternion group.

3.1. Heisenberg Group. A group of 3×3 upper triangular matrices contains the several representations in terms of functional spaces whose center acts nontrivially on it. A matrix multiplication is in the form

$$\begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix}, \tag{9}$$

where elements of a, b, c belong to commutative ring elements. Further, the real/integer numbers belong to ring structured elements, known by respective continuous/discrete Heisenberg group [40]. The continuous group comes from the description of quantum systems in one dimension. The association with n-dimensional systems is more general in this regard. The products of two Heisenberg matrices in the three-dimensional case are given by

$$\begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & a' & b' \\ 0 & 1 & c' \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & a + a' & b + b' + ac' \\ 0 & 1 & c + c' \\ 0 & 0 & 1 \end{pmatrix}. \tag{10}$$

The Heisenberg neutral element of group is the identity matrix. The discrete Heisenberg group x, y, z generator is the nonabelian group of ring elements on the integers a, b, c:

$$x = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \tag{11}$$

and relations $z = xx^{-1}yy^{-1}$, $xz = zx$, and $yz = zy$, where $z = \left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$ is the generator with the center. A polynomial growth rate of order 3 using Bass's theorem is used to generate any element through

$$x = \begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix} = y^b z^c x^a. \tag{12}$$

The behavior of the Heisenberg group to modulo odd prime p over a finite field is called extra special group of exponent p.

3.1.1. Security Strength of Heisenberg Group. The Heisenberg group on public key cryptography follows polycyclic behaviors if and only if a subseries $G_1 \triangleright G_2 \triangleright \cdots \triangleright G_{n+1} = \{1\}$ for a group G (where $\triangleright$ denotes variations of G in cyclic form). Each positive integer of the nth elements of Heisenberg generates an infinite nonabelian form (using the binary addition and multiplication operations on matrix or sparse matrix elements), which makes the scheme of Heisenberg be practical choice for an efficient implementation on hardware and software. This gives a unique normal form just after group operations, so the group may be considered to be an effective solution provider for cryptographic use.

3.2. Dihedral Order 8. The dihedral order is a group of operations on a finite set of elements that includes the problems of mathematics and physics. A cycle of rotations and reflections on group elements is the basis that forms the properties of this group. One of the simplest examples of nonabelian group is dihedral order 6 [41].

In the proposed work the minimum order is dihedral order 8, denoted by D_8 (or also called D_4) [42]. The subgroups of this (dihedral order group G) are generated by rotations and/or reflections; those are forming a cyclic subgroup which is one of the key advantages. For representing the dihedral order 8, a glass square of certain thickness with letter "F" is considered. The identity element is denoted by e. The letter "F" makes a visible difference upon rotation on 0°, 90°, 180°, and 270° in clockwise directions, as shown in Figure 1; it is further used in cryptographic purposes.

One more "b" (reflection) operation is used relative to its corresponding above four rotations. Further, to define the

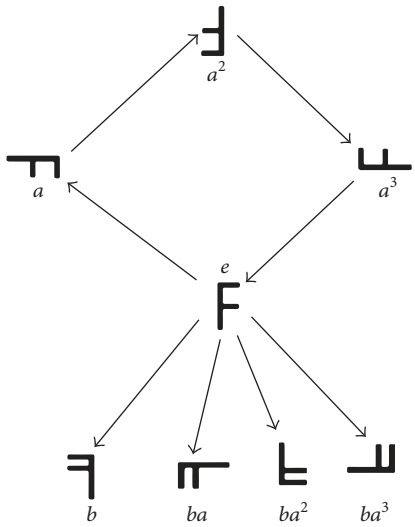

Figure 1: Symmetries of dihedral order 8.

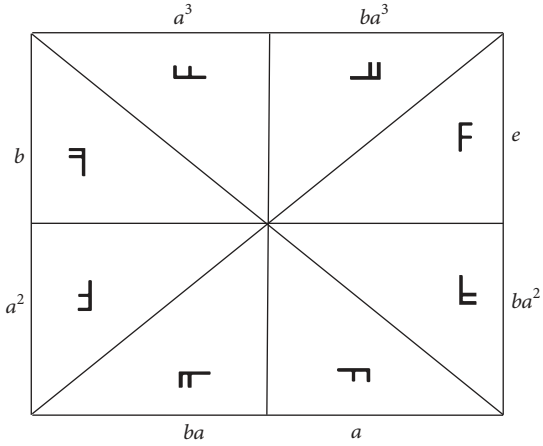

Figure 2: Four-dimensional representations.

composition movement such as "ba," first do the operation for "a" and after that apply "b" as shown. For the remaining two of ba^2 and ba^3 are working as the previous one, now, after the corresponding operation, the same can be represented in four dimensions, as depicted in Figure 2. The group element is a property whose center and derived subgroup are fixed on explicit limitations under it.

The abstract movements of all operations are fixed on certain boundaries, and these are generally represented by Cayley graph. The graph is mixed with eight vertices, four edges, and eight arrows. This is one of the fundamental tools in combinatorial theory to make group elements revolve around the fixed axis, as elaborated in Figure 3.

Again, a table known by Cayley table is presented for a finite set of elements in all possible permutations by arranging its products in a square table reminiscent to multiplication. For the same, the dihedral order 8 based Cayley table is shown in Table 1.

Finally, we are correlating the same concept from mathematics. Here, the composition of eight different but interrelated operations for D_8 is particularly specified for the

Table 1: Cayley table.

	e	a	a^2	a^3	b	ba	ba^2	ba^3
e	e	a	a^2	a^3	b	ba	ba^2	ba^3
a	a	a^2	a^3	e	ba^3	b	ba	ba^2
a^2	a^2	a^3	e	a	ba^2	ba^3	b	ba
a^3	a^3	e	a	a^2	ba	ba^2	ba^3	b
b	b	ba	ba^2	ba^3	e	a^3	a^2	a
ba	ba	ba^2	ba^3	b	a	e	a^3	a^2
ba^2	ba^2	ba^3	b	ba	a^2	a	e	a^3
ba^3	ba^3	b	ba	ba^2	a^3	a^2	a	e

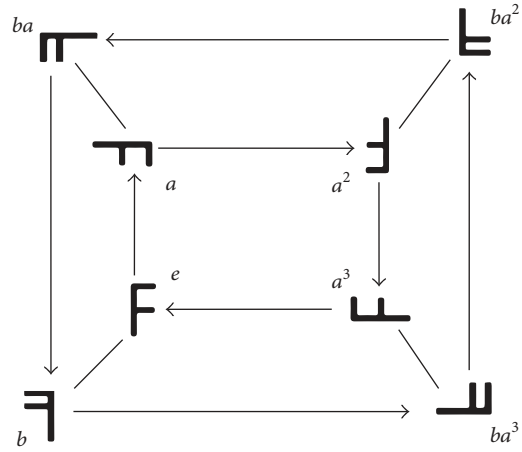

Figure 3: Cayley graph of D_4.

mathematical suites that will be used in cryptographic applications, where mathematics is the foundation for almost all applications. Here is a similar consideration of the above concept on square glass; a different perspective to distinguish the same for the cryptography purposes is presented as a schematic representation in Figure 4. These are in the ordered group elements from G_1 to G_8 for rotations/movements and reflections in $e, a, a^2, a^3, b, ba, ba^2, ba^3$, as a resultant. A detailed cryptographic application scheme is considered in Sections 5.3 and 5.4.

3.3. Quaternion Group. The quaternion group [43] is a nonabelian order of eight elements that forms of four-dimensional vector space over the real numbers. These are isomorphic to a subset of certain eight elements under multiplication. The group is generally indicated by Q or Q_8 and is given by the group representation $Q = (-1, i, j, k) \mid (-1)^2 = 1,\ i^2 = j^2 = k^2 = ijk = -1$, where 1 is the identity element and -1 commutes with the same. This is the same order as dihedral order D_8, but the only difference is in its structure. So, it may be considered an immoderation of dihedral order 8. Depicted in Table 2 is a Cayley table for Q_8.

3.3.1. Security Strength of Quaternion Group. Quaternion group using number theory gives multifold security properties in cryptography. The real beauty of quaternion is noncommutative nature and multiplication on these groups lies on a sphere in four-dimensional space. Due this nature,

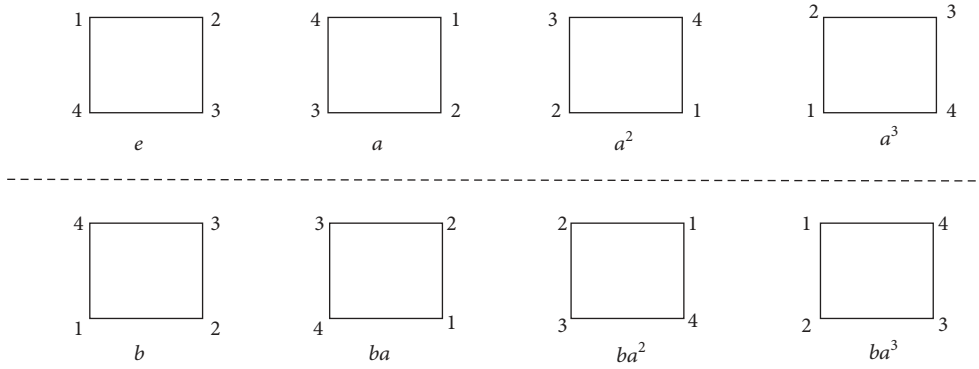

Figure 4: Schematic representation on dihedral order 8.

Table 2: Cayley table (quaternion group).

	1	−1	i	$-i$	j	$-j$	k	$-k$
1	1	−1	i	$-i$	j	$-j$	k	$-k$
−1	−1	1	$-i$	i	$-j$	j	$-k$	k
i	i	$-i$	−1	1	k	$-k$	$-j$	j
$-i$	$-i$	I	1	−1	$-k$	k	j	$-j$
j	j	$-j$	$-k$	k	−1	1	i	$-i$
$-j$	$-j$	J	k	$-k$	1	−1	$-i$	i
k	k	$-k$	j	$-j$	$-i$	i	−1	1
$-k$	$-k$	k	$-j$	j	i	$-i$	1	−1

highest level of probable confusion can be achieved in applied applications and it can derive for enormous applications. The used matrices and algebra (where multiplication order is important for end user applications) make quaternion natured group elements bigger significance for the future security proposals. The resultant of quaternion easily converts to other representations just like the two original unit quaternions, where from the adversary side it is likely to be impossible to break such kind of scheme. Further, still analysis and implementation in cryptography are the need, which may give a high security specification on the quaternion group.

4. Noncommutative Cryptography on Groups and Rings

The mathematical rationalization over matrix group or ring is exemplified on $M(\mathbb{Z}_N)$, based on $N = p \cdot q$, where p and q are two secure primes. This is intractable, in view of the fact that $A = \left(\begin{smallmatrix} a & 0 \\ 0 & 0 \end{smallmatrix}\right) \in M(\mathbb{Z}_N)$, $a \in (\mathbb{Z}_N)$, from $A^2 = \left(\begin{smallmatrix} a^2 & 0 \\ 0 & 0 \end{smallmatrix}\right) \in M(\mathbb{Z}_N)$ with no significant factors of N [32].

The above-mentioned ring can be enhanced with respect to security by using special or peculiar sparse matrices of rings elements, as our contribution, which shows the stronger security specifications on described Sections 3.1.1 and 3.3.1 which are based on Heisenberg and quaternion group, respectively. Further, noncommutative nature of generated semiring elements for dihedral order of 8 (presented in next section) also satisfies the properties of N very well.

4.1. Key Exchange Algorithm on Noncommutative Cryptography. The noncommutative key exchange cryptography works reminiscent of Diffie-Hellman key exchange [44] similar to a commutative case, but the major distinction is the itinerary actions on selection of global parameters, generation of private keys, production rule for shared secret session keys, and encryption-decryption. The effectiveness of the algorithm depends on the impenetrability of computing the DLP. The security of the algorithm lies in the prime factorization on two secure primes, random private polynomial chosen by user A and user B, respectively. A detailed elaboration through the numerical example on ring and quaternion group for key exchange and encryption-decryption is presented in this section, which belongs to the extra special group.

The key exchange agreement over matrix ring elements is depicted as follows.

Key Exchange on Noncommutative Ring

Global Public Parameters

m, n: integers Z^+

a, b: ring elements

User A Key Generation

(i) Select private random polynomial: $f(x)$.

(ii) If $f(a) \neq 0$, then $f(a)$ is considered as private key.

(iii) Generation of public key X_A: $X_A = f(a)^m \cdot b \cdot f(a)^n$.

User B Key Generation

(i) Select private random polynomial: $h(x)$.

(ii) If $h(a) \neq 0$, then $h(a)$ is considered as private key.

(iii) Generation of public key X_B: $X_B = h(a)^m \cdot b \cdot h(a)^n$.

Generation of Secret Key by User A

$$K_A = f(a)^m \cdot X_B \cdot f(a)^n.$$

Generation of Secret Key by User B

$$K_B = h(a)^m \cdot X_A \cdot h(a)^n.$$

The global parameters are

$$\begin{aligned} m &= 3, \\ n &= 5, \\ a &= \begin{pmatrix} 17 & 5 \\ 7 & 4 \end{pmatrix}, \\ b &= \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix}, \\ N &= 7 * 11. \end{aligned} \tag{13}$$

User A chose their random polynomial $f(x) = 3x^3 + 4x^2 + 5x + 1$. Evaluate the polynomial $f(a)$; if $f(a) \neq 0$ then the polynomial will be considered as a private key for user A. The A's private key is as follows:

$$\begin{aligned} f(a) &= 3\begin{pmatrix} 17 & 5 \\ 7 & 4 \end{pmatrix}^3 + 4\begin{pmatrix} 17 & 5 \\ 7 & 4 \end{pmatrix}^2 + 5\begin{pmatrix} 17 & 5 \\ 7 & 4 \end{pmatrix} + 1 \cdot I \\ &= \begin{pmatrix} 19 & 20 \\ 28 & 44 \end{pmatrix} \bmod 77. \end{aligned} \tag{14}$$

Now, the generation of public key X_A by user A is as follows:

$$\begin{aligned} X_A &= f(a)^m \cdot b \cdot f(a)^n \\ &= \begin{pmatrix} 19 & 20 \\ 28 & 44 \end{pmatrix}^3 \cdot \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix} \cdot \begin{pmatrix} 19 & 20 \\ 28 & 44 \end{pmatrix}^5 \\ &= \begin{pmatrix} 3 & 56 \\ 9 & 2 \end{pmatrix} \bmod 77. \end{aligned} \tag{15}$$

At the other end user B chose their own random polynomial $h(x) = x^5 + 5x + 1$. Further, evaluate the polynomial $h(a)$, and if $h(a) \neq 0$ then this polynomial value will be considered as private key:

$$\begin{aligned} h(a) &= \begin{pmatrix} 17 & 5 \\ 7 & 4 \end{pmatrix}^5 + 5\begin{pmatrix} 17 & 5 \\ 7 & 4 \end{pmatrix} + 1 \cdot I \\ &= \begin{pmatrix} 70 & 52 \\ 42 & 58 \end{pmatrix} \bmod 77 \end{aligned} \tag{16}$$

and the generation of public key for user B is as follows:

$$\begin{aligned} X_B &= h(a)^m \cdot b \cdot h(a)^n \\ &= \begin{pmatrix} 70 & 52 \\ 42 & 58 \end{pmatrix}^3 \cdot \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix} \cdot \begin{pmatrix} 70 & 52 \\ 42 & 58 \end{pmatrix}^5 \\ &= \begin{pmatrix} 0 & 39 \\ 35 & 68 \end{pmatrix} \bmod 77. \end{aligned} \tag{17}$$

Finally, the session key is extracted by the user A as K_A:

$$\begin{aligned} K_A &= f(a)^m \cdot X_B \cdot f(a)^n \\ &= \begin{pmatrix} 19 & 20 \\ 28 & 44 \end{pmatrix}^3 \cdot \begin{pmatrix} 0 & 39 \\ 35 & 68 \end{pmatrix} \cdot \begin{pmatrix} 19 & 20 \\ 28 & 44 \end{pmatrix}^5 \\ &= \begin{pmatrix} 21 & 37 \\ 49 & 69 \end{pmatrix} \bmod 77 \end{aligned} \tag{18}$$

and the session key is extracted from user B as K_B:

$$\begin{aligned} K_B &= h(a)^m \cdot X_A \cdot h(a)^n \\ &= \begin{pmatrix} 70 & 52 \\ 42 & 48 \end{pmatrix}^3 \cdot \begin{pmatrix} 3 & 56 \\ 9 & 2 \end{pmatrix} \cdot \begin{pmatrix} 70 & 52 \\ 42 & 48 \end{pmatrix}^5 \\ &= \begin{pmatrix} 21 & 37 \\ 49 & 69 \end{pmatrix} \bmod 77. \end{aligned} \tag{19}$$

4.2. Key Exchange Using Heisenberg Group (Upper Triangular Matrices). Further, we applied the same algorithm for session key establishment over a Heisenberg group. It is demonstrated on the global parameters, where assumptions are

$$\begin{aligned} m &= 3, \\ n &= 5, \\ a &= \begin{pmatrix} 1 & 5 & 7 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}, \\ b &= \begin{pmatrix} 1 & 6 & 9 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}, \\ N &= 7 * 11. \end{aligned} \tag{20}$$

For user A, a random polynomial is chosen: $f(x) = 3x^3 + 4x^2 + 5x + 6$. Evaluate the polynomial on $f(a)$, and if $f(a) \neq 0$ then polynomial value is considered as a private key for user A:

$$\begin{aligned} f(a) &= 3\begin{pmatrix} 1 & 5 & 7 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}^3 + 4\begin{pmatrix} 1 & 5 & 7 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}^2 \\ &\quad + 5\begin{pmatrix} 1 & 5 & 7 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} + 6I \\ &= \begin{pmatrix} 18 & 33 & 29 \\ 0 & 18 & 11 \\ 0 & 0 & 18 \end{pmatrix} \bmod 77. \end{aligned} \tag{21}$$

The generation of public key X_A by user A is as follows:

$$X_A = f(a)^m \cdot b \cdot f(a)^n = \begin{pmatrix} 18 & 33 & 29 \\ 0 & 18 & 11 \\ 0 & 0 & 18 \end{pmatrix}^3 \cdot \begin{pmatrix} 1 & 6 & 9 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 18 & 33 & 29 \\ 0 & 18 & 11 \\ 0 & 0 & 18 \end{pmatrix}^5 = \begin{pmatrix} 9 & 32 & 10 \\ 0 & 9 & 71 \\ 0 & 0 & 9 \end{pmatrix} \bmod 77. \tag{22}$$

At the other end user B chose their own random polynomial $h(x) = x^5+5x+1$. Evaluating the polynomial $h(a)$, the private key is as follows:

$$h(a) = \begin{pmatrix} 1 & 5 & 7 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}^5 + 5\begin{pmatrix} 1 & 5 & 7 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} + 1I = \begin{pmatrix} 7 & 50 & 39 \\ 0 & 7 & 40 \\ 0 & 0 & 7 \end{pmatrix} \bmod 77 \tag{23}$$

and the generation of public key for user B is as follows:

$$X_B = h(a)^m \cdot b \cdot h(a)^n = \begin{pmatrix} 7 & 50 & 39 \\ 0 & 7 & 40 \\ 0 & 0 & 7 \end{pmatrix}^3 \cdot \begin{pmatrix} 1 & 6 & 9 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 7 & 50 & 39 \\ 0 & 7 & 40 \\ 0 & 0 & 7 \end{pmatrix}^5 = \begin{pmatrix} 42 & 56 & 35 \\ 0 & 42 & 0 \\ 0 & 0 & 42 \end{pmatrix} \bmod 77. \tag{24}$$

Finally, the session key extracted by the user A as K_A is as follows:

$$K_A = f(a)^m \cdot X_B \cdot f(a)^n = \begin{pmatrix} 18 & 33 & 29 \\ 0 & 18 & 11 \\ 0 & 0 & 18 \end{pmatrix}^3 \cdot \begin{pmatrix} 42 & 56 & 35 \\ 0 & 42 & 0 \\ 0 & 0 & 42 \end{pmatrix} \cdot \begin{pmatrix} 18 & 33 & 29 \\ 0 & 18 & 11 \\ 0 & 0 & 18 \end{pmatrix}^5 = \begin{pmatrix} 70 & 42 & 28 \\ 0 & 70 & 0 \\ 0 & 0 & 70 \end{pmatrix} \bmod 77 \tag{25}$$

and the session key extracted by user B as K_B is as follows:

$$K_B = h(a)^m \cdot X_A \cdot h(a)^n = \begin{pmatrix} 7 & 50 & 39 \\ 0 & 7 & 40 \\ 0 & 0 & 7 \end{pmatrix}^3 \cdot \begin{pmatrix} 9 & 32 & 10 \\ 0 & 9 & 71 \\ 0 & 0 & 9 \end{pmatrix} \cdot \begin{pmatrix} 7 & 50 & 39 \\ 0 & 7 & 40 \\ 0 & 0 & 7 \end{pmatrix}^5 = \begin{pmatrix} 70 & 42 & 28 \\ 0 & 70 & 0 \\ 0 & 0 & 70 \end{pmatrix} \bmod 77. \tag{26}$$

4.3. Encryption-Decryption Algorithm on Heisenberg Group. The encryption-decryption procedure on Heisenberg group is offered as follows.

Encryption-Decryption Algorithm on Noncommutative Ring

Global Public Parameters

m, n: integers Z^+

a, b: ring elements

M: message

$H(M)$: hashed message

User A Key Generation

(i) Select private random polynomial: $f(x)$.

(ii) If $f(a) \neq 0$, then $f(a)$ is considered as private key.

(iii) Generation of public key X_A: $X_A = f(a)^m \cdot b \cdot f(a)^n$.

User B Key Generation

(i) Select private random polynomial: $h(x)$.

(ii) If $h(a) \neq 0$, then $h(a)$ is considered as private key.

(iii) Generation of public key X_B: $X_B = h(a)^m \cdot b \cdot h(a)^n$.

Encryption (by Sender B)

C: ciphertext

D: decryption key

$C = h(a)^m \cdot b \cdot h(a)^n$, $D = H(h(a)^m \cdot X_A \cdot h(a)^n) \oplus M$

Decryption

$M = H(f(a)^m \cdot C \cdot f(a)^n) \oplus D$

The approach of noncommutative cryptography works like the general case, where our assumptions are as

$$
\begin{aligned}
&m = 3,\\
&n = 5,\\
&a = \begin{pmatrix} 1 & 5 & 9 \\ 0 & 1 & 9 \\ 0 & 0 & 1 \end{pmatrix},\\
&b = \begin{pmatrix} 1 & 9 & 5 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix},\\
&N = 7 * 11,\\
&M = \begin{pmatrix} 27 & 19 & 25 \\ 34 & 8 & 7 \\ 45 & 5 & 9 \end{pmatrix}.
\end{aligned} \tag{27}
$$

User A randomly chose a random polynomial $f(x) = 3x^3 + 4x^2 + 5x + 6$; then $f(a)$ is considered to be private key:

$$
\begin{aligned}
f(a) &= 3\begin{pmatrix} 1 & 5 & 9 \\ 0 & 1 & 9 \\ 0 & 0 & 1 \end{pmatrix}^3 + 4\begin{pmatrix} 1 & 5 & 9 \\ 0 & 1 & 9 \\ 0 & 0 & 1 \end{pmatrix}^2\\
&\quad + 5\begin{pmatrix} 1 & 5 & 9 \\ 0 & 1 & 9 \\ 0 & 0 & 1 \end{pmatrix} + 6I\\
&= \begin{pmatrix} 18 & 33 & 13 \\ 0 & 18 & 44 \\ 0 & 0 & 18 \end{pmatrix} \bmod 77.
\end{aligned} \tag{28}
$$

The generation of public key is as follows:

$$
\begin{aligned}
X_A &= f(a)^m \cdot b \cdot f(a)^n\\
&= \begin{pmatrix} 18 & 33 & 13 \\ 0 & 18 & 44 \\ 0 & 0 & 18 \end{pmatrix}^3 \cdot \begin{pmatrix} 1 & 9 & 5 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}\\
&\quad \cdot \begin{pmatrix} 18 & 33 & 13 \\ 0 & 18 & 44 \\ 0 & 0 & 18 \end{pmatrix}^5 = \begin{pmatrix} 9 & 59 & 42 \\ 0 & 9 & 49 \\ 0 & 0 & 9 \end{pmatrix} \bmod 77.
\end{aligned} \tag{29}
$$

Moving onwards, user B randomly chose their own random polynomial $h(x) = x^5 + 5x + 1$ and computes private key if $h(a) \neq 0$:

$$
\begin{aligned}
h(a) &= \begin{pmatrix} 1 & 5 & 9 \\ 0 & 1 & 9 \\ 0 & 0 & 1 \end{pmatrix}^5 + 5\begin{pmatrix} 1 & 5 & 9 \\ 0 & 1 & 9 \\ 0 & 0 & 1 \end{pmatrix} + 1 \cdot I\\
&= \begin{pmatrix} 7 & 50 & 1 \\ 0 & 7 & 13 \\ 0 & 0 & 7 \end{pmatrix} \bmod 77
\end{aligned} \tag{30}
$$

and the public key generated for user B is as follows:

$$
\begin{aligned}
X_B &= h(a)^m \cdot b \cdot h(a)^n\\
&= \begin{pmatrix} 7 & 50 & 1 \\ 0 & 7 & 13 \\ 0 & 0 & 7 \end{pmatrix}^3 \cdot \begin{pmatrix} 1 & 9 & 5 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 7 & 50 & 1 \\ 0 & 7 & 13 \\ 0 & 0 & 7 \end{pmatrix}^5\\
&= \begin{pmatrix} 42 & 28 & 35 \\ 0 & 42 & 35 \\ 0 & 0 & 42 \end{pmatrix} \bmod 77.
\end{aligned} \tag{31}
$$

The sender of the public key is treated as ciphertext C (in our case user B is sender):

$$
\begin{aligned}
C &= h(a)^m \cdot b \cdot h(a)^n = \begin{pmatrix} 42 & 28 & 35 \\ 0 & 42 & 35 \\ 0 & 0 & 42 \end{pmatrix},\\
D &= H\left(h(a)^m \cdot X_A \cdot h(a)^n\right) \oplus M\\
&= H\left(\left(\begin{pmatrix} 7 & 50 & 1 \\ 0 & 7 & 13 \\ 0 & 0 & 7 \end{pmatrix}^3 \cdot \begin{pmatrix} 9 & 59 & 42 \\ 0 & 9 & 49 \\ 0 & 0 & 9 \end{pmatrix} \cdot \begin{pmatrix} 9 & 59 & 42 \\ 0 & 9 & 49 \\ 0 & 0 & 9 \end{pmatrix}^5\right) \oplus \begin{pmatrix} 27 & 19 & 25 \\ 34 & 8 & 7 \\ 45 & 5 & 9 \end{pmatrix}\right)\\
&= H\left(\begin{pmatrix} 70 & 21 & 35 \\ 0 & 70 & 7 \\ 0 & 0 & 70 \end{pmatrix}\right) \oplus \begin{pmatrix} 27 & 19 & 25 \\ 34 & 8 & 7 \\ 45 & 5 & 9 \end{pmatrix}\\
&= \begin{pmatrix} 12 & 42 & 57 \\ 35 & 31 & 52 \\ 44 & 4 & 30 \end{pmatrix}.
\end{aligned} \tag{32}
$$

The original message is as follows:

$$
\begin{aligned}
M' &= H\left(f(a)^m \cdot C \cdot f(a)^n\right) \oplus D \\
&= H\left(\left(\begin{pmatrix} 18 & 33 & 13 \\ 0 & 18 & 44 \\ 0 & 0 & 18 \end{pmatrix}^3 \cdot \begin{pmatrix} 42 & 28 & 35 \\ 0 & 42 & 35 \\ 0 & 0 & 42 \end{pmatrix}\right.\right. \\
&\quad \left.\left. \cdot \begin{pmatrix} 18 & 33 & 13 \\ 0 & 18 & 44 \\ 0 & 0 & 18 \end{pmatrix}^5\right) \oplus \begin{pmatrix} 12 & 42 & 57 \\ 35 & 31 & 52 \\ 44 & 4 & 30 \end{pmatrix} \\
&= \begin{pmatrix} 27 & 19 & 25 \\ 34 & 8 & 7 \\ 45 & 5 & 9 \end{pmatrix}.
\end{aligned} \tag{33}
$$

4.4. Analysis and Strength of Proposed Scheme. We are now explaining the computational hardness or complexity analysis with its related strength as security and performance considerations (mostly on each parameter of algorithms).

Prime Factors of N. The proposed procedure stands on hidden prime factorization of N (N is absent in the proposed algorithm, but due to explicit clarification it is shown wherever needed) being the points mentioned below in support of strong security analysis.

(i) Since $N = 7 * 11$ is based on two prime factors and factorization of N has extreme difficulty in finding its exact factors due its computer intensive nature for large primes, to find an algorithm which does it fast is one of unsolved problems of computer science.

(ii) The time required into prime factor grows exponentially, so if the algorithm uses large prime based integers, it is unrealistic to crack it down.

(iii) Prime factorization is mostly a unique problem, and all integers (except 1 and 0) are made up of primes, because this ingeniousness allows hardly encoding any information of any length as a single integer is inflexible.

Private Keys. A secret key generation is based on random chosen polynomial $f(x)$ or $h(x)$, since polynomial is called irreducible if and only if it cannot be expressed as product of two polynomials. An integer analogy of irreducible polynomial is called a prime polynomial. A polynomial contains the three classes of polynomials such as

(i) ordinary polynomial,

(ii) modulo prime based polynomial,

(iii) modulo prime based polynomial defined on another polynomial whose power is in some integer n.

In class (i) arithmetic operation (addition, subtraction, and multiplication) performs on polynomials using the ordinary rules of algebra, and division is only possible if the field elements are coefficients of the same. Class (ii) contains the arithmetic operations as of (i), but the division result is used (in general) in quotient and remainder forms. This represents a special significance in cryptography because it gives a unique solution for the above prime factorization on specified problem. Here class (iii) is not elaborated, because proposed approach is working on (i) and/or (ii).

Public Keys. The polynomial functions of $f(x)$ or $h(x)$ to the power of m and n with two multiplications on modulo prime are the basis for public key generation for respective senders and receivers. The generated public key (which passed into the medium) is represented as a Discrete Log Problem (DLP) for the algorithm since it is established on modulo prime based polynomials that are irreducible in these contexts. Adversaries try to conceptualize the secret key on the global parameters and public key; those are freely available. According to proposed scheme, a large prime factor of N (standard length 160 bits) may be sufficient for making adversaries against getting fruitful ideas or valid secret keys.

Timing Attack. Timing attacks is a stunning way abstracting the pattern generated from the cryptographic algorithm and trying to access the security appearance from electromagnetic signals released from the computer systems. The release of signals and transmissions are the part of computer operations. The signals are alarming in the two senses: (i) a random interference comes first, which can be only be burglars, and (ii) these signals can be amplified through some auxiliary equipment for some useful purposes. A report is available suggesting electromagnetic radiation interference with radio navigation devices, as (i) it is a general procedure and it is not a point to be considerable issue and (ii) it applies to those interested in such pattern of abstractions for decoding that may lead to vulnerable information safeties, feedbacks observations, and/or secret information leakage, where an adversary tries to determine the private key by keeping track on how long a computer takes to decipher the secrets messages.

In practice, polynomial modular exponentiation implementation does lead to extreme timing variations. Therefore, the uniqueness nature of polynomial based cryptographic measures makes a practical choice for future security work, with specific addition to noncommutative properties. Instead of the same, there are some countermeasures, which may lead to strengthening the measures in timing variation effects.

Polynomial Exponentiation Time. Since all exponentials take different amount of time before returning to final result, this one is simply fix, so performance analysis does not degrade its efficiency with its variances.

Random Delay. One can get a better performance by adding a random delay to the exponentials in applied algorithms and may confuse the timing attacks.

Blinding. It can confuse the adversary by multiplying a random number into the ciphertext before performing exponentiation. This can be one way to make adversaries outreach from original ciphers.

Brute-Force Attacks. The brute-force attacks refer to finding all the possible secret keys. The defense against attacks shows larger randomness and unpredictability behavior on a shorter key length on our proposed approach; since it is a special case of Elliptic Curve Cryptography (ECC), therefore the algorithm sufficiently works on a smaller length keys. The execution time of smaller length key takes a shorter time, so it reflects a big impact on efficiency. A lot of reports are available regarding the computational performance for ECC and RSA algorithms, where our approach (noncommutative) regarding the efficiency, speed, and cryptanalysis is better in them.

Chosen Ciphertext Attacks. This attack is a form of active attack, where adversaries try to find plaintext corresponding to its ciphertexts by its choice. The first choice may experience decryption module on a random chosen ciphertext, before the actual ciphertext sent for an interested use. The second choice involves the same module on input of one's choice at any time, where all these are recorded and try to gain the actual plaintexts. The presented algorithm experienced a blind feedback, where the noncommutative cryptography is not a vulnerable one to chosen ciphertext attacks (CCA) especially for ring or semiring, group, and Heisenberg elements, because in CCA an adversary chooses a number of ciphertexts and tries decryption with targeted private keys, where the chosen ciphertext is hashed with the corresponding polynomial exponentials.

Simulation and Importance of Hash Uses. The simulation of hash H is based on power of 2 functions on Matlab tool, where the importance of hash function dictates the following properties: (i) output of hash generates pseudorandomness for the standard cryptographic tests, (ii) hash is relatively easy to compute for any given input that makes a practical use for hardware and software implementation, (iii) for any given hash H, it is computationally infeasible to find y such that $H(y) = x$, (iv) for any pair (x, y) it is computationally infeasible to find $H(x) = H(y)$ is a strong collision resistant property, and (v) for any block x, it is computationally infeasible to find $y \neq x$ is a weak collision resistant property.

5. Monomials Based Cryptography Using Noncommutative Groups and Semirings

The polynomial used in $\mathbb{Z}$-modular method for noncommutative cryptography is based on the group elements running at the back end and its equivalent semirings elements work from the front, known by the monomials generated schemes. In this regard the original information is hidden, where for an adversary it will be practically impossible to decipher the original information. Such kind of participation in computation is viewed as a special case. We have formulated the semiring elements that are working perfectly under the assumptions of our dihedral order 8, which is a part of extra special group. The section is first exploring the basic assumptions on monomials and then proposed works are detailed.

5.1. Extension of Noncommutative Groups. For a noncommutative group $(G, \cdot, 1_G)$ and ring elements $(R, +, \cdot, 1_R)$, monomials generations that are possible through the use of group and ring elements can be defined as τ: $(G, \cdot, 1_G) \to (R, \cdot, 1_R)$. The inverse map τ^{-1}: $\tau(G) \to G$ is also a well-defined monomial. If $a, b \in G$, then it is true for $\tau(a)+\tau(b) \in \tau(G)$; from the same one can assign a new element $c \in G$ as $c \triangleq \tau^{-1}(\tau(a) + \tau(b))$ which is possible. Here c is called as a quasi-sum of a and b and is denoted by $c = a \boxplus b$ [31]. Similarly, for $k \in R$ and $a \in G$, if $k \cdot \tau(a) \in \tau(G)$, then one can assign a new element $d \in G$ as $d \cong \tau^{-1}(k \cdot \tau(a))$ and call d as quasi-multiple of a, denoted by $d = k \boxtimes a$. Then, the monomial τ in a linear sense holds the following equalities:

$$\begin{aligned}
\tau(k \boxtimes a \boxplus b) &= \tau((k \boxtimes a) \boxplus b) = \tau(d \boxplus b) \\
&= \tau\left(\tau^{-1}(\tau(d))\right) \boxplus \tau(b) \\
&= \tau\left(\tau^{-1}\left(\tau\left(\tau^{-1}(k \cdot \tau(a))\right)\right)\right) \boxplus \tau(b) \\
&= \tau\left(\tau^{-1}(k \cdot \tau(a))\right) \boxplus \tau(b) \\
&= k \cdot \tau(a) + \tau(b).
\end{aligned} \tag{34}$$

For $a, b \in G$ and $k \cdot \tau(a) + \tau(b) \in \tau(G)$, function $f(x) = z_0 + z_1 x + \cdots + z_n x^n \in Z[x]$ can be defined as

$$f(\tau(a)) = z_0 \cdot 1_R + z_1 \cdot \tau(a) + \cdots z_n \cdot \tau(a)^n \in \tau(G). \tag{35}$$

Now, assign a new element $e \in G$ as

$$\begin{aligned}
e &= \tau^{-1}(f(a)) \\
&= \tau^{-1}\left(z_0 \cdot 1_R + z_1 \cdot \tau(a) + \cdots z_n \cdot \tau(a)^n\right).
\end{aligned} \tag{36}$$

If it holds to find the inverse of polynomials, call e as quasi-polynomial of f on a, denoted by $e = f(a)$. For, arbitrary $a, b \in G$, $k \in R$ and $f(x) \in Z[x]$, $a \boxplus b$, $k \boxtimes a$, and $f(a)$ are not always well defined. Theorem 7 is natural and general scheme, which works for noncommutative monomials.

Theorem 7. *For some $a \in G$ and some $f(x), h(x) \in Z[x]$, if $f(a)$ and $h(a)$ are well defined, then (i) $\tau(f(a)) = f(\tau(a))$ and (ii) $f(a) \cdot h(a) = h(a) \cdot f(a)$.*

Proof. (i) Due to property of monomials on quasi-polynomial, the group element for any function f applies with its equivalent ring elements, so in the intermediary function f results in ring or semiring R and is observed on numerical analysis; it results in the same elements of ring R. It can also be validated on LHS and RHS consideration.

LHS

$$\begin{aligned}\tau\left(f\left(a\right)\right) &= \tau\left(G\right) \quad \left(\because f\left(a\right) = G, \text{ is group elements}\right)\\ &= R \quad \left(\because \tau \text{ is a monomial, so } \tau\left(G\right) \to R, \text{ since } R \text{ is inverse of Group}\right)\end{aligned} \tag{37}$$

RHS

$$\begin{aligned}f\left(\tau\left(\mathrm{a}\right)\right) &= f\left(R\right) \quad \left(\because \tau\left(a\right) = R, \text{ is ring elements}\right)\\ &= R \quad \left(\because f\left(R\right) \text{ generates ring elements}\right).\end{aligned} \tag{38}$$

(ii)

$$\begin{aligned}f\left(a\right)\cdot h\left(a\right) &= \tau\left(\tau^{-1}\left(f\left(a\right)\right)\right)\cdot\tau\left(\tau^{-1}\left(h\left(a\right)\right)\right)\\ &\qquad \left(\because \tau\left(\tau^{-1}\left(g\right)\right) = g, \ g \in G\right)\\ &= \tau\left(\tau^{-1}\left(f\left(a\right)\right)\cdot\tau^{-1}\left(h\left(a\right)\right)\right)\\ &\qquad \left(\because \tau \text{ is a monomial}\right)\\ &= \tau\left(\tau^{-1}\left(f\left(a\right)\cdot h\left(a\right)\right)\right)\\ &\qquad \left(\because \tau^{-1} \text{ is monomial}\right)\\ &= \tau\left(\tau^{-1}\left(h\left(a\right)\cdot f\left(a\right)\right)\right)\\ &\qquad \left(\because \text{Theorem 2}\right)\\ &= \tau\left(\tau^{-1}\left(h\left(a\right)\right)\cdot\tau^{-1}\left(f\left(a\right)\right)\right)\\ &= \tau\left(\tau^{-1}\left(h\left(a\right)\right)\right)\cdot\tau\left(\tau^{-1}\left(f\left(a\right)\right)\right)\\ &= h\left(a\right)\cdot f\left(a\right).\end{aligned} \tag{39}$$

□

5.2. Further Assumptions on Noncommutative Groups. Consider the assumption on polynomial version over the noncommutative group, for any randomly picked-up element of $a \in G$ and to define a polynomial set $P_a \in G$ by: $P_a \cong \{f(a) \in \tau(G) : f(x) \in Z[x]\}$.

Then, the definition on group G over $(G, \cdot)$ says the following.

(i) *Polynomial Symmetrical Decomposition (PSD) Problems over Noncommutative Group G.* Given $(a, x, y) \in G^3$ and $m, n \in Z$, find $z \in P_a$ such that $y = z^m \cdot x \cdot z^n$.

(ii) *Polynomial Diffie-Hellman (PDH) Problems over Noncommutative Group G.* Compute $x^{z_1 z_2}$ or $x^{z_2 z_1}$ for given a, x, x^{z_1} and x^{z_2}, where $a, x \in G$ and $z_1, z_2 \in P_a$.

The PSD or PDH on cryptographic assumptions over $(G, \cdot)$ is intractable, and there is not any probabilistic polynomial time algorithm subsisting to solve this problem with accurateness admiration [31].

Theorem 8. *The generalized extra special p-group over the monomials is free from attacks.*

Proof. Suppose the group on G with $(G, \cdot, 1_G)$ is a noncommutative group and semiring R on $(R, \cdot, 1_R)$ is semiring and its monomials are defined as $\tau : (G, \cdot, 1_G) \to (R, \cdot, 1_R)$, such that the group elements are always working at the back end, and computation is only defined on the monomials semiring elements. In this regard, the original extent of the algorithm is always hidden. The working of this prime p-group is an example of hidden subgroup or subfield problem. Hence, the theorem proves the generalized extra special p-group over the monomials is free from attacks. □

5.3. Monomials Like Key Exchange Algorithm. The global parameters of the proposed algorithm, at dihedral order 8, for key exchange using monomials are presented as follows.

Monomials Key Exchange on Noncommutative Ring

Global Public Parameters

m, n: integers Z^+

a, b: group elements from ring

Supposing $(G, \cdot, 1_G)$ is a noncommutative group, $(R, \cdot, 1_R)$ is ring, and $\tau : (G, \cdot, 1_G) \to (R, \cdot, 1_R)$ is monomorphism

User A Key Generation

(i) $f(x)$ is random polynomial chosen by A.

(ii) Select $f(x) \in Z(x)$ at random so that $f(a)$ is well defined; that is, $f(\tau(a)) \in \tau(G)$; then user A takes $f(a)$ as private key X_A: $X_A = f(a)^m \cdot b \cdot f(a)^n$.

User B Key Generation

(i) $h(a)$ is random polynomial chosen by B.

(ii) Select $h(x) \in Z(x)$ at random so that $h(a)$ is well defined; that is, $h(\tau(a)) \in \tau(G)$; then user B takes $h(a)$ as private key X_B: $X_B = h(a)^m \cdot b \cdot h(a)^n$.

Generation of Secret Shared Session Key by User A

$$K_A = f(a)^m \cdot X_B \cdot f(a)^n.$$

Generation of Secret Shared Session Key by User B

$$K_B = h(a)^m \cdot X_A \cdot h(a)^n,$$

where our assumptions are as follows:

$$\begin{aligned} m &= 16, \\ n &= 55, \\ a &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}, \\ b &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix} \end{aligned} \tag{40}$$

and the relative monomials of group elements $\tau : (G, \cdot, 1_G) \to (R, \cdot, 1_R)$ are represented below, respectively. At dihedral group of order 8 there is a possibility of 8 groups; we assumed the same from G_1 to G_8 and its corresponding ring monomials from R_1 to R_8. Such group and ring elements are assigned in sequence as $G_1 \to R_1$, $G_2 \to R_2$, $G_3 \to R_3$, $G_4 \to R_4$, $G_5 \to R_5$, $G_6 \to R_6$, $G_7 \to R_7$, and $G_8 \to R_8$. These all will be used in cryptographic primitives.

$$\begin{aligned} G_1 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \\ G_2 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}, \\ G_3 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}, \\ G_4 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}, \\ G_5 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}, \\ G_6 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}, \\ G_7 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}, \\ G_8 &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}, \\ R_1 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\ R_2 &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \\ R_3 &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \\ R_4 &= \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \\ R_5 &= \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}, \\ R_6 &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \\ R_7 &= \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \\ R_8 &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \end{aligned} \tag{41}$$

In the computation process, ring elements are generated on negative modulo prime, where Lemma 9 is well distinct.

Lemma 9. *The variability generation for equivalent monomials ring structured elements in the range of* $\{-1, 0, 1\}$ *is negative modulo prime of* (-2).

Proof. By inspection, it is observed that the negative modulo prime on (-2) results in the variations of $\{-1, 0, 1\}$, which well suits equivalence in the monomials like generation elements to our proposed scheme of dihedral order of 8 (where dihedral order 8 is specially a part of extra special group).

The user A chooses a random polynomial $f(x) = 4x^2 + x + 2$ and on $f(a) \neq 0$ the secret/private key elected for user A as

$$\begin{aligned} f(a) &= \tau^{-1}\left(f(\tau(a))\right) \\ &= \tau^{-1}\left(4\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^2 + \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} + 2\right) \\ &= \tau^{-1}\left(\left(\begin{pmatrix} -2 & 3 \\ 1 & -2 \end{pmatrix} \bmod (-2)\right) \quad [\because \text{Lemma 9}]\right. \\ &= \tau^{-1}\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \xrightarrow{R_5 \to G_5} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}. \end{aligned} \tag{42}$$

The generation of public key X_A is as follows:

$$\begin{aligned} X_A &= f(a)^m \cdot b \cdot f(a)^n \\ &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}^{16} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}^{55} \\ &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}. \end{aligned} \tag{43}$$

Further, a random polynomial is chosen by user B as $h(x) = 3x^4 + x^3 + 4x^2 + 3x + 4$ and computes private key:

$$\begin{aligned} h(a) &= \tau^{-1}(h(\tau(A))) \\ &= \tau^{-1}\left(3\begin{pmatrix}0&1\\-1&0\end{pmatrix}^4 + \begin{pmatrix}0&1\\-1&0\end{pmatrix}^3 + 4\begin{pmatrix}0&1\\-1&0\end{pmatrix}^2 \right. \\ &\quad \left. + 3\begin{pmatrix}0&1\\-1&0\end{pmatrix} + 4\right) \\ &= \tau^{-1}\left(\begin{pmatrix}3&6\\2&3\end{pmatrix} \bmod (-2)\right) \quad [\because \text{Lemma 9}] \\ &= \tau^{-1}\begin{pmatrix}-1&0\\0&-1\end{pmatrix} \xrightarrow{R_7 \to G_7} \begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix} \end{aligned} \tag{44}$$

and generation of public key for user B afterwards sends it to user A:

$$\begin{aligned} X_B &= h(a)^m \cdot b \cdot h(a)^n \\ &= \begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}^{16} \cdot \begin{pmatrix}1&2&3&4\\1&4&3&2\end{pmatrix} \cdot \begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}^{55} \\ &= \begin{pmatrix}1&2&3&4\\2&3&4&1\end{pmatrix}. \end{aligned} \tag{45}$$

Now, user A extracts the session key as

$$\begin{aligned} K_A &= f(a)^m \cdot X_B \cdot f(a)^n \\ &= \begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}^{16} \cdot \begin{pmatrix}1&2&3&4\\2&3&4&1\end{pmatrix} \cdot \begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}^{55} \\ &= \begin{pmatrix}1&2&3&4\\4&1&2&3\end{pmatrix} \end{aligned} \tag{46}$$

and user B extracts the session key as

$$\begin{aligned} K_B &= h(a)^m \cdot X_A \cdot h(a)^n \\ &= \begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}^{16} \cdot \begin{pmatrix}1&2&3&4\\3&2&1&4\end{pmatrix} \cdot \begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}^{55} \\ &= \begin{pmatrix}1&2&3&4\\4&1&2&3\end{pmatrix}. \end{aligned} \tag{47}$$

□

5.4. Monomials Like Encryption-Decryption Algorithm on Noncommutative Cryptography. The way for encryption and decryption module for monomials algorithm is presented at dihedral order 8, as follows on step-by-step procedure (as carried out below).

Monomials Based Noncommutative Algorithm for Encryption-Decryption

Global Public Parameters

m, n: integers Z^+

a, b: group elements from ring

p, q: secure primes

g: generator function

M: message

Supposing $(G, \cdot, 1_G)$ is a noncommutative group, $(R, \cdot, 1_R)$ is ring, and $\tau : (G, \cdot, 1_G) \to (R, \cdot, 1_R)$ is monomorphism

User A Key Generation

(i) $f(x)$ is random polynomial chosen by A.

(ii) Select $f(x) \in Z(x)$ at random so that $f(a)$ is well defined; that is, $f(\tau(a)) \in \tau(G)$; then user A takes $f(a)$ as private key X_A: $X_A = f(a)^m \cdot b \cdot f(a)^n$.

User B Key Generation

(i) $h(a)$ is random polynomial chosen by B.

(ii) Select $h(x) \in Z(x)$ at random so that $h(a)$ is well defined; that is, $h(\tau(a)) \in \tau(G)$; then user B takes $h(a)$ as private keyX_B: $X_B = h(a)^m \cdot b \cdot h(a)^n$.

Encryption (User B)

Ciphertext: C

Decryption key: D

C: sender public key

D: $H(h(a)^m \cdot X_A \cdot h(a)^n) \oplus M$

Decryption (User A)

Original message M': $H(h(a)^m \cdot C \cdot h(a)^n) \oplus M$

The algorithm is presented on two random primes p and q, such that $q \mid p-1 \neq 0$, and generator function g is an order of q and message M. The numerical dictation is elaborating here, where global assumptions are

$$\begin{aligned} m &= 12, \\ n &= 19, \\ a &= \begin{pmatrix}1&2&3&4\\3&2&1&4\end{pmatrix}, \\ b &= \begin{pmatrix}1&2&3&4\\1&4&3&2\end{pmatrix}, \\ p &= 23, \\ q &= 11, \\ g &= 6, \\ M &= 17. \end{aligned} \tag{48}$$

The random polynomial $f(x) = 2x^5 - 5x^2 + 3$ is chosen by user A; the private key is as follows:

$$\begin{aligned} f(a) &= \tau^{-1}\left(f(\tau(a))\right) \\ &= \tau^{-1}\left(2\cdot\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}^5 - 5\cdot\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}^2 + 3\right) \\ &= \tau^{-1}\left(\begin{pmatrix}8 & 1\\ 5 & 8\end{pmatrix} \bmod (-2)\right) \quad [\because \text{Lemma 9}] \\ &= \tau^{-1}\begin{pmatrix}0 & -1\\ -1 & 0\end{pmatrix} \xrightarrow{R_5 \to G_5} \begin{pmatrix}1 & 2 & 3 & 4\\ 4 & 3 & 2 & 1\end{pmatrix}. \end{aligned} \tag{49}$$

The public key generated as X_A is as follows:

$$\begin{aligned} X_A &= f(a)^m \cdot b \cdot f(a)^n \\ &= \begin{pmatrix}1 & 2 & 3 & 4\\ 4 & 3 & 2 & 1\end{pmatrix}^{12} \cdot \begin{pmatrix}1 & 2 & 3 & 4\\ 1 & 4 & 3 & 2\end{pmatrix} \cdot \begin{pmatrix}1 & 2 & 3 & 4\\ 4 & 3 & 2 & 1\end{pmatrix}^{19} \\ &= \begin{pmatrix}1 & 2 & 3 & 4\\ 3 & 2 & 1 & 4\end{pmatrix}. \end{aligned} \tag{50}$$

Moving ahead, user B chose their own random polynomial $h(x) = 9x^4 + x^3 + 4x^2 + 9x + 4$ and computes private key as

$$\begin{aligned} h(a) &= \tau^{-1}\left(h(\tau(A))\right) \\ &= \tau^{-1}\left(9\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}^4 + \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}^3 + 4\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}^2 + 9\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} + 4\right) \\ &= \tau^{-1}\left(\begin{pmatrix}9 & -4\\ 12 & 9\end{pmatrix} \bmod (-2)\right) \quad [\because \text{Lemma 9}] \\ &= \tau^{-1}\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix} \xrightarrow{R_7 \to G_7} \begin{pmatrix}1 & 2 & 3 & 4\\ 2 & 1 & 4 & 3\end{pmatrix}. \end{aligned} \tag{51}$$

The public key generation for user B as X_B is as follows:

$$\begin{aligned} X_B &= h(a)^m \cdot b \cdot h(a)^n \\ &= \begin{pmatrix}1 & 2 & 3 & 4\\ 2 & 1 & 4 & 3\end{pmatrix}^{12} \cdot \begin{pmatrix}1 & 2 & 3 & 4\\ 1 & 4 & 3 & 2\end{pmatrix} \cdot \begin{pmatrix}1 & 2 & 3 & 4\\ 2 & 1 & 4 & 3\end{pmatrix}^{19} \\ &= \begin{pmatrix}1 & 2 & 3 & 4\\ 2 & 3 & 4 & 1\end{pmatrix}. \end{aligned} \tag{52}$$

In the next step, we need to use a hash function, where we are exploring the same through Lemma 10.

Lemma 10. *Then hash function is defined on*

$$H: \begin{pmatrix}1 & 2 & 3 & 4\\ \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4\end{pmatrix} \longrightarrow \left(g^{2^0\cdot\sigma_1 + 2^1\cdot\sigma_2 + 2^2\cdot\sigma_3 + 2^3\cdot\sigma_4}\right) \bmod p. \tag{53}$$

Proof. By hypothesis for hash H assumed by Cao et al. 2007 [32], which is based on dihedral order of 6 for hash $H : \left(\begin{smallmatrix}1 & 2 & 3\\ \sigma_1 & \sigma_2 & \sigma_3\end{smallmatrix}\right)$, in the present work, as a contribution, the authenticity of hash is preserved by applying to dihedral order of 8 (a part of extra special group) without hampering the original concepts.

Suppose user B is sender, then, according to our proposed algorithm its public key is treated as our ciphertext. The decryption key D is assigned as

$$\begin{aligned} D &= H\left(h(x)^m \cdot X_A \cdot h(x)^n\right) \oplus M \\ &= H\left(\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}^{12} \cdot \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} \cdot \begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}^{19}\right) \oplus 17 \\ &= H\left(\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}\right) \oplus 17 \\ &\xrightarrow{R_2 \to G_2} \left(\left(g^{2^0\cdot 4 + 2^1\cdot 1 + 2^2\cdot 2 + 2^3\cdot 3}\right) \bmod p\right) \oplus 17 \\ &\qquad [\because \text{Lemma 10}] \\ &= \left(\left(6^{2^0\cdot 4 + 2^1\cdot 1 + 2^2\cdot 2 + 2^3\cdot 3}\right) \bmod 23\right) \oplus 17 \\ &= \left(\left(6^{38}\right) \bmod 23\right) \oplus 17 = 6 \oplus 17 = 23. \end{aligned} \tag{54}$$

Now, the receiver A decrypts the encrypted message as:

$$\begin{aligned} &= H\left(f(x)^m \cdot \text{Cipher} \cdot f(x)^n\right) \oplus D \\ &= H\left(\begin{pmatrix}0 & -1\\ -1 & 0\end{pmatrix}^{12} \cdot \begin{pmatrix}-1 & 0\\ 0 & 1\end{pmatrix} \cdot \begin{pmatrix}0 & -1\\ -1 & 0\end{pmatrix}^{19}\right) \oplus 23 \\ &= H\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix} \oplus 23 \\ &\xrightarrow{R_2 \to G_2} \left(\left(g^{2^0\cdot 4 + 2^1\cdot 1 + 2^2\cdot 2 + 2^3\cdot 3}\right) \bmod p\right) \oplus 23 \\ &\qquad [\because \text{Lemma 10}] \\ &= \left(\left(6^{2^0\cdot 4 + 2^1\cdot 1 + 2^2\cdot 2 + 2^3\cdot 3}\right) \bmod 23\right) \oplus 23 \\ &= \left(\left(6^{38}\right) \bmod 23\right) \oplus 23 = 6 \oplus 23 = 17. \end{aligned} \tag{55}$$

□

5.5. Security Analysis on Monomials. The security strength analysis as presented in Section 4 for general structure schemes also works in a similar fashion for monomials structures like schemes. The following factors in correlation

with the same play a crucial role for the cryptographic schemes generation such as the following: (i) One is generation of equivalent monomials ring elements on negative modulo prime behaving like semiring elements and these are considered as a natural generalization of noncommutative scheme in the sense that the binary addition and multiplication operations are not required to be commutative. The semigroup action suggests the exponential growth on key, which does not make any chance to find the solution. (ii) In hash generation (as Lemma 10) prime factorization of P keeps all the similar analysis results for cryptographic existence as presented in previous section. (iii) As the generation of private keys and public keys is based on monomials like structured elements, where the working scheme is initiated on monomials semiring elements for equivalent group elements, this means the original information of the group elements is totally in hidden form. The original elements are used for verification purposes only in proposed work. The generated discrete log value does not keep any significant information for adversaries. (iv) DLP provides a big conflict of interests on randomness and unpredictability generation for secret keys that also maintains a balance between key sizes and security extents. Therefore, with regard to the above-mentioned points, brute-force attacks and chosen ciphertext attacks are extremely resistant against the proposed scheme.

5.6. Efficiency Issues on General and Monomials Noncommutative Schemes. For noncommutative monoid, $F_{a^t}{}^{(b)} = a^t \cdot b \cdot \overline{(a^t)}$, it first computes a^t and then its inversion $\overline{(a^t)}$ and finally takes two multiplications in the underlying implementation. Here t represents either m or n for the polynomial function F. When t is considered to be a big digit, the computer arithmetic successively does doubling, rather than multiplying "a" for t times, so in this case the performance evaluation takes $O(\log_2(t))$ times to complete the task. In the present scenario 160-bit long t is enough to resist exhaustive attacks. The assumptions apply on length of group elements G (here proposed extra special group with one of the latest and longest group lengths) such as $a, b, a^t, F_{a^t}{}^{(b)}$ being a polynomial as a system security parameter, where the results are generated using the conventional (bit-by-bit) operations.

Moreover, for the secure and efficient architecture of the group elements, it represents the following facts regarding the same:

(i) Using the above described representation of group G element is unique. Otherwise the scheme (proposed) cannot work.

(ii) The transition from group G elements to its equivalent ring elements finishes efficiently. Otherwise, the scheme is impractical.

(iii) $F_{a^t}{}^{(b)}$ does not reveal any information regarding polynomial a^t. Otherwise, the proposed assumptions (in algorithm) can suffer from the length based attacks.

6. Basic Length Based Attacks

It is a heuristic procedure for finding the recipient's secret keys and is representing one of the procedures for recovering each of the conjugating factors as a major goal. The successful procedure results in actual conjugator as a product of group elements. The length based attack [45, 46] on dihedral order 6 is presented in [38]. Our proposed approach is based on dihedral order 8, that is, $k = 4$; a number of elements play an enormous role of generation of a subset of 8 group elements defined as $S_G = \{g_1^{\pm 1}, g_2^{\pm 1}, g_3^{\pm 1}, g_4^{\pm 1}\}$. We consider a random input series $y = g_1 g_2^{-1} g_3 g_4^{-1}$, for length $n = 4$. On choosing input sequence, the operation performs on $2k$-ary tree. It begins with an empty word e and searches for one of the child nodes of 8 group elements, with successful generation of 8 individual groups. For each element presented in input sequence is traced on successful generation, this procedure repeats until some n input of y_n length chosen for $y = y_1 y_2 \cdots y_n$ satisfies, as shown in Figure 5. This is based on the nth level that contains $(2k)^n$ leaf-nodes elements. Each leaf-node is likely to be a potential value for y. The fact behind solving the CDP is easy but the fact to solve the CSP is unavailable, so it can be considered to be secure against the brute-force search.

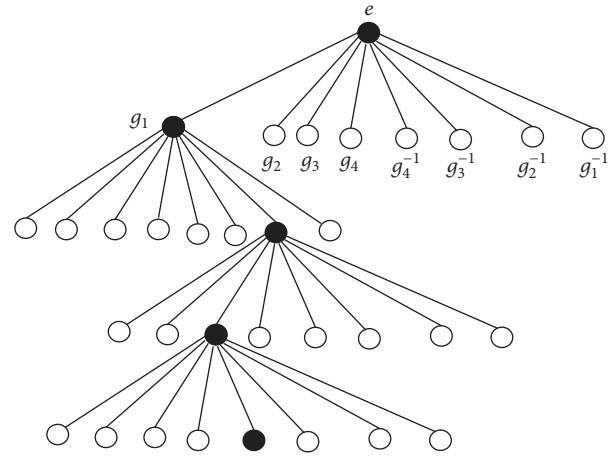

Figure 5: Process of generating $y = g_1 g_2^{-1} g_3 g_4^{-1}$.

For example, during the process of searching, if there are two children-nodes P and Q with equal length and if the algorithm wrongly predicts any node in one of them, it makes the algorithm fall into an exponential search in the worst case for negligible solution. On average, 8 candidates (in dihedral order 8) nodes in each level represent the time complexity of attack algorithm on $O(8^{2n})$ for all n word per each level length, on the success or failure attempts.

The process of attack is reversed, for instance, searching for the $2k$-ary tree. This means attack is a reversal procedure, which works on successful cryptanalysis; at first level it should satisfy 64 elements from 8 groups; similarly for the second level, it should again satisfy the same, and it should repeat to word length input. An example is dictated on the target nodes represented as a dark node that forms the optimal search path (as shown in Figure 6). The best result of an attack algorithm is to find this path which indicates decomposition of y.

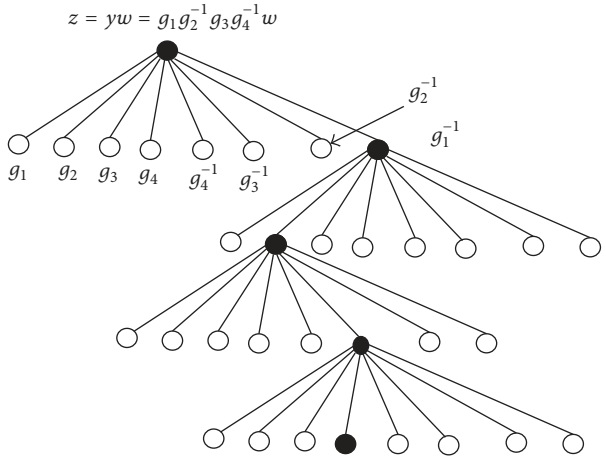

Figure 6: Process of decomposing $y = g_1 g_2^{-1} g_3 g_4^{-1}$.

6.1. Analysis on Length Based Attacks. Dihedral order of 8 uses 4 (four) elements to form one group of 8 elements each (a total of 8 groups' formation with a total of 64 elements), so adversaries (attackers) try to obtain the input sequence on level by level; at the first level an adversary needs to satisfy 64 corresponding elements and again for the second level needs to satisfy the next 64 corresponding elements and it will continue to repeat until length does not reach to maximum length. So, one can represent its complexity in the worst case being $O(8^{2n})$. Here Conjugacy Decisional Problem (CDP) is easy to apply according to algorithm for the same, but Conjugacy Search Problem (CSP) does not work correctly. The CDP and CSP are the two advantageous approaches for making the exponential impulsiveness and arbitrariness for nonnegligible solution. Here no such Conjugacy Search Problem exists to solve such larger scaled problems. Therefore, in the concluding remarks it can be mention that our proposed problem is making a big impact with regard to cryptographic schemes.

Further, the performance may also improve using the following artillery variations by using memory uses, repetitions avoidance, look-ahead, alternative solutions, and automorphism attacks implementation into the algorithms. A brief idea is presented below.

Memory Uses. To compute the $2kn$-child nodes from the N subtrees, the width of search memory increases which chooses the shortest N ones as the roots of the subtrees for the next computations. As a result, the search width is enhanced from 1 to N. According to principle the algorithm becomes efficient, but it is still exponential. For example, if in any loop the right node is not included in the M candidates, then the algorithm may degenerate to exponential. If a random input y_j is chosen from an adversary that multiplies for a word (or key) length w_j, the necessary and sufficient condition may be incorporation on $l(y_j w_{j+1}) < l(w_{j+1})$, where $w_{j+1} = y_{j+1}$ to y_n, and y_j left multiplying to w_{j+1} forms its length reduction. So, it can be considered, as corrected node is not included in the N candidates list, because length of child-node increases, which happens incorrectly, and therefore perceptibly on successful attacks (rate) will decrease.

Repetitions Avoidance. Avoiding repetition is an improvement usually seen in research. The algorithm maintains a hash table of recorded values of visited nodes, and the chance may be that the two nodes in the search tree have the same value. If the same node value appears again, then from the candidate list node value will be cancelled for same. So, to improve the algorithm, avoiding repetition method used in list not only improves efficiency but also prevents the algorithm from trapping into a closed loop.

Look-Ahead. This increases depth of searching. The original algorithm searches one level each time, while the method searches n levels each time; here is a possible way to make a practical choice to avoid attacks. For better algorithm this is one of the promising optional problems; the cost of traversing time at n-level subtree is $(2k)^n$. The algorithm increases in length of multiple right nodes for n-steps; the look-ahead problem never finds the right node. Thus, the algorithm again falls in exponential search.

Alternative Solutions. The alternative solution is a specific one, generated on the monomials like cryptographic schemes. This type of improvement is much more efficient, and in addition it does not change the search complexity. It helps to infinitely decrease the search time to find the right nodes. But one of the basic conditions involved with this algorithm is that search direction should be correct. A large possibility is available to make the algorithm fall into an exponential search, if once it enters into a wrong subtree.

Automorphism Attacks. Under the parameters selected, the random automorphism functions are extremely allied to each other on random polynomial chosen for participants, so the generation of these values is considered to be negligible.

7. Conclusion

In the manuscript, the noncommutative cryptographic scheme on the extra special group for the multidisciplinary perspective has been considered. Regarding this the minimum group of the dihedral changes from D_3 to D_4 enhances the search space, and two additional group benefits of Heisenberg and quaternion groups make the proposal stronger than all the previously predicted groups. The scheme is processed at the noncommutative platform for the prospective advantages of typical sparse matrices for general structures like group, ring, or semiring elements. The proposed security assumptions are based on the hidden subgroup or subfields problem (HSP) on the random polynomials chosen for end users, and monomials generation is presented where Conjugacy Search Problem (CSP) is likely to be intractable. For the adversary, the attacks like length based, brute-force, automorphism, are being negligible.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transaction on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[3] V. S. Miller, "Use of elliptic curves in cryptography," in *Proceedings of the ACM Advances in Cryptology (CRYPTO '85)*, pp. 417–426, 1985.

[4] W. S. Peter, "Algorithms for quantum computation: discrete logarithms and factorings," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.

[5] A. Kitaev, "Quantum Measurements and the Abelian Stabilizer Problem. Electronic Colloquium on Computational Complexity," Vol. 3, 1996, http://eccc.hpi-web.de/eccc-reports/1996/TR96-003/index.html.

[6] E. Lee, "Braid groups in Cryptology," *ICICE Transactions on Fundamentals*, vol. 87, no. 5, pp. 986–992, 2004.

[7] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," *Quantum Information and Computation*, vol. 3, no. 4, pp. 317–344, 2003.

[8] M. Rotteler, "Quantum algorithm: a survey of some recent results," *Information Forensic Entwistle*, vol. 21, pp. 3–20, 2006.

[9] N. R. Wagner and M. R. Magyarik, "A public-key cryptosystem based on the word problem," in *Advances in Cryptology—Proceedings of CRYPTO 84*, G. R. Blakley and D. Chaum, Eds., vol. 196 of *Lecture Notes in Computer Science*, pp. 19–36, Springer, Berlin, Germany, 1985.

[10] J.-C. Birget, S. S. Magliveras, and M. Sramka, "On public-key cryptosystems based on combinatorial group theory," *Tatra Mountains Mathematical Publications*, vol. 33, pp. 137–148, 2006.

[11] I. Anshel, M. Anshel, and D. Goldfeld, "An algebraic method for public-key cryptography," *Mathematical Research Letters*, vol. 6, no. 3, pp. 287–291, 1999.

[12] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J.-S. Kang, and C. Park, "New public-key cryptosystem using braid groups," in *CRYPTO 2000*, M. Bellare, Ed., vol. 1880 of *Lecture Notes in Computer Science*, pp. 166–183, Springer, Berlin, Germany, 2000.

[13] P. Dehornoy, "Braid-based cryptography," *Contemporary Mathematics*, vol. 360, pp. 5–33, 2004.

[14] I. Anshel, M. Anshel, and D. Goldfeld, "Non-abelian key agreement protocols," *Discrete Applied Mathematics*, vol. 130, no. 1, pp. 3–12, 2003.

[15] I. Anshel, M. Anshel, and D. Goldfeld, "A linear time matrix key agreement protocol over small finite fields," *Applicable Algebra in Engineering, Communication and Computing*, vol. 17, no. 3, pp. 195–203, 2006.

[16] J. C. Cha, K. H. Ko, S. J. Lee, J. W. Han, and J. H. Cheon, "An efficient implementation of braid groups," in *Advances in Cryptology—ASIACRYPT 2001*, C. Boyd, Ed., vol. 2248 of *Lecture Notes in Computer Science*, pp. 144–156, Springer, Berlin, Germany, 2001.

[17] K. H. Ko, D. H. Choi, M. S. Cho, and J.-W. Lee, "New signature scheme using conjugacy problem," Cryptology ePrint Archive: Report 2002/168, 2002, https://eprint.iacr.org/2002/168.

[18] J. H. Cheon and B. Jun, "A polynomial time algorithm for the braid diffie-hellman conjugacy problem," in *Advances in Cryptology—CRYPTO 2003*, D. Boneh, Ed., vol. 2729 of *Lecture Notes in Computer Science*, pp. 212–225, Springer, Berlin, Germany, 2003.

[19] J. Hughes and A. Tannenbaum, *Length-Based Attacks for Certain Group Based Encryption Rewriting Systems*, Institute for Mathematics and Its Application, 2000, http://purl.umn.edu/3443.

[20] J.-M. Bohli, B. Glas, and R. Steinwandt, "Towards provable secure group key agreement building on group theory," Cryptology ePrint Archive: Report 2006/079, 2006, https://eprint.iacr.org/2006/079.

[21] P. Dehornoy, "Braid-based cryptography," in *Group Theory, Statistics, and Cryptography*, A. G. Myasnikov and V. Shpilrain, Eds., vol. 360 of *Contemporary Mathematics*, pp. 5–33, 2004.

[22] S.-H. Paeng, K.-C. Ha, J. H. Kim, S. Chee, and C. Park, "New public key cryptosystem using finite non abelian groups," in *Advances in Cryptology—CRYPTO 2001*, J. Kilian, Ed., vol. 2139 of *Lecture Notes in Computer Science*, pp. 470–485, Springer, Berlin, Germany, 2001.

[23] S. S. Magliveras, D. R. Stinson, and T. van Trung, "New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups," *Journal of Cryptology*, vol. 15, no. 4, pp. 285–297, 2002.

[24] M. I. G. Vasco, C. Martinez, and R. Steinwandt, "Towards a uniform description of several group based cryptographic primitives," Cryptology ePrint Archive: Report 2002/048, 2002.

[25] S. S. Magliveras, D. R. Stinson, and T. Van Trung, "New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups," *Journal of Cryptology*, vol. 15, no. 4, pp. 285–297, 2002.

[26] D. Grigoriev and I. Ponomarenko, "On non-Abelian homomorphic public-key cryptosystems," *Journal of Mathematical Sciences*, vol. 126, no. 3, pp. 1158–1166, 2002.

[27] D. Grigoriev and I. Ponomarenko, "Homomorphic public-key cryptosystems over groups and rings," https://arxiv.org/abs/cs/0309010v1.

[28] B. Eick and D. Kahrobaei, "Polycyclic groups: a new platform for cryptology?" https://arxiv.org/abs/math/0411077v1.

[29] V. Shpilrain and A. Ushakov, "Thompson's group and public key cryptography," in *Applied Cryptography and Network Security*, J. Ioannidis, A. Keromytis, and M. Yung, Eds., vol. 3531 of *Lecture Notes in Computer Science*, pp. 151–163, Springer, Berlin, Germany, 2005.

[30] A. Mahalanobis, *The diffie-hellman key exchange protocol, its generalization and nilpotent groups [Ph.D. dissertation]*, Florida Atlantic University, Boca Raton, Fla, USA, 2005.

[31] L. Wang, L. Wang, Z. Cao, E. Okamoto, and J. Shao, "New constructions of public-key encryption schemes from conjugacy search problems," in *Information Security and Cryptology: 6th International Conference, Inscrypt 2010, Shanghai, China, October 20–24, 2010, Revised Selected Papers*, vol. 6584 of *Lectures Notes in Computer Science*, pp. 1–17, Springer, Berlin, Germany, 2010.

[32] Z. Cao, X. Dong, and L. Wang, "New public key cryptosystems using polynomials over noncommutative rings ," *Journal of Cryptology—IACR*, vol. 9, pp. 1–35, 2007.

[33] J. Kubo, "The dihedral group as a family group," in *Quantum Field Theory and Beyond*, W. Zimmermann, E. Seiler, and K. Sibold, Eds., pp. 46–63, World Science Publication, Hackensack, NJ, USA, 2008.

[34] P. V. Reddy, G. S. G. N. Anjaneyulu, D. V. Ramakoti Reddy, and M. Padmavathamma, "New digital signature scheme using polynomials over noncommutative groups," *International Journal of Computer Science and Network Security*, vol. 8, no. 1, pp. 245–250, 2008.

[35] D. N. Moldovyan and N. A. Moldovyan, "A new hard problem over non-commutative finite groups for cryptographic protocols," in *Computer Network Security: 5th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2010, St. Petersburg, Russia, September 8–10, 2010. Proceedings*, vol. 6258 of *Lecture Notes in Computer Science*, pp. 183–194, Springer, Berlin, Germany, 2010.

[36] A. D. Myasnikov and A. Ushakov, "Cryptanalysis of matrix conjugation schemes," *Journal of Mathematical Cryptology*, vol. 8, no. 2, pp. 95–114, 2014.

[37] K. Svozil, "Non-contextual chocolate balls versus value indefinite quantum cryptography," *Theoretical Computer Science*, vol. 560, part 1, pp. 82–90, 2014.

[38] "Noncommutative cryptography," in *New Directions of Modern Cryptography*, Z. Cao, Ed., CRC Press, New York, NY, USA, 2013.

[39] S. R. Blackburn, "Groups of prime power order with derived subgroup of prime order," *Journal of Algebra*, vol. 219, no. 2, pp. 625–657, 1999.

[40] B. C. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, vol. 222, Springer, New York, NY, USA, 2003.

[41] M. Uno and M. Kano, "Visual cryptography schemes with dihedral group access structure," in *Proceedings of the 3rd International Conference on Information Security Practice and Experience (ISPEC '07)*, pp. 344–359, Springer, Berlin, Germany, 2007.

[42] D. S. Dummit and R. M. Foote, *Abstract Algebra*, John Wiley & Sons, Hoboken, NJ, USA, 3rd edition, 2004.

[43] T. Y. Lam, Ed., *Introduction to Quadratic Forms Over Fields Quaternion Algebras and Their Norm Forms*, vol. 67, American Mathematical Society, Berkeley, Calif, USA, 2005.

[44] "Diffie-hellman algorithm," in *Cryptography and Network Security: Principles and Practice*, W. Stallings, Ed., chapter 10, Pearson Education, New York, NY, USA, 5th edition, 2011.

[45] D. Ruinskiy, A. Shamir, and B. Tsaban, "Length-based cryptanalysis: the case of Thompson's group," *Journal of Mathematical Cryptology*, vol. 1, no. 4, pp. 359–372, 2007.

[46] A. D. Myasnikov and A. Ushakov, "Length based attack and braid groups: cryptanalysis of ANShel-ANShel-Goldfeld key exchange protocol," in *Public Key Cryptography*, vol. 4450 of *Lecture Notes in Computer Science*, pp. 76–88, Springer, Berlin, Germany, 2007.