Hindawi
Security and Communication Networks
Volume 2024, Article ID 9783712, 1 page
https://doi.org/10.1155/2024/9783712

Retraction

Retracted: A New Key Exchange Protocol Based on Infinite Non-Abelian Groups

Security and Communication Networks

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Zhang, Y.-J. Yang, and Y.-P. Li, "A New Key Exchange Protocol Based on Infinite Non-Abelian Groups," *Security and Communication Networks*, vol. 2022, Article ID 7942353, 6 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7942353, 6 pages
https://doi.org/10.1155/2022/7942353

Research Article

A New Key Exchange Protocol Based on Infinite Non-Abelian Groups

Jing Zhang,[1] Ya-Juan Yang ,[2] and Yi-Peng Li[3]

[1]*Modern Industrial Innovation Practice Center, Dongguan Polytechnic College, Dongguan, Guangdong, China*
[2]*School of Finance and Trade, Dongguan City University, Dongguan, Guangdong, China*
[3]*Department of Applied Mathematics, Xi'an University of Science and Technology, Xi'an, Shanxi, China*

Correspondence should be addressed to Ya-Juan Yang; 1909853gbm30002@student.must.edu.mo

Received 1 January 2022; Accepted 15 February 2022; Published 24 March 2022

Academic Editor: Xingsi Xue

In order to resist quantum attacks, a key exchange protocol based on infinite non-abelian groups is proposed in this paper. For the purpose, by the composition of twice the operation of a semidirect product, we construct a shared secret key which contains two hard problems of equivalent decomposition problem (EDP) and discrete logarithm problem (DLP). Then, two methods , algebra attack and brute force attack, were employed to verify the antiattack for the proposed protocol. By a sound mathematical inference, it demonstrates that the proposed protocol possesses security positively. Finally, we analyzed the computational complexity and bit complexity when the protocol being implemented on braid groups, and furthermore, the complexity data confirm the feasibility of establishing the key exchange protocol there. Thus, in any case, security or complexity, the actual use of the proposed protocol means achievable in practice.

1. Introduction

In 1993, Sidelnikov et al. proposed a new idea that infinite non-abelian group and semigroup can be used in public key cryptography [1]. The main problem of key exchange protocol on infinite non-abelian (semi) group is to hide some factors by hard problems, such as conjugacy search problem (CSP), decomposition problem (DP), subgroup membership search problem (MSP), discrete logarithm problem (DLP), and homomorphism search problem (HSP). Then since, many research studies have come up with some key protocols with a hard problem on the algebra structure. For example, in [2], the authors presented a key exchange protocol based on the DP for non-abelian groups. In [3], the authors presented a new cryptosystem with CSP on a braid group which is an infinite non-abelian group and has a good feature for cryptography. With the development of quantum algorithms and the improved factorization algorithms, we hold the opinion that only one hard problem is insufficient for the security of cryptosystem, especially only CSP is insufficient for the security of public key cryptography [4]. For enhancing security, it is a viable idea to carry out several hard problems simultaneously for a key exchange protocol.

In 2006, Sakalauskas et al. employed CSP and DLP to build a key agreement protocol in the group representation level and guaranteed there that it is sufficient to use two hard problems at the same time for the entire security of key exchange protocols [5]. In 2013, Habeeb et al. proposed a new kind of key exchange protocol which was the first time that a semidirect product of groups was applied to the cryptosystem [6]. In 2020, Skuratovskii built a key exchange protocol based on the metacyclic group of Miller's Moreno type which is a minimal non-abelian group, and it improves the efficiency of the key exchange [7].

At the same time, Aleksejus et al. defined a key exchange protocol using the matrix power function (MPF) on a non-abelian group M_{16}, modular group of order 16, whose security is based on the NP-complete LRMPF, left-to-right MPF, and decision problem [8], so researchers believe that the protocol is not vulnerable to quantum attack. And the cryptography on the non-abelian algebra structure has been an important research field to antiquantum attack [9–11].

In this paper, motivated by Habeeb's method for constructing a protocol on non-abelian groups, we proposed a key exchange protocol with DLP and EDP by the operation of a semidirect product on the infinite non-abelian group. Since there is no effective quantum algorithm of hard problem on the non-abelian group, it is desirable to use two hard problems in the proposed key protocol at the same time. Certainly, our protocol constructed in this way is antiquantum attack. Also, the security analysis of the protocol is carried out by algebra attack and brute force attack, and as an application example, the key exchange protocol was implemented on a braid group.

The remaining of this paper are organized as follows: mathematical preliminaries are introduced in Section 2 that are needed in the proposed protocol. The main results of this paper and a key exchange protocol based on infinite non-abelian groups are proposed in Section 3, while the security analysis for the protocol are located in section 4. Section 5 deals with the application of the protocol being implemented on a braid group, and Section 6 presents some new research fields of cryptography as a concluding remark.

2. Preliminaries

Group theory is the basis of this research; as a preliminary, we give a brief introductory on groups which will be used later, and details of these results may be found in [12, 13].

Definition 1 (semidirect product). Let $(G, \cdot)$ and $(G', \circ)$ be the semigroups, $Aut(G)$ be the group of automorphisms of G with composition operation, and let ρ: $G' \longrightarrow Aut(G)$ be a homomorphism. Then, the semidirect product of G and G' is the set of pairs

$$\Gamma = G \times_{\rho} G' = \{(g, h): g \in G, \ h \in G'\}, \tag{1}$$

with the binary operation

$$(g, h) * (g', h') = \left(g^{\rho(h')} \cdot g', h \circ h'\right). \tag{2}$$

Here, $g, g' \in G$ and $h, h' \in G'$, and $g^{\rho(h')}$ denotes the image of g under the automorphism $\rho(h')$.

It is proved that $(\Gamma, *)$ is a group and an extending group of direct products. In addition, if $(G, \cdot)$ and $(G', \circ)$ are the infinite group, then $(\Gamma, *)$ is also an infinite non-abelian group.

It is easy to see the followings: if we let $G' = Aut(G)$, $\rho = id_G$, then

$$\Gamma = G \times G' = \{(g, \phi): g \in G, \phi \in Aut(G)\}, \tag{3}$$

with the group operation

$$(g, \phi) * (g', \phi') = (\phi'(g) \cdot g', \phi \circ \phi'), \tag{4}$$

where $\phi \circ \phi$' denote a composition of automorphism and ϕ' acting first.

Note 1

Let $Bij(G)$ denote a set of all the bijection on $G \longrightarrow G$. In our key exchange protocol, we only need that mapping ϕ is a bijection, that is $\phi \in Bij(G)$. In the set $\Gamma = G \times G' = \{(g, \phi): g \in G, \phi \in Bij(G)\}$, we still define the binary operation as above $(g, \phi) * (g', \phi') = (\phi'(g) \cdot g', \phi \circ \phi')$, and define $(g, \phi)^m = (g, \phi)^{m-1}(g, \phi)$ where $m \in Z_+$. If G is a group, then for $\forall a, b \in G$, let $\phi = \phi_b(a) = b^{-1}a$; obviously, $\phi_b(a)$ is a bijection, and $\phi_b^m(a) = b^{-m}a$. So, $(g, \phi_b)^m = (b^{-(m-1)} \cdot g^m, \phi_b^m)$.

Definition 2 (Centralizer). Let G be a group, $g \in G$; the set $C_G(g) = \{a \in G | a^{-1}ga = g\}$ is called the centralizer of g.

In Fact, $C_G(g)$ is a Subgroup of G satisfying $ag = ga$ for $\forall a \in C_G(g)$

Decomposition problem (DP): let G be a platform group, for $\forall \mu, \vartheta \in G$ and two subgroups A, B < G, to find $\alpha \in A$, $\beta \in B$ satisfying $\mu = \alpha \cdot \vartheta \cdot \beta$. It is a hard problem. Here, we proposed an equivalent decomposition problem and proved that the hardness of the two problems is similar.

Definition 3 (equivalent decomposition problem (EDP)). Let G be a platform group, for $\forall \mu, \vartheta \in G$, to find $\alpha, \theta^n \in G$ satisfying $\mu = \rho \cdot \vartheta^n$; here, $n \in Z_+$.

Since $\mu = \alpha \cdot \vartheta^n \Leftrightarrow \mu = \alpha \cdot \vartheta \cdot \vartheta^{n-1}$, it is easy to see that finding α and θ^n is the same as finding α and θ^{n-1}. This is equivalent to find α and β, in the decomposition problem. So, the hardness of EDP is the same with the decomposition problem.

Definition 4 (discrete logarithm problem). Let G be a platform group, for $\forall \mu, \vartheta \in G$, to find $n \in Z$ satisfying $\mu = \vartheta^n$.

3. Key Exchange Protocol

Suppose both parties of cryptography are Alice and Bob, they transmit messages by a public channel. Before describing our protocol, we first introduce Habeeb's semidirect product method applied to the protocol construction.

3.1. Semidirect Product Method. Habeeb et al. proposed a new kind of key exchange protocol based on the semidirect product of group in [6]. Let G be a grou p with automorphism $\phi \in Aut(G)$, and element $g \in G$. Suppose Alice and Bob agree on group G, Alice chooses a private $m \in Z_+$ and Bob chooses a private $n \in Z_+$. Habeeb's key exchange protocol is constructed as follows:

Public key: g

Private key: m, n

step 1. Alice computes $(g, \phi)^m = (\phi^{m-1}(g) \cdots \phi^2(g)\phi(g)g, \phi^m)$, denoting $A = \phi^{m-1}(g) \cdots \phi^2(g)\phi(g)g$, and sends A to Bob

step 2. Bob computes $(g, \phi)^n = (\phi^{n-1}(g) \cdots \phi^2(g)\phi(g)g, \phi^n)$, denoting $B = \phi^{n-1}(g) \cdots \phi^2(g)\phi(g)g$, and sends B to Alice

step 3. Alice computes $(B, x) \cdot (A, \phi^m) = (\phi^m(B)A, x\phi^m)$; her key now is $K_A = \phi^m(B)A$

step 4. Bob computes $(A, y) \cdot (B, \phi^n) = (\phi^n(A)B, y\phi^n)$; his key now is $K_B = \phi^n(A)B$

Since $(B, x)\cdot(A, \phi^m) = (A, y)\cdot(B, \phi^n) = (g, \phi)^{m+n}$, so $K_A = K_B = K$, that is, the shared secret key.

For the security of the above protocol, the authors only addressed the security of a particular instantiation, which does not depend on any open hard problem. But, due to the security, it is necessary to build a key exchange protocol with open hard problems. That's the reason why we propose a key exchange protocol on non-abelian groups as follows.

3.2. A New Key Exchange Protocol. Now, we introduce the key exchange protocol on infinite non-abelian groups. Let G be an infinite non-abelian multiplication group, mapping $\phi = \phi_b(a) = b^{-1}a \in Bij(G)$; here, Alice and Bob can select a different parameter b in the protocol for $\forall b,\ a \in G$. Suppose Alice and Bob are agreed on group G, element $g \in G$, a key exchange protocol is then be constructed as follows:

Public key: g

Private keys: m, n, k, h

step 1. Alice selects private keys $m \in Z_+$ and $k \in G$ satisfying $kg \neq gk$; then, she has a mapping $\phi_k(a) = k^{-1}a$: $G \longrightarrow G$. By computing the centralizer $C_G(k^{-1})$, and selecting a subgroup G' of $C_G(k^{-1})$ such that $|G'| \geq N$, here N is a positive integer, and she computes the product

$$(g, \phi_k)^m = \left(k^{-(m-1)}g^m, \phi_k^m\right), \tag{5}$$

and then, by letting $A = k^{(-(m-1))}g^m$, $A' = k^{-1}A = k^{-m}g^m$, she sends A' and G' to Bob.

step 2. Bob selects private keys $n \in Z_+$ and $h \in G^{'}$ satisfying $hg \neq gh$; then, he has a mapping $\phi_h(a) = h^{-1}a$: $G \longrightarrow G$ and computes the product

$$(g, \phi_h)^n = \left(h^{-(n-1)}g^n, \phi_h^n\right), \tag{6}$$

and then, by letting $B = h^{-(n-1)}g^n$, $B^{'} = h^{-1}B = h^{-n}g^n$, he sends $B^{'}$ to Alice.

step 3. Alice computes

$$\left(B'\cdot k^{(m-1)}, x\right)(A, \phi_k^m) = \left(\phi_k^m\left(B'\cdot k^{(m-1)}\right)A, x\phi_k^m\right). \tag{7}$$

In fact, Alice needs only to compute $\phi_k^m(B'\cdot k^{(m-1)})A$ without having to calculate $x\phi_k^m$ since she does not know the mapping $x = \phi_h^n$, and her key is

$$\begin{aligned} K_{\text{Alice}} &= \phi_k^m\left(B'\cdot k^{(m-1)}\right)A, \\ &= k^{-m}h^{-n}g^n k^{(m-1)}k^{-(m-1)}g^m, \\ &= k^{-m}h^{-n}g^{m+n}. \end{aligned} \tag{8}$$

step 4. Bob computes

$$\left(A'\cdot h^{(n-1)}, y\right)(B, \phi_h^n) = \left(\phi_h^n\left(A'\cdot h^{(n-1)}\right)B, y\phi_h^n\right). \tag{9}$$

The same as Alice, Bob need not compute $y\phi_h^n$, and his key is

$$\begin{aligned} K_{Bob} &= \phi_h^n\left(A'\cdot h^{(n-1)}\right)B, \\ &= h^{-n}k^{-m}g^m h^{(n-1)}h^{-(n-1)}g^n, \\ &= h^{-n}k^{-m}g^{m+n}. \end{aligned} \tag{10}$$

Since $h \in G'$, G' is a group, so $h^{-1} \in G'$, t hus, we have the equation

$$h^{-1}k^{-1} = k^{-1}h^{-1} \Rightarrow h^{-n}k^{-m} = k^{-m}h^{-n}. \tag{11}$$

That is, for $\forall m, n \in Z_+$, $h^{-n}k^{-m}g^{m+n} = k^{-m}h^{-n}g^{m+n}$, so $K = K_{Alice} = K_{Bob}$. Thus, the shared secret key will be

$$K = k^{-m}h^{-n}g^{m+n} = h^{-n}k^{-m}g^{m+n}. \tag{12}$$

In computing $C_G(k^{-1})$, Alice need not figure out all the elements of $C_G(k^{-1})$; she need to only calculate the elements that satisfy the safety requirements. In addition, when Alice sends subgroup G' to Bob, she needs just to send the generator set S of G', namely, $G' = <S>$.

Note 2

Although the proposed protocol employing the semidirect product of group which was being used by Habeeb's in [6], ours have several differences with Habeeb's, mainly in the following aspects:

(1) In Habeeb's protocol using the semidirect product of group, mapping ϕ is an automorphism of group. Our protocol only uses the operation rule of the semidirect product, and mapping ϕ is just a bijection on group G.

(2) Centralizer is applied in our protocol, which could guarantee the commutativity between Alice's private key k and Bob's private key h, but Habeeb's protocol has no element commutativity on the group.

(3) Obviously, our protocol contains two private keys for both parties, but Habeeb's contains only one.

(4) The security assumptions are different. The security of our protocol is based on both hard problems: EDP and DLP, but Habeeb's protocol does not.

In our protocol, the security assumption is that it is difficult to solve EDP and DLP on the non-abelian group. Next, in order to verify the security of our protocol, we have to examine it.

4. Security Analysis

Security of key exchange protocol relies on both hard problems on the infinite non-abelian group: equivalent decomposition problem (EDP) and discrete logarithm problem (DLP). During the construction of the key, A' and B' are transferred. Since k^{-m} *and* h^{-n} can be regarded as

entirety, respectively, which are unknown for attacker, it is an EDP to find k^{-m}, h^{-n}, g^m, g^n from equations $A' = k^{-m}g^m$ and $B' = h^{-n}g^n$. And to find private keys m and n, the attacker must solve DLP.

Alice and Bob exchange messages by a public channel; an attacker can observe the transmission of the protocol. He can get triple $(g, A' = k^{(-m)}g^m, B' = h^{-n}g^n)$, and his aim is to get the shared secret key K. If he can figure out k^{-m}, g^m from A', then $K = k^{-m}B'g^m$ can be captured by him. Similarly, if he can get h^{-n}, g^n from B', then the shared secret key also can be captured. Let us consider one of the both situations.

4.1. Algebra Attack. Due to k *and* m are unknown, hence k^{-m}, g^m cannot be figured out directly, and an attacker can choose an arbitrary element $\overline{k} \in G$; replacing k^{-m} with $\overline{k}$, he gets

$$\overline{k}g^m = A' \Rightarrow g^m = A'\overline{k}^{-1}. \tag{13}$$

It is a DLP to compute m from the above equation, so there is nopolynomial time algorithm to find m. Thus, letting g^m replaced by $A'\overline{k}^{-1}$, he has

$$\overline{K} = \overline{k}B'g^m = \overline{k}h^{-n}g^nA'\overline{k}^{-1} = \overline{k}h^{-n}g^nk^{-m}g^m\overline{k}^{-1}. \tag{14}$$

Since G is a non-abelian group, so $\overline{K} \neq K$.

On the contrary, an attacker also can choose an arbitrary integer $\overline{m} \in Z_+$; replacing m with $\overline{m}$ and g^m with $g^{\overline{m}}$, he gets

$$k^{-m}g^{\overline{m}} = A' \Rightarrow k^{-m} = A'g^{-\overline{m}}. \tag{15}$$

Let $\overline{K} = k^{(-m)}B'g^{\overline{m}} = A'g^{(-\overline{m})}B'g^{\overline{m}} = k^{(-m)}g^mg^{(-\overline{m})}$ $h^{(-n)}$ $g^ng^{\overline{m}}$, since G is a non-abelian group, so $\overline{K} \neq K$.

The algebra attack is the same for h *and* n, and we omit the deductions here. These attacking results show that an attacker cannot get the shared key and so the protocol is safe for algebra attack.

4.2. Brute Force Attack. Brute force attack means that the attacker exhausts all possibilities to find a private message. Because k comes from group G, h comes from group G' and $G' < G$; therefore , it is more effective to attack the protocol from G'. That means an attacker tries to find out all the possible $\overline{h} \in G'$ for $n \in Z_+$ satisfying $\overline{h}g^n = B'$.

Since $G' < C_G(k)$, we have $N \leq |G'| \leq |C_G(k)|$, and there are at least N possibilities of $\overline{h}$. Assuming that the order of g is γ, then the attacker needs to compute g^i for γ times $(i \leq \gamma)$. Because G is an infinite group, $|C_G(k)|$ and γ could be large enough; this means that the complexity of the brute force attack is between $N\gamma$ and $|C_G(k)|\gamma$.

If G is chosen as a proper group, such as a braid group, it is impossible to get a shared key by brute force attack.

5. Application to Braid Group

In the cryptosystem, braid groups have given rise to the attention of cryptographer for several years [16–22]. Because braid groups are infinite non-abelian groups with exponential growth respect to the braid index, there exist fast algorithms to perform on group operations in a normal way of elements for the braid group, and there are many hard problems based on topological or group-theoretical open problems on the braid group. Thus, it is suitable to implement our key exchange protocol on a braid group.

5.1. Braid Group. Given an integer $\overline{n} \geq 2$, the $\overline{n}$-braid group $B_{\overline{n}}$ is defined by following group presentation: $B_{\overline{n}} = (e, \sigma_1, \cdots, \sigma_{n-1} | \sigma_i\sigma_j = \sigma_j\sigma_i \ \ if\ |i-j| \geq 2;\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_{(j)} \ \ if\ |i-j| = 1)$, here, the integer $\overline{n}$ is called the braid index, and each element of $B_{\overline{n}}$ is called a $\overline{n}$ braid.

There are several kinds of group representation for braid groups, such as Burau representation, Garsides representation, and Birman–Ko–Lee representation. Our protocol is based on the Elrifai–Morton representation for the braid group [16], where each braid has a unique left-canonical form.

Theorem 1 (see [16]). *For any $W \in B_{\overline{n}}$, there is a unique representation, called the left-canonical form:*

$$W = \Delta^u A_1 A_2 \cdots A_P, \quad u \in Z,\ A_i \in \widetilde{\Sigma}_{\overline{n}} \backslash \{e, \Delta\}, \tag{16}$$

Here $\widetilde{\Sigma}_{\overline{n}}$ is the set of all permutation braids, Δ is the fundamental braid, and A_iA_{i+1} is the left-weighted for $1 \leq i \leq p-1$.

Then, a braid $W = \Delta^u A_1 A_2 \cdots A_P$ could be described as a tuple $(u, \pi_1, \pi_2, \cdots \pi_p)$, where permutation braids A_i corresponding to the permutation π_i, and p is called the canonical length of W denoted by len (W). In [16], the authors also proposed the word algorithm for the representation of word in Artin generators.

5.2. Key Exchange Protocol on Braid Group. In our key exchange protocol, $m, n \in Z_+$, let the public key be $g \in B_{\overline{n}}$ and the private key be $k \in B_{\overline{n}}$ *and* $h \in B_{\overline{n}}$. By the word algorithm, the braids are represented in the left-canonical form, and the public key g is represented in advance. Centralizer $C_{B_{\overline{n}}}(k^{-1})$ could be obtained by the algorithm in [17].

5.3. Complexity Analysis. According the left-canonical form of braids, two parameters, the braid index and the canonical length, need be considered. For simplicity, we assume that the braid index is $\overline{n}$ and the canonical length is p for all the braids in our key exchange protocol.

5.3.1. Computational Complexity. In the key exchange protocol, for Alice, there exists one braid $k \in B_{\overline{n}}$ that needs be represented in the left-canonical form. She needs to compute k^{-1}, $k^{-m}g^m$, $C_{B_n}(k^{-1})$, and $k^{-m}B'k^{(m-1)}A$, while k^{-m} and $k^{(m-1)}$ should be computed in advance. It has been proved that we compute $C_{B_{\overline{n}}}(k^{-1})$ in needing times $O(t^3\ell^2\overline{n}^6\log\overline{n})$, see [17]; here, t is the number of elements in supersummit set of k^{-1}, which is bounded by a polynomial in ℓ. Since Alice needs only to calculate some elements of

Table 1: Time complexity for Alice.

Operations	The number of operations	Time complexity
Word algorithm of k	1 time	$O(\ell^2\overline{n}\log\overline{n})$
k^{-1}	1 inverse operation	$O(p\overline{n})$
$k^{-m}g^m$	$2m$ multiplication	$O(2mp^2\overline{n}\log\overline{n})$
$k^{-m}B'k^{(m-1)}A$	3 multiplications	$O(p^2\overline{n}\log\overline{n})$
$C_{B_{\overline{n}}}(k^{-1})$	1 time	$O(\ell^5\overline{n}^6\log\overline{n})$

Table 2: Time complexity for Bob.

Operations	Number of operations	Time complexity
Word algorithm of h	1 time	$O(\ell^2\overline{n}\log\overline{n})$
h^{-1}	1 inverse operation	$O(p\overline{n})$
$h^{-n}g^n$	$2n$ multiplication	$O(2np^2\overline{n}\log\overline{n})$
$h^{-n}A'h^{(n-1)}B$	3 multiplications	$O(p^2\overline{n}\log\overline{n})$

Table 3: Bit complexity for Alice.

Braids	Bit complexity
Generators of subgroup G'	$p\frac{\overline{n}^2(\overline{n}-1)}{2}\log\overline{n}$
$k^{(m-1)}$	$(m-1)p\overline{n}\log\overline{n}$
A	$(m-1)mp\overline{n}\log\overline{n}$
ϕ_k^m	$mp\overline{n}\log\overline{n}$
g	$p\overline{n}\log\overline{n}$
K	$2(m+n)p\overline{n}\log\overline{n}$

Table 4: Bit complexity for Bob.

Braids	Bit complexity
$h^{(n-1)}$	$(n-1)p\overline{n}\log\overline{n}$
B	$(n-1)np\overline{n}\log\overline{n}$
ϕ_h^n	$np\overline{n}\log\overline{n}$
g	$p\overline{n}\log\overline{n}$
K	$2(m+n)p\overline{n}\log\overline{n}$

$C_{B_{\overline{n}}}(k^{-1})$, it is appropriate to assume that the complexity is $O(\ell^5\overline{n}^6\log\overline{n})$.

Lemma 1 (see [3]).

(1) Let W be a word on σ_i's with a word length ℓ; then, the left-canonical form of W can be computed in time $O(\ell^2\overline{n}\log\overline{n})$.

(2) Let $U = \Delta^u A_1 A_2 \cdots A_P$ and $V = \Delta^v A_1 A_2 \cdots A_q$ be the left-canonical form of $\overline{n}$ braids, then we compute the left-canonical forms of UV in time $O(pq\overline{n}\log\overline{n})$.

(3) If $U = \Delta^u A_1 A_2 \cdots A_P$ be the left-canonical form of $U \in B_{\overline{n}}$, then we compute the left-canonical form of U^{-1} in time $O(p\overline{n})$.

By Lemma 1, the computational complexity is summarized in Table 1 for Alice and Table 2 for Bob.

5.3.2. Bit Complexity. While implementing our key exchange protocol algorithm, for Alice, a subgroup G' of $C_{B_{\overline{n}}}(k^{-1})$, A, $k^{(m-1)}$, mapping ϕ_k^m, public key g, and shared secret key K needs be stored. For $\forall a \in B_{\overline{n}}$, generators of $C_{B_{\overline{n}}}(a)$ are less than $\overline{n}(\overline{n}-1)/2$ (see [18]). If we let $G' = C_{B_{\overline{n}}}(k^{-1})$, it is necessary to store $\overline{n}(\overline{n}-1)/2$ braids which is the worst case for storage space. For mapping ϕ_k^m, it is enough to store the braid k^{-m}.

A braid with p canonical factors can be represented by a bit string of size $p\overline{n}\log\overline{n}$, for braids $a, b \in B_{\overline{n}}$, $len(ab) \le len(a) + len(b)$ (see [3]). In the worst case, the bit complexity is summarized in Table 3 for Alice and Table 4 for Bob.

Therefore, the bit complexity is less than $(\overline{n}(\overline{n}-1)/2 + m^2 + 3m + 2n)\log\overline{n}$ for Alice and less than $(n^2 + 3n + 2m)\log\overline{n}$ for Bob.

6. Concluding Remarks

Nowadays, the main trend in cryptography theory locates still on constructing the cryptosystem based on a hard mathematical problem. However, quantum computer is no longer a dream in the near future; by then, many cryptosystems may be crumbled. So, it is urgent to design the cryptosystem against the quantum computing attacks. In our opinion, as a platform, a non-abelian algebra structure may be a good option.

In addition, developed in recent decades, bionic algorithms, such as evolutionary algorithm, neural network algorithm, genetic algorithm, meme algorithm, and DNA algorithm, have become an important way to transform traditional cryptography. In particular, the key evolution algorithm and meme algorithm have become the focus of research in recent years. The progress of evolutionary algorithms and the related problems in recent years refer to [23, 24], and the state for memetic algorithms can refer to [25–27]. In terms of methodology, it can be predicted that the bionic algorithms could be a promising research area of cryptography theory in the future. [14, 15].

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the financial support from the Office of Philosophy and Social Science Research Project of Guang Dong Province under grant no. GD17XYJ29.

References

[1] V. Sidelnikov, M. Cherepnev, and V. Yaschenko, "Systems of open distribution of keys on the basis of noncommutative semigroup," *Russian Acad. Sci.Dokl. Math.*vol. 48, no. 2, pp. 566-567, 1993.

[2] V. Shpilrain and A. Ushakov, *A New Key Exchange Protocol Based on the Decomposition Problem*, Cryptology and Information Security Series, vol. 2005, pp. 447–453, 2005.

[3] 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," *Advances in Cryptology - CRYPTO 2000* in *Proceedings of Annual International Cryptology Conference*, vol. 1880, Springer, Berlin, pp. 166–183, 2000.

[4] V. Shpilrain and A. Ushakov, "The conjugacy search problem in public key cryptography: unnecessary and insufficient," *Applicable Algebra in Engineering, Communication and Computing*, vol. 17, no. 3-4, pp. 285–289, 2006.

[5] E. Sakalauskas, P. Tvarijonas, and A. Raulynaitis, "Key agreement protocol (kap) using conjugacy and discrete logarithm problems in group representation level," *Informatica*, vol. 18, no. 1, pp. 115–124, 2006.

[6] M. Habeeb, D. Kahrobaei, C. Koupparis, and V. Shpilrain, *Public Key Exchange Using Semidirect Product of (Semi)groups* in *International Conference on Applied Cryptography and Network Security*, vol. 2013, pp. 226–237, Cryptology and Information Security Series, 2013.

[7] R. Skuratovskii and A. Williams, "Some approach to key exchange protocol based on non-commutative groups," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 14, no. 2, pp. 5–7, 2020.

[8] M. Aleksejus, S. Eligijus, and L. Kestutis, "Key exchange protocol defined over a non-commuting group based on an NP-complete decisional problem," *Symmetry*, vol. 12, p. 1389, 2020.

[9] A. D. Myasnikov and A. Ushakov, "Quantum algorithm for the discrete logarithm problem for matrices over finite group rings," *Groups Complexity Cryptology*, vol. 6, no. 1, pp. 31–36, 2014.

[10] D. Kahrobaei, H. T. Lam, and V. Shpilrain, "Pubilc key exchange using extensions by endomorphisms and matrices over a Galois field," in *Proceedings of the DIMACS Workshop on Multicore and Cryptography*pp. 1–9, Hoboken NJ USA, 2014.

[11] V. Shpilrain and G. Zapata, "Combinatorial group theory and public key cryptography," *Applicable Algebra in Engineering, Communication and Computing*, vol. 17, no. 3-4, pp. 291–302, 2006.

[12] J. S. Birman, *Braids, Links and Mapping Class Groups*, Princeton university press, Princeton New Jersey, 1976.

[13] T. W. Hungerford, *Algebra*, pp. 59–63, Springer-Verlag press, New York, USA, 1974.

[14] Y. Jun and L. Qing, "Group-based cryptanalysis of a key exchange protocol based on semidirect products," *Journal of Southwest University for Nationalities (Natural Science Edition)*, vol. 43, no. 2, pp. 157–160, 2017.

[15] Z. Run-Zhi and W. Li-Bin, "An efficient certificateless authenticated key exchange protocol," *Journal of Cryptologic Research*, vol. 7, no. 4, pp. 421–429, 2020.

[16] E. A. Elrifai and H. R. Morton, "Algorithms for positive braids," *The Quarterly Journal of Mathematics*, vol. 45, no. 4, pp. 479–497, 1994.

[17] N. Franco and J. González-Meneses, "Computation of centralizers in braid groups and garside groups," *Revista Matemática Iberoamericana*, vol. 19, no. 2, pp. 367–384, 2007.

[18] J. Gonzalezmeneses and B. Wiest, "On the structure of the centralizer of a braid," *Annales Scientifiques de l'Ecole Normale Superieure*, vol. 37, no. 5, pp. 729–757, 2004.

[19] S. B. Gashkov and I. S. Sergeev, "Complexity of computation in finite fields," *Journal of Mathematical Sciences*, vol. 191, no. 5, pp. 661–685, 2013.

[20] M. Kreuzer, A. D. Myasnikov, and A. Ushakov, "A linear algebra attack to group-ring-based key exchange protocols," *Applied Cryptography and Network Security*, pp. 37–43, 2014.

[21] R. Gennaro and D. Micciancio, "Cryptanalysis of a pseudorandom generator based on braid groups," *Advances in Cryptology - EUROCRYPT 2002*, pp. 1–13, 2002.

[22] J. Longrigg and A.. Ushakov, "A practical attack on a certain braid group based shifted conjugacy authentication protocol," *Groups Complexity Cryptology*, vol. 1, no. 2, pp. 275–286, 2009.

[23] X. Xue and J.-S. Pan, "An overview on evolutionary algorithm based ontology matching," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 1, pp. 75–88, 2018.

[24] C. Dai, Y. Wang, M. Ye, X. Xue, and H. Liu, "An orthogonal evolutionary algorithm with learning automata for mo," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3306–3319, 2016.

[25] X. Xue and Y. Wang, "Optimizing ontology alignments through a memetic algorithm using both MatchFmeasure and uir," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[26] X. Xue and J. Chen, "Optimizing ontology alignment through hybrid population-based incremental learning algorithm," *Memetic Computing*, vol. 11, no. 2, pp. 209–217, 2019.

[27] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2016.