JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 1035-1050 (2022)
DOI: 10.6688/JISE.202209_38(5).0010

Semidirect Products of Finite Groups in Public-Key Cryptographic Protocols

G. H. J. LANEL[1], T. M. K. K. JINASENA[2]
AND B. A. K. WELIHINDA[1,+]
[1]*Department of Mathematics*
[2]*Department of Computer Science*
University of Sri Jayewardenepura
Nugegoda 10250, Sri Lanka
E-mail: ghjlanel@sjp.ac.lk; kasun@sjp.ac.lk; kasuniwe@gmail.com[+]

Public-key cryptosystems using non-abelian groups had been a research inspiration especially since the proposal of Shor's quantum algorithm attack in 1994. In this article, we prove two approaches to novel encryption schemes using elements of some non-abelian groups based on an intractable problem of determining automorphisms and generating elements of a group. We show that the difficult problem of determining paths and cycles of Cayley graphs including Hamiltonian paths and cycles could be reduced to this intractable problem. Moreover, achievement of resistance to algebraic span cryptanalysis by integrating a technique introduced in the existing literature is discussed.

Keywords: Algebraic span cryptanalysis, Cayley graphs, Hamiltonian Path/Cycle Problem, Non-abelian/Non-commutative, Semidirect products

1. INTRODUCTION

Modern Cryptography, also known as Algebraic Cryptography has become a crucial and an attractive field of research over the years. Public-key cryptosystems used at present, utilize abelian groups and are mainly dependent on Number theoretical principles. However, with the identification of insecurities if the quantum computers are to be invented and the development of more and more cryptographic attacks to the existing protocols, attention is diverted on developing novel cryptographic protocols based on varied approaches. One such approach is the non-abelian group based Cryptography.

Our attention was directed towards using non-abelian groups for Cryptography during a study on the existence of Hamiltonian cycles in Cayley graphs (for a recent literature review and some advancements in this direction, see [1–9] etc.). There, we had developed the intuition that the Hamiltonian Cycle Problem (HCP) and Hamiltonian Path Problem (HPP) will be suitable choices of intractable problems, particularly over Cayley graphs of non-abelian groups for the development of novel cryptographic protocols. Study of the

Received February 23, 2021; revised September 17, 2021; accepted November 5, 2021.
Communicated by Raylin Tso.
[+] Corresponding author.

related literature (see [10–15] etc.) proved that indeed, non-abelian group based Cryptography has already become a latest attraction of many scholars and that it is identified to be resistant to quantum computer attacks due to the non-abelian structure (for instance, see [16–18] for clarifications on the emergence of non-abelian group based Cryptography as a possible solution to quantum attacks).

The past researches were focused on underlying problems such as Conjugacy Problem, Conjugacy Search Problem, Word Problem, Factorization and Membership Search Problems etc. in non-abelian groups, but a significant application of the HCP or HPP was lacking. Most of intractable problems are actually generalizations of some conventional cryptographic problem to non-abelian groups: e.g. [19, 20] show one way of generalizing the traditional Discrete Logarithm Problem (DLP) to problems over non-abelian groups.

Anshel-Anshel-Goldfeld [19] and Ko-Lee [20] protocols are two of the very first developments based on non-abelian groups making use of conjugacy related problems. Following their initiation, many other scholars (see [15, 21, 22]) have attempted in presenting generalizations of the DLP. Furthermore, novel schemes presented in studies such as [13, 16, 17, 23, 24] show the applications of different variants of Factorization Problems and Membership Search Problems over non-abelian groups whereas [25–27] are some applications based on the Word Problem. Several scholars have also conducted studies on using automorphisms of non-abelian groups to develop cryptosystems; the encryption and signature protocols by A. Mahalanobis [28], the MOR cryptosystem [29], Moldenhauer protocol [30] and the protocol by Paeng et al. [31] are some of the examples.

Almost all of the non-abelian group based schemes in the existing literature were proven to be vulnerable (see [12, 32–34] for some of the attacks developed by various authors). Especially, the recent introduction of the algebraic span method of cryptanalysis by B. Tsaban [32] has led to the breakage of the security of majority of schemes whose platform groups can be efficiently and faithfully represented as matrix groups. Faithful representations as matrix groups, either efficient or otherwise, always exist for finite groups. Hence, this novel attack is particularly challenging. V. Roman'kov [35], has introduced a protocol which is an advanced version of the Anshel-Anshel-Goldfeld [36] scheme, offering resistance to the algebraic span cryptanalysis.

In this manuscript, we propose two novel techniques for encryption protocols using the elements of semidirect products of finite groups based on an intractable problem of determining automorphisms and generating elements of a group. Thereafter, we show that the mathematically hard problem of determining paths and cycles, including Hamiltonian paths and cycles in Cayley graphs can also be reduced to this intractable problem. Hence, we suggest that Hamiltonian paths/cycles and in fact, any random paths/cycles could be used in the above protocols. Furthermore, we discuss modifications that can be included by integrating the technique proposed in [35] such that the protocols will be further resistant to Tsaban's span method of cryptanalysis.

Paper outline. In Section 2, we recall the fundamental mathematical concepts required to understand our results. Section 3 includes the novel cryptograpic protocols proposed followed by a discussion on its security. The next section is devoted for presenting the use of paths and cycles in Cayley graphs in the protocols. Thereafter, we discuss about securing the schemes against the algebraic span cryptanalysis and associated computational costs. The final section includes conclusions and future recommendations.

2. PRELIMINARIES

This section briefly discusses the fundamental concepts related to this study.

A semidirect product G, between two finite groups H and K, where H is normal, is denoted by $G = H \rtimes_\phi K$. It is defined by the homomorphism $\phi : K \to \mathrm{Aut}(H)$ and maps $k \in K$ to automorphisms ϕ_k of H. Here, $\mathrm{Aut}(H)$ denotes the automorphism group of H. Since the order of G is same as that of the semidirect product of H and K whose underlying set is $H \times K$, $|G| = |H| \times |K|$.

The group law in G can be stated as, $(h,k)(h_1,k_1) = (h\phi_k(h_1), kk_1)$, where $(h,k),(h_1,k_1) \in G$. $\phi_k(h) = k^{-1}hk$, for $k \in K, h \in H$. The (additive) identity element of G can be denoted as (e_H, e_K), while the inverse of an element (h,k) is $(\phi_{k^{-1}}(h^{-1}), k^{-1})$. For any ϕ_k, $\phi_k(h) = k^{-1}hk$, and $\phi_{k^{-1}}(h) = (k^{-1})^{-1}hk^{-1} = khk^{-1}$. Hence, $\phi_{k^{-1}}(h) = \phi_k^{-1}(h)$. Also, $\phi_k^m(h) = \underbrace{\phi_k(\phi_k(\cdots(\phi_k}_{m-times}(h)))) = k^{-m}hk^m = (k^m)^{-1}hk^m = \phi_{k^m}(h)$.

Definition 1. A vertex-transitive graph is a graph X, with automorphisms of X which maps v_1 to v_2, for any two vertices $v_1, v_2 \in V(X)$, where $V(X)$ is the set of vertices of X.

A Cayley graph of a finite group G defined as follows, with respect to a finite subset S of G, where $1 \notin S$ and S is inverse closed, is a type of a vertex-transitive graphs.

Definition 2. The Cayley graph of G with respect to S, $Cay(G,S)$ is the graph whose vertices are the elements of G and g is adjacent to gs for all $g \in G$, $s \in S$.

A cycle/path in a spanning subgraph is also a cycle/path in the ambient graph. Therefore, it is sufficient to identify the existence of cycles/paths in Cayley graphs with respect to irredundant generating sets.

Definition 3. An irredundant generating set for a Cayley graph X is a generating set S such that no proper subset of S generates X.

A Hamilton path is a one that visits every vertex exactly once. If there is an edge between the starting and ending vertices, it is a Hamilton cycle. Determining whether a graph consists of a Hamiltonian cycle or a path are known as HCP and HPP, respectively.

3. ENCRYPTION PROTOCOLS USING SEMIDIRECT PRODUCTS

Firstly, we present the novel notion proposed by us, restating the Lemma 1 from [37], which will be the foundation for the proposed protocols. Consider a semidirect product, $G = H \rtimes_\mu K$, where H and K are finite groups and H is the normal subgroup. We assume that the group operations are efficiently computable.

For any two groups, say $G_1 = H \rtimes_\phi K$ and $G_2 = H \rtimes_\theta K$, the underlying set is the same, which is $H \times K$. Thus the elements of both are the same. The Cayley graphs of each with respect to their corresponding irredundant generating sets will be different.

If communicating parties are to communicate using two such groups chosen by each, both will be performing computations under the same $mod\,p, mod\,q$ etc., using same elements whereas the product of any two elements has to be computed using two different homomorphisms ϕ and θ by each user. When two parties choose two groups like G_1, G_2, both of them as well as any eavesdropper is aware of H, K and the elements of the group.

But homomorphisms ϕ, θ and generating sets chosen are only known by respective party.

Lemma 1. [37] Determining ϕ or a set of generating elements of a semidirect product $H \rtimes_\phi K$ used by a communicating party (when only H, K, a sequence of elements of the form $\phi_{s_j^k}(h)$, where $h \in H$ is known), is a mathematically intractable problem.

Let the communicating parties be Alice and Bob. Suppose Alice communicates using $G_1 = H \rtimes_\phi K$ whereas Bob uses $G_2 = H \rtimes_\theta K$.

Assumption 1. Assume that the messages are represented as elements of $H \times K$.

3.1 Encryption Protocol 1:

1. Alice chooses an irredundant generating set for H, say, $H = \langle h_1, \cdots, h_{i_a} \rangle$.

   Let S_a be an irredundant generating set for G_1, chosen by Alice. $S_a = \{t_1, \cdots, t_{m_a}, s_1, \cdots, s_{n_a}\}$; $t_1, \cdots, t_{m_a} \in H$ and $s_1, \cdots, s_{n_a} \in K$. Some (or all) of the elements in $\{t_1, \cdots, t_{m_a}\}$ might be equal to some (or all) elements in $\{h_1, \cdots, h_{i_a}\}$ or could be expressed in terms of the product of several elements in $\{h_1, \cdots, h_{i_a}\}$. Let the homomorphism ϕ be defined by the action of the elements $s_1, \cdots, s_{n_a} \in K$ on H. For $s_{j_a} \in K$, $1 \leq j_a \leq n_a$, let $\phi(s_{j_a})$ be denoted by $\phi_{s_{j_a}}$. Similarly, Bob can choose $S_b = \{t'_1, \cdots, t'_{m_b}, s'_1, \cdots, s'_{n_b}\}$; $t'_1, \cdots, t'_{m_b} \in H$ and $s'_1, \cdots, s'_{n_b} \in K$.

2. Alice makes a random element, say, $(h_H, s_H) \in G_1$ and a sequence of elements $\{\phi_{s_{j_a}}(h_1), \phi_{s_{j_a}}(h_{i_a+1}), \cdots, \phi_{s_{j_a}}(h_{i_a+3}), \cdots, \phi_{s_{j_a}}(h_{i_a})\}_{j_a=1}^{n_a}$, public, where $\phi_{s_{j_a}}(h_{i_a+1}), \phi_{s_{j_a}}(h_{i_a+2}), \phi_{s_{j_a}}(h_{i_a+3}), \cdots$ are extra additions at random positions known only by her, to the sequence $\{\phi_{s_{j_a}}(h_1), \cdots, \phi_{s_{j_a}}(h_{i_a})\}_{j_a=1}^{n_a}$, which maps the generating elements chosen for H.

3. Bob can encrypt a message which is represented in the form of an element of $H \times K$, say (h_M, s_M), using Alice's public key and θ. Here, $h_M \neq e_H$ $(\in H)$ and $s_M \neq e_K$ $(\in K)$.

4. *Encryption:* Bob computes the inverse of the message vertex, which is, $(\theta_{s_M^{-1}}(h_M^{-1}), s_M^{-1})$. Then, $(\theta_{s_M^{-1}}(h_M^{-1}), s_M^{-1})\,(h_H, s_H) = (\theta_{s_M^{-1}}(h_M^{-1})\,\theta_{s_M^{-1}}(h_H), s_M^{-1} \cdot s_H) = (h_C, s_C)$ is calculated, where (h_C, s_C) is the required cipher-text.

5. Bob sends the encrypted message (cipher-text), (h_C, s_C) together with the sequence $\{\theta_{s_M}(\phi_{s_{j_a}}(h_1)), \theta_{s_M}(\phi_{s_{j_a}}(h_{i_a+1})), \cdots, \theta_{s_M}(\phi_{s_{j_a}}(h_{i_a+3})), \cdots, \theta_{s_M}(\phi_{s_{j_a}}(h_{i_a}))\}_{j_a=1}^{n_a}$ in the same order Alice has sent.

6. *Decryption:* Upon receiving the encrypted message, Alice first computes $\{\theta_{s_M}(h_1), \cdots, \theta_{s_M}(h_{i_a})\}$. Thereafter, she uses these values to obtain, $(\theta_{s_M}(h_C), s_C \cdot s_H^{-1}) = (h_R, s_M^{-1})$ and $h_H \cdot h_R^{-1} = h_M$. The original message sent by Bob is (h_M, s_M).

3.2 Encryption Protocol 2:

1. Let Alice choose an irredundant generating set for H, say, $H = \langle h_1, \cdots, h_{i_a} \rangle$.

2. She randomly selects two elements $(h, s), (h', s') \in H \times K$ such that, $(h, s)(h', s') = (h \cdot \phi_s(h'), s \cdot s') = (e_H, e_K)$.

3. Next, she makes (h,s) (or (h',s')) public, together with a sequence of elements $\{\phi_s(h_1), \phi_s(h_{i_a+1}), \cdots, \phi_s(h_{i_a+3}), \cdots, \phi_s(h_{i_a})\}$, where $\phi_s(h_{i_a+1}), \phi_s(h_{i_a+2}), \phi_s(h_{i_a+3}), \cdots$ are extra elements added same as in the above scheme.

4. Bob encrypt a message, say (h_M, s_M), by computing $(h_M, s_M)(h,s)$.
$$(h_M, s_M)(h,s) = (h_M \cdot \theta_{s_M}(h), s_M \cdot s) = (h_C, s_C)$$

5. Bob sends the encrypted message (h_C, s_C) together with the sequence $\{\theta_{s_M}(\phi_s(h_1)), \theta_{s_M}(\phi_s(h_{i_a+1})), \cdots, \theta_{s_M}(\phi_s(h_{i_a+3})), \cdots, \theta_{s_M}(\phi_s(h_{i_a}))\}$ in the same order sent by Alice.

6. Upon receiving the encrypted message, Alice first computes $\{\theta_{s_M}(h_1), \cdots, \theta_{s_M}(h_{i_a})\}$. Thereafter, she uses these values to obtain, $h_C \cdot \theta_{s_M}(\phi_s(h')) = h_M$ and $s_C \cdot s' = s_M$.
$$h_C \cdot \theta_{s_M}(\phi_s(h')) = h_M \cdot \theta_{s_M}(h) \cdot \theta_{s_M}(\phi_s(h')) = h_M \cdot \theta_{s_M}(h \cdot \phi_s(h'))$$
$$= \underbrace{h_M \cdot \theta_{s_M}(\underbrace{h \cdot \phi_s(h')}_{e_H})}_{e_H} = h_M \quad \text{and} \quad s_C \cdot s' = s_M \cdot \underbrace{s \cdot s'}_{e_K} = s_M$$

3.3 Security Analysis and Discussion

In the decryption step of the Protocol 1, in order to determine $\theta_{s_M}(h_{y_a})$ for $1 \le y_a \le i_a$, Alice can identify $g_{y_a} \in \mathbb{Z}_p \times \mathbb{Z}_p$, such that, $\phi_{s_{j_a}}(g_{y_a}) = h_{y_a}$ for all j_a, y_a, and calculate, $\theta_{s_M}(\phi_{s_{j_a}}(g_{y_a})) = \theta_{s_M}(h_{y_a})$. This follows similarly for Protocol 2, where she can determine $g_{y_a} \in \mathbb{Z}_p \times \mathbb{Z}_p$, such that, $\phi_s(g_{y_a}) = h_{y_a}$ for all y_a, and calculate, $\theta_{s_M}(\phi_s(g_{y_a})) = \theta_{s_M}(h_{y_a})$.

When Bob takes the product of (h_H, s_H) with $(\theta_{s_M}^{-1}(h_M^{-1}), s_M^{-1})$ (in Protocol 1),
$$(\theta_{s_M}^{-1}(h_M^{-1}), s_M^{-1})(h_H, s_H) = (\theta_{s_M}^{-1}(h_M^{-1}) \cdot \theta_{s_M^{-1}}(h_H), s_M^{-1} \cdot s_H)$$
$$= (\theta_{s_M}^{-1}(h_M^{-1}) \cdot \theta_{s_M}^{-1}(h_M \cdot h_R), \underbrace{s_M^{-1} \cdot s_H}_{s_C}) \quad (\text{since } \theta_{s_M^{-1}}(h_H) = \theta_{s_M}^{-1}(h_H))$$
$$= (\theta_{s_M}^{-1}(h_M^{-1}) \cdot \theta_{s_M}^{-1}(h_M) \cdot \theta_{s_M}^{-1}(h_R), s_C) \text{ (by the properties of homomorphisms)}$$
$$= (\theta_{s_M}^{-1}(\underbrace{h_M^{-1} \cdot h_M}_{e_H}) \cdot \theta_{s_M}^{-1}(h_R), s_C) \quad \text{(by the properties of homomorphisms)}$$
$$= (\theta_{s_M}^{-1}(h_R), s_C) = (h_C, s_C)$$

By the Lemma 1, it is proven that the sequences chosen to be public do not compromise security, when made public. A third party can not determine the choice of the generating sets nor the $\phi_{s_{j_a}}$'s (due to the properties given in premises in Lemma 1 [37]), even if he try to check all possible generating elements and homomorphisms (ϕ's).

That is, for any $\phi_{s_{j_a}}(g) = h_{y_a}$, there exists a $\theta : K \to \mathrm{Aut}(H)$, such that, $\theta_{s^l}(f_l) = h_{y_a}$, thus making it impossible to determine ϕ. All the possible homomorphisms, ϕ's satisfy this condition, for each automorphism denoted by each value of l. Moreover, for any $\phi_{s_{j_a}}(g) = h_{y_a}$, there exists $\phi_{s_{j_a}^k}(f_k) = h_{y_a}$, for all k, which also creates further hardness in determining which generating elements, and ϕ might have been chosen by Alice.

It is interesting to notice that, multiplication of matrices corresponding to two linear transformations are abelian if they are present as diagonal matrices. This implies that the corresponding automorphisms are abelian (see Example 1 to identify how linear transformations could be used). Being abelian (due to the presence of diagonal matrices or

otherwise), contributes in adding a significant simplification to the decryption step in our proposed protocols, where Alice can determine θ_{s_M} using the abelian-ness of θ and ϕ. For an element, $\theta_{s_M}(\phi_{s_{j_a}}(h_{y_a}))$, Alice can compose by $\phi_{s_{j_a}}^{-1}$,

$\phi_{s_{j_a}}^{-1}(\theta_{s_M}(\phi_{s_{j_a}}(h_{y_a})))$ and simplify using the abelian property as follows.

$$\phi_{s_{j_a}}^{-1}(\theta_{s_M}(\phi_{s_{j_a}}(h_{y_a}))) = \theta_{s_M}(\phi_{s_{j_a}}^{-1}(\phi_{s_{j_a}}(h_{y_a}))) = \theta_{s_M}(\underbrace{\phi_{s_{j_a}}^{-1}(\phi_{s_{j_a}}}_{\phi_{e_K}}(h_{y_a}))) = \theta_{s_M}(h_{y_a})$$

Since h_{y_a}'s are generating elements of H and $\phi_{s_{j_a}}$'s are automorphisms of H, $\phi_{s_{j_a}}(h_{y_a})$'s are also generating elements of H. Then, if an eavesdropper knows $\phi_{s_{j_a}}(h_{y_a})$'s and the corresponding $\theta_{s_M}(\phi_{s_{j_a}}(h_{y_a}))$'s, he can use it to determine θ_{s_M}. Therefore, in the public-key sequences, it is required that $\{\phi_{s_{j_a}}(h_1),\cdots,\phi_{s_{j_a}}(h_{i_a})\}_{j_a=1}^{n_a}$, which are values corresponding to generating elements, are not distinguishable from the extra additions, $\phi_{s_{j_a}}(h_{i_a+1}),\phi_{s_{j_a}}(h_{i_a+2}),\phi_{s_{j_a}}(h_{i_a+3}),\cdots$. The platform group for implementation of the cryptographic schemes has to be chosen suitably, satisfying this requirement as well.

But if the orders of the generating elements in different possible generating sets are different, it will further help to hide which type of a generating set was chosen by Alice. As an example, consider the presence of two irredundant generating sets $S_1=\{s_1,s_2\}$, where $|s_1|=|s_2|=3$ and $S_2=\{s,t\}$, where $|s|=3,|t|=p$, for a semidirect product $(\mathbb{Z}_p\times\mathbb{Z}_p)\rtimes_\phi\mathbb{Z}_3$. There, if only one type of generating set, say S_2 was available, then even though an order p element out of the set of order p elements say, $t_1,t_2,\cdots$ will be chosen for t randomly, the possible elements could be guessed by identifying the order p elements from the public sequence. But this vulnerability is reduced when several different sets like S_1,S_2 are present.

Alice can also choose an arbitrary automorphism like $\phi_{s_{j_a}^k}$ (in Protocol 1) corresponding to her ϕ, rather than being restricted to the values obtained by $\phi_{s_{j_a}}$'s: $\{\phi_{s_{j_a}^k}(h_1),\phi_{s_{j_a}^k}(h_{i_a+1}),\cdots,\phi_{s_{j_a}^k}(h_{i_a+3}),\cdots,\phi_{s_{j_a}^k}(h_{i_a})\}_{j_a=1}^{n_a}$. This will make it more difficult for an eavesdropper to determine ϕ.

Even though we have used as examples to explain our ideas more clearly, the groups $(\mathbb{Z}_p\times\mathbb{Z}_p)\rtimes_\phi\mathbb{Z}_q$, where p,q are distinct primes are not suitable platforms for our proposals. The reason is that they offer efficient representations as matrix groups and hence are vulnerable to the algebraic span cryptanalysis. Another weakness is that, since all the elements except the identity in $\mathbb{Z}_p\times\mathbb{Z}_p$ can be used in generating sets as generating elements, when the sequence $\{\theta_{s_{j_a}}(h_1),\theta_{s_{j_a}}(h_{i_a+1}),\cdots,\theta_{s_{j_a}}(h_{i_a+3}),\cdots,\theta_{s_{j_a}}(h_{i_a})\}_{j_a=1}^{n_a}$ is made public, an eavesdropper can attempt to arbitrarily choose a suitable set and use it to determine θ_{s_M}.

4. USING HAMILTONIAN PATHS AND CYCLES

The above Protocol 1 can also be implemented by taking (h_H,s_H) to be the ending vertex of a Hamiltonian path or any random path in a Cayley graph of the corresponding group, while the Protocol 2 can be implemented by considering a Hamiltonian cycle or any random cycle. Even though the use of paths and cycles in Cayley graphs do not indicate a specific advantage for the above protocols, we illustrate the concept below with the supposition that it will be a useful insight and an initiation for future studies where the use of paths in Cayley graphs present actual benefits.

Generally, in Cayley graphs where the existence of a Hamilton cycle or a path have been proven, the cycle or the path can be written using mathematical formulae.

Example 1. Consider a group of order $3p^2$, $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_\phi \mathbb{Z}_3$. An irredundant generating set is $\{s, h_1\}$, with $|s| = 3, |h_1| = p$, where the action of s on $\mathbb{Z}_p \times \mathbb{Z}_p$ can be defined as follows [1].

Define a linear transformation T on $\mathbb{Z}_p \times \mathbb{Z}_p$ by $T(h) = s^{-1}hs$. Let $m(x)$ be the minimal polynomial of T and $h_2 = T(h_1) = s^{-1}h_1 s$. Since $|s| = 3$, $T^3 = I$, so $m(x)$ divides $x^3 - 1 = (x-1)(x^2+x+1)$ (Refer [1], for a complete argument on determining that $m(x) = x^2 + x + 1$). With respect to a basis $\{h_1, h_2\}$ of $\mathbb{Z}_p \times \mathbb{Z}_p$, T can be defined as $T(x_1, x_2) = (-x_2, x_1 - x_2)$. That is, $T({h_1}^{x_1}{h_2}^{x_2}) = s^{-1}({h_1}^{x_1}{h_2}^{x_2})s = {h_1}^{-x_2}{h_2}^{x_1-x_2}$ (considering the rational cannonical form, using $m(x)$). Using T corresponds to using ϕ_s for computations, because $\mathrm{Aut}(\mathbb{Z}_p \times \mathbb{Z}_p) \cong GL_2(\mathbb{Z}_p)$, where $GL_2(\mathbb{Z}_p)$ is the general linear group of 2×2 matrices over $\mathbb{Z}_p$.

A Hamiltonian cycle in the corresponding Cayley graph of $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_\phi \mathbb{Z}_3$ is [1], $([({h_1}^{-1})^{3j-1}, s, (h_1)^{-3j-1}, s^{-1}]_{j=1}^{\frac{(p-1)}{2}}, [(h_1)^{\frac{(p-5)}{2}}, s^{-1}, {h_1}^{\frac{(p+1)}{2}}, s], [{h_1}^{p-1}, s^{-1}], [{h_1}^{3j-1}, s, ({h_1}^{-1})^{-3j-1}, s^*]_{j=\frac{-(p-1)}{2}}^{-k-1}, [({h_1}^{-1})^{3j-1}, s^{-1}, {h_1}^{-3j-1}, s^{-1}]_{j=-k}^{-2}, [({h_1}^{-1})^{p-4}, s^{-1}, t^2, s])$, where $p = 3k+1$ or $p = 3k+2$, and $s^* = s$ or $s^* = s^{-1}$ based on the value of j.

It is clear that using ϕ, p, q, s and h_1, a Hamiltonian cycle can be written based on the mathematical proof, without generating the graph. Similarly, any random cycle or path of any length can be mathematically written once ϕ, p, q, s and h_1 are known. Hence, in the cryptographic protocols, we propose to obtain the Hamiltonian paths/cycles (or any random paths/cycles) via the mathematical proofs without generating the Cayley graphs.

The difficulty of determining the Hamiltonian paths or cycles chosen by the communicating parties corresponds to the difficulty of determining ϕ and the generating set chosen by each individual party.

Premise 1. [37] When obtaining Hamiltonian paths using the mathematical proofs as we have suggested, it is clear that a person need to know ϕ and suitable generating elements used in defining ϕ_s (or $\phi_{s_{j_a}}$'s; $1 \leq j_a \leq n_a$). The vertex obtained as the ending vertex of a Hamiltonian path is different for different choices of the generating elements.

By Lemma 1 and Premise 1, it is proven that publishing of the ending vertex of the Hamiltonian path or the random path chosen does not reveal the path chosen, when ϕ and the generating elements are kept as secrets. In fact, had the choice been a random path rather than a Hamilton path, the problem is even harder, due to the presence of many random paths ending at the same vertex (more than Hamiltonian paths).

Suppose $X_a = Cay(H \rtimes_\phi K, S_a)$ is Cayley graph of Alice and $X_b = Cay(H \rtimes_\theta K, S_b)$ is Cayley graph of Bob, with respect to the generating sets S_a, S_b mentioned previously. In Protocol 1, as the public element (h_H, s_H), Alice can choose the ending vertex of a Hamiltonian path or any random path (starting at identity vertex) in the Cayley graph (if the path doesn't start at identity vertex, product of the generating elements representing the path will not be equal to (h_H, s_H) and in some cases the ending vertex might even be (e_H, e_K) and these deviations will have to be taken in to account when doing computations). Thereafter, steps of the protocol can be followed similarly as proposed above.

In Protocol 2, Alice can determine a Hamiltonian cycle or any random cycle and take

the product of some consecutive elements along the cycle as (h,s) and the product of the remaining consecutive elements as (h',s'). Since it is a cycle, $(h,s)(h',s') = (e_H, e_K)$ is satisfied. The rest of the steps can be followed similarly.

As seen by Example 1, it is possible to obtain Hamiltonian paths/cycles using the mathematical proofs without generating the graphs, if H, K and ϕ are known. This is an advantage, with respect to both efficiency and security of the protocols (if paths in a graphs are to be used). However, a naïve attack, which is, checking all possible homomorphisms (ϕ's) for known H, K, and attempting to identify ending vertices of all possible Hamiltonian paths in Cayley graphs is possible. If there is only one generating set rather than a randomly chosen one, the ending vertices will be unique and identifiable. Hence, the cryptographic schemes are made secure against this attack by the random choice of the generating sets and keeping the generating elements and automorphisms as secrets.

5. RESISTANCE TO ALGEBRAIC SPAN CRYPTANALYSIS

As mentioned previously, many cryptosystems in the literature were identified to be unshielded from cryptanalytic techniques introduced by various scholars. In our study, we have focused our attention on the "algebraic span cryptanalysis" [32] that had been evolved based upon previous techniques such as Cheon-Jun method [38] and Tsaban's linear centralizer method [39] and is the most prominent attack applicable to our protocols. It is widely applicable since it can cryptanalyze any system where the platform group used for the protocol has efficient and faithful representations as matrix groups and the problem could be reduced to a system of linear equations using matrix pairs g and $f = g^z$, once g and f are known. This technique uses spans of algebras to generate solutions.

V. Roman'kov [35] has proposed the use of a system of the form $f = (cg)^z$, where f and g are known, so that it will not reduce to a system of linear equations. The author had used a notion of marginal sets for the proposal. The method of using a "salt" γ to offer resistance against conjugacy search in [40] also quite resembles this approach in the way of utilizing a product of elements cg, γB to hide the original g and B.

Suppose F is a free group on a countably infinite set $\{x_1, x_2, \cdots\}$ and W is a non-empty subset of F. For $w = w(x_1, \cdots, x_n) \in W$, and $g_1, \cdots, g_n \in G$, where G is a group, the value of the word w at $(g_1, \cdots, g_n)$ is defined to be $w(g_1, \cdots, g_n)$.

Definition 4. [35] For $n \in \mathbb{N}$, let $w = w(x_1, \cdots, x_n)$ be a group word, G be a group and $\bar{g} = (g_1, \cdots, g_n)$ be a tuple of elements of G. A tuple $\bar{c} = (c_1, \cdots, c_n) \in G^n$ is said to be a "marginal tuple" determined by w and $\bar{g}$ if, $w(c_1 g_1, \cdots, c_n g_n) = w(g_1, \cdots, g_n)$. This is denoted by $\bar{c} \perp w(\bar{g})$.

Moreover, a marginal set $\bar{C} \subseteq G^n$ with respect to w and $\bar{g}$, denoted by $\bar{C} \perp w(\bar{g})$, is a set where, $\bar{c} \perp w(\bar{g})$ for every $\bar{c} \in \bar{C}$. In [35], also discussed is a universal method of obtaining a marginal set $\bar{C}$ by a word w, while there are numerous ways to compute such sets. Consider $w = w(a_1, \cdots, a_k) = a_1 a_2 \cdots a_k$, $a_i \in G$, $1 \leq i \leq k$, which is any expression in straight form of a fixed element $f \in G$. It is possible that $a_i = a_j$ or $a_i = a_j^{-1}$ for $i \neq j$. This word is non-reducible.

$$x_1 a_1 x_2 a_2 \cdots x_k a_k = f \tag{1}$$

Every solution of Eq. (1) can be included in a marginal set $\bar{C}$. We can fix i and choose any value $x_j = c_j$, $j \neq i$, $c_j \in G$. Then, the solution of Eq. (1) is,

$$x_i = a_{i-1}^{-1} c_{i-1}^{-1} \cdots a_1^{-1} c_1^{-1} f a_k^{-1} c_k^{-1} \cdots a_{i+1}^{-1} c_{i+1}^{-1} \tag{2}$$

Solutions of Eq. (1) can be computed by a sequence of the following random elementary insertions. Let $(c_1, \cdots, c_k)$ be a solution of Eq. (1). For any i and any random element $d \in G$, c_i can be changed to $c_i' = c_i a_i d a_i^{-1}$ and c_{i+1} to $c_{i+1}' = d c_{i+1}$. This results in a new solution of Eq. (1). Continuing this process with random d and i, a series of new solutions of Eq. (1) can be obtained. Employing $(c_1 a_1, \cdots, c_k a_k)$ instead of $(a_1, \cdots, a_k)$ offer security to an algorithm while not creating the slightest change to the results of computations since $w(c_1 g_1, \cdots, c_n g_n) = w(g_1, \cdots, g_n)$.

All the above proposals were made by us independently before reading and understanding the literature [32, 35]. However, if the platform group chosen to implement the schemes has an efficient matrix representation they are vulnerable to the algebraic span cryptanalysis. Therefore, we suggest modified computations integrating the proposal in [35] to overcome this attack as below. Suppose all the notations and variables are same as in the original proposal in Section 3.

5.1 Modified Encryption Protocol 1

1. Alice chooses an irredundant generating set for H, say, $H = \langle h_1, \cdots, h_{i_a} \rangle$ and makes an element $(h_H, s_H) \in G_1$ public.
2. Bob considers the message vertex (h_M, s_M) and computes the inverse; $(\theta_{s_M^{-1}}(h_M^{-1}), s_M^{-1})$.

   Then, $(\theta_{s_M^{-1}}(h_M^{-1}), s_M^{-1})\,(h_H, s_H) = (\theta_{s_M^{-1}}(h_M^{-1})\,\theta_{s_M^{-1}}(h_H), s_M^{-1} \cdot s_H) = (h_C, s_C)$ is calculated and (h_C, s_C) is sent to Alice.
3. Alice determines $elt_1, \cdots, elt_L$ such that, $h_C = elt_1^{r_1} elt_2^{r_2} \cdots elt_L^{r_L}$, and a set of marginal tuples $\bar{C}$ such that for $\bar{c} = (c_1, c_2, \cdots, c_x) \in \bar{C}$,

   $$h_C = \underbrace{c_1 \cdot elt_1 \cdot c_2 \cdot elt_1 \cdots c_{r_1} \cdot elt_1}_{r_1 - terms} \cdots \underbrace{c_{x-(r_L-1)} \cdot elt_L \cdot c_{x-(r_L-2)} \cdot elt_L \cdots c_x \cdot elt_L}_{r_L - terms}.$$
4. She makes $\{elt_1, elt_1, elt_{L+1}, \cdots, elt_{L+3}, \cdots, elt_L\}$, where $elt_{L+1}, elt_{L+2}, elt_{L+3}, \cdots$ are extra elements added at random positions known only by her and a modified marginal set $\bar{C}'$, where extra elements are added to each marginal tuple at the same corresponding positions, public. A modified marginal tuple $\bar{c}'$ would be like,

   $\bar{c}' = (c_1, c_2, c_{x+1}, \cdots, c_{x+3}, \cdots, c_x)$, with the extra elements $c_{x+1}, c_{x+2}, c_{x+3}, \cdots$.
5. Bob chooses a random $\bar{c}' \in \bar{C}'$ and first calculates, $\{c_1 \cdot elt_1, c_2 \cdot elt_1, c_{x+1} \cdot elt_{L+1}, \cdots, c_{x+3} \cdot elt_{L+3}, \cdots, c_x \cdot elt_L\}$.

   Next, he obtains $\{\theta_{s_M}(c_1 \cdot elt_1), \theta_{s_M}(c_2 \cdot elt_1), \theta_{s_M}(c_{x+1} \cdot elt_{L+1}), \cdots, \theta_{s_M}(c_{x+3} \cdot elt_{L+3}), \cdots, \theta_{s_M}(c_x \cdot elt_L)\}$ using the inverse of $\theta_{s_M^{-1}}$ he used in encrypting the message and transmits the sequence to Alice in the same order she has sent.
6. Now Alice identifies the extra elements by their positions in the sequence, discards them and uses the remaining elements to compute, $\theta_{s_M}(c_1 \cdot elt_1) \cdot \theta_{s_M}(c_2 \cdot elt_1) \cdots \theta_{s_M}(c_x \cdot elt_L) = \theta_{s_M}(c_1 \cdot elt_1 \cdot c_2 \cdot elt_1 \cdots c_x \cdot elt_L) = \theta_{s_M}(h_C) = h_R$ and $h_H \cdot h_R^{-1} = h_M$.

The determination of the y-coordinate, s_M is same as in the initial proposal since no modification was suggested by us.

5.2 Modified Encryption Protocol 2

The modified steps for this protocol basically follow the above suggestion.

1. Let Alice choose an irredundant generating set for H, say, $H = \langle h_1, \cdots, h_{i_a} \rangle$ and two elements $(h,s),(h',s') \in H \times K$ such that, $(h,s)(h',s') = (h \cdot \phi_s(h'), s \cdot s') = (e_H, e_K)$. She makes one element, say (h,s) public.
2. As in the Protocol 1, Alice determines $elt_1, \cdots, elt_L$ such that, $\phi_s(h') = elt_1^{r_1} elt_2^{r_2} \cdots elt_L^{r_L}$,
and a set of marginal tuples $\bar{C}$ such that, $\phi_s(h') = \underbrace{c_1 \cdot elt_1 \cdot c_2 \cdot elt_1 \cdots c_{r_1} \cdot elt_1}_{r_1-terms} \cdots$
$\underbrace{c_{x-(r_L-1)} \cdot elt_L \cdot c_{x-(r_L-2)} \cdot elt_L \cdots c_x \cdot elt_L}_{r_L-terms}$; $\bar{c} = (c_1, c_2, \cdots, c_x) \in \bar{C}$. She makes $\{elt_1, elt_1, elt_{L+1}, \cdots, elt_{L+3}, \cdots, elt_L\}$, where $elt_{L+1}, elt_{L+2}, elt_{L+3}, \cdots$ are extra additions at random positions and a modified marginal set $\bar{C}'$, where extra elements are added to each marginal tuple at the same positions, public as well. A modified marginal tuple $\bar{c}' \in \bar{C}'$ will be, $\bar{c}' = (c_1, c_2, c_{x+1}, \cdots, c_{x+3}, \cdots, c_x)$, with the extra elements $c_{x+1}, c_{x+2}, c_{x+3}, \cdots$.
3. Bob encrypts a message, say (h_M, s_M), by computing $(h_M, s_M)(h,s)$.
$$(h_M, s_M)(h,s) = (h_M \cdot \theta_{s_M}(h), s_M \cdot s) = (h_C, s_C).$$
4. Moreover, Bob chooses a random $\bar{c}' \in \bar{C}'$ and calculates, $\{c_1 \cdot elt_1, c_2 \cdot elt_1, c_{x+1} \cdot elt_{L+1}, \cdots, c_{x+3} \cdot elt_{L+3}, \cdots, c_x \cdot elt_L\}$. Next, he obtain $\{\theta_{s_M}(c_1 \cdot elt_1), \theta_{s_M}(c_2 \cdot elt_1), \theta_{s_M}(c_{x+1} \cdot elt_{L+1}), \cdots, \theta_{s_M}(c_{x+3} \cdot elt_{L+3}), \cdots, \theta_{s_M}(c_x \cdot elt_L)\}$ using the θ_{s_M} he used in encrypting the message and transmits to Alice together with (h_C, s_C).
5. Now Alice identifies the required elements from the sequence and computes, $\theta_{s_M}(c_1 \cdot elt_1) \cdot \theta_{s_M}(c_2 \cdot elt_1) \cdots \theta_{s_M}(c_x \cdot elt_L) = \theta_{s_M}(c_1 \cdot elt_1 \cdot c_2 \cdot elt_1 \cdots c_x \cdot elt_L) = \theta_{s_M}(\phi_s(h'))$ and $h_C \cdot \theta_{s_M}(\phi_s(h')) = h_M$. The value of s_M is obtained by the same computations stated in the original proposal.

6. COMPUTATIONAL COST

As mentioned in analyses done in previous studies on non-abelian group based cryptosystems (e.g. [12, 41]), the time complexities as well as the ability to resist quantum cybersecurity threats are dependent on the non-abelian group utilized and the representation of the group (that is, whether it is represented as a permutation group or a matrix group etc.) in addition to the steps of the algorithm. Following the notes in the literature, in this section we discuss computational costs viable for our protocols.

6.1 Encryption Protocol 1

Based on the platform groups G_1, G_2, chosen by each individual and their representations, there is a cost of determining the irredundant generating sets for each G_1 and

G_2. There can be many generating sets since there can be many choices for generating elements (as also explained in the proof of Lemma 1 [37]). If G_1 and G_2 have m_1 and m_2 number of generating sets respectively, then there is $O(m_1)$ and $O(m_2)$ costs for Alice and Bob respectively.

Next, both of them have to select an irredundant generating set for H out of the possible irredundant generating sets for H. Suppose there are m_3 such possible sets. Then there are $O(m_3)$ operations for this choice (eventhough only Alice's choice of generating elements will be used in her public-key sequence in the instances where Bob is the sender of messages, he also has to have a chosen set of generating elements for H in order to conduct computations such as $\theta_{s_M^{-1}}(h_M^{-1})$ etc.).

The cost for choosing a random element (h_H, s_H) by Alice is related to the order of G_1; $O(|G_1|)$, while there is a cost of applying $\phi_{s_{ja}}$'s on any element which is based on the platform group chosen. Particularly, $\phi_{s_{ja}}$'s denote automorphisms of H and hence the cost is dependent on what the group H is. In our protocols, $\phi_{s_{ja}}$'s correspond to conjugations. Therefore, applying $\phi_{s_{ja}}$'s amount to just two multiplications of elements in G_1.

If powers of automorphisms such as $\phi_{s_{ja}^k}$'s were chosen to be applied, then the computation of powers of $\phi_{s_{ja}}$'s corresponds to exponentiation of two elements namely, the conjugating element and its inverse. The exponentiation of elements can be computed by the "square and multiply" method same as in the standard Diffie-Hellman protocol (also mentioned in [41]). The computational cost for the application of $\phi_{s_{ja}}$'s or $\phi_{s_{ja}^k}$'s to Alice's choice of generating elements for H and extra random elements added is also dependent on the number of elements in the generating set for H, that is, i_a and the number of random elements added, say x. That is, $O(i_a) + O(x)$.

For Bob to compute the inverse of the message vertex and multiply with Alice's public-key vertex to generate the cipher-text (h_C, s_C), there is an associated cost, again based on the platform group chosen and its representation. And the computational cost for the application of θ_{s_M} to the sequence of elements corresponds to $O(i_a) + O(x)$.

Upon the receival of cipher-text, Alice has to compute $\{\theta_{s_M}(h_1), \cdots, \theta_{s_M}(h_{i_a})\}$. There, checking for suitable values g_{y_a}'s such that $\phi_{s_{ja}}(g_{y_a}) = h_{y_a}$'s is an application of automorphisms to elements and will eventually reveal the values of $\theta_{s_M}(h_{y_a})$'s. This could easily be achieved through multiplication of suitable terms in $\{\theta_{s_M}(\phi_{s_{ja}}(h_1)), \cdots, \theta_{s_M}(\phi_{s_{ja}}(h_{i_a}))\}_{ja=1}^{n_a}$. For example, if $\phi_{s_{ja}}(h_1^y) = h_1$, then $\theta_{s_M}(\phi_{s_{ja}}(h_1))^y = \theta_{s_M}(h_1)$ and could be computed using the "square and multiply" method.

The decryption step require computation of $\theta_{s_M}(h_C)$ that could be done through multiplication of suitable elements from $\{\theta_{s_M}(h_1), \cdots, \theta_{s_M}(h_{i_a})\}$ whose product will form $\theta_{s_M}(h_C)$. The calculation of the inverse h_R^{-1} and the two multiplications $h_H \cdot h_R^{-1}$, $s_C \cdot s_H^{-1}$, involve computational costs that are dependent on the platform group.

6.2 Encryption Protocol 2

The selection of two elements (h, s), (h', s') and checking whether the product $(h, s)(h', s')$ is equal to (e_H, e_K) necessitates an additional cost. It also includes the cost of application of automorphism ϕ_s on h' prior to obtaining the product with h. These costs depend on the group chosen and its representation as mentioned previously.

The cost of computing $h_C \cdot \theta_{s_M}(\phi_s(h'))$ is quite similar to that in decryption step of Protocol 1 and lacks the requirement to compute an inverse of an element. The costs

associated with all the remaining steps are similar to that of Protocol 1.

6.3 Using Paths and Cycles in Cayley Graphs

If Hamiltonian paths/cycles or any random paths/cycles are to be used, an additional cost to generate the particular paths/cycles are associated, based on mathematical proofs for the relevant group and of course, the type of the platform group and its representation. Protocols could be programmed to have the starting vertex of paths as (e_H, e_K) to minimize extra costs such as, in having to execute a loop again and again if the ending vertex would be (e_H, e_K) until an ending vertex that is not equal to identity will be obtained.

Use of cycles have lower computational cost in comparison with the Protocol 2 for the step of checking whether $(h,s)(h',s')$ is equal to (e_H, e_K), since in a cycle such a checking is not required. The reason is that, in cycles the product of any set of consecutive elements with the next set of consecutive elements is always (e_H, e_K). Complexities of the remaining steps of the algorithms are similar to that described above under each protocol.

6.4 Modified Encryption Protocol 1 and 2

When considering the modified protocols, a computational cost is aggregated to determine elements $\{elt_1, \cdots, elt_L\}$, compute marginal and modified marginal tuples, apply θ_{s_M} to a sequence of products of elements and obtain the respective products, all of which are dependent on the platform group chosen. If the number of marginal tuples in the set $\bar{C}'$ is m_4, then selection of a random tuple $\bar{c}'$ by Bob is $O(m_4)$. The calculations $\theta_{s_M}(c_1 \cdot elt_1) \cdot \theta_{s_M}(c_2 \cdot elt_1) \cdots \theta_{s_M}(c_x \cdot elt_L) = \theta_{s_M}(h_C)$, $h_H \cdot h_R^{-1}$ and $h_C \cdot \theta_{s_M}(\phi_s(h'))$ are concerned with computation of inverse of elements and product of elements as in the previous descriptions.

In comparison to the improved Anshel-Anshel-Goldfeld scheme proposed in [35] which is the currently existing protocol resistant to the algebraic span cryptanalysis, our protocols involve more steps and hence could be considered to associate higher costs.

Throughout the paper, we had discussed all the applications with respect to irredundant generating sets but they can be applied for any generating set following the same manner. In fact, the use of any generating set is more complicated and hence is more secured than considering the irredundant sets.

7. CONCLUSION AND FUTURE STUDIES

Non-abelian group based Cryptography is a latest attraction for innovative research on cryptosystems. In this paper, we have proposed new cryptographic protocols based on an intractable problem of determining automorphisms and generating elements of a group. Whereas most of the previous non-abelian group based cryptographic schemes were proven to be vulnerable, this could be a suggestion of a pathway on new further researches for secure schemes. The relation of paths and cycles in Cayley graphs to this problem implies that the HPP and HCP in Cayley graphs of non-abelian groups and in fact the difficulty of determining random paths/cycles in Cayley graphs can also be considered in developing cryptographic protocols. This can be regarded as the first study, where paths and cycles in Cayley graphs were suggested to be generated by considering the abstract properties of the graphs via mathematical proofs rather than having to generate the graphs.

Moreover, the ability to use two different groups by the two communicating parties while using elements from the same underlying set $H \times K$ is a special advantage.

In our proposals, an eavesdropper has the ability to easily determine the y-coordinate, s_M, even though the x-coordinate is proven to be secured. Hence, it would be interesting to focus future studies in developing techniques to protect the y-coordinates involved in encryption. Furthermore, suitable platforms for the implementation of these protocols and further improvements to the security could be explored and proposed. Novel protocols utilizing the Hamilton paths/cycles in a more beneficial manner may also be investigated.

REFERENCES

1. K. Kutnar, D. Marusic, D. W. Morris, J. Morris, and P. Sparl, "Hamiltonian cycles in Cayley graphs whose order has few prime factors," *Ars Mathematica Contemporanea*, Vol. 5, 2010, pp. 27-71.
2. G. H. J. Lanel, H. K. Pallage, J. K. Ratnayake, S. Thevasha, and B. A. K. Welihinda, "A survey on Hamiltonicity in Cayley graphs and digraphs on different groups," *Discrete Mathematics, Algorithms and Applications*, Vol. 11, 2019, p. 1930002.
3. G. H. J. Lanel, T. M. K. K. Jinasena, and B. A. K. Welihinda, "Hamiltonian Cycles in Cayley Graphs of Semidirect Products of Finite Groups," *European Modern Studies Journal*, Vol. 4, 2020, pp. 1-19.
4. F. Maghsoudi, "Cayley graphs of order 6p are Hamiltonian," Department of Mathematics and Computer Science, University of Lethbridge, Canada, 2020.
5. E. Ghaderpour, and D. W. Morris, "Cayley graphs on nilpotent groups with cyclic commutator subgroup are Hamiltonian," *Ars Mathematica Contemporanea*, Vol. 7, 2014, pp. 55-72.
6. D. W. Morris, "Infinitely many nonsolvable groups whose Cayley graphs are Hamiltonian," *Journal of Algebra Combinatorics Discrete Structures and Applications*, Vol. 3, 2016, pp. 13-30.
7. K. Keating, and D. Witte, "On Hamilton cycles in Cayley graphs in groups with cyclic commutator subgroup," *European Modern Studies Journal*, Vol. 4, 2020, pp. 1-19.
8. D. W. Morris, "Cayley graphs on groups with commutator subgroup of order $2p$ are Hamiltonian," *arXiv Preprint*, 2017, arXiv:1703.06377.
9. D. W. Morris, and K. Wilk, "Cayley graphs of order kp are Hamiltonian for $k < 48$," *arXiv Preprint*, 2018, arXiv:1805.00149.
10. B. Fine, M. Habeeb, D. Kahrobaei, and G. Rosenberger, "Aspects of Non-Abelian group based Cryptography: a survey and open problems," *JP Journal of Algebra, Number Theory and Applications*, Vol. 7, 2011, pp. 55-72.
11. G. H. J. Lanel, T. M. K. K. Jinasena, and B. A. K. Welihinda, "A Survey of Public-Key Cryptography over Non-Abelian Groups," *International Journal of Computer Science and Network Security*, Vol. 21, 2021, pp. 289-300.
12. T. C. Lin, "A study of Non-abelian Public-key Cryptography," *International Journal of Network Security*, Vol. 20, 2018, pp. 278-290.
13. V. Roman'kov, "Two general schemes of Algebraic Cryptography," *Groups Complexity Cryptology*, Vol. 10, 2018, pp. 83-98.

14. I. Ilic, "The Discrete Logarithm Problem in Non-abelian Groups," *Computing*, Vol. 1, 2010, pp. 1.
15. L. C. Klingler, S. S. Magliveras, F. Richman, and M. Sramka, "Discrete logarithms for finite groups," *Computing*, Vol. 85, 2009, p. 3.
16. L. Gu, L. Wang, K. Ota, M. Dong, Z. Cao, and Y. Yang, "New Public-Key Cryptosystems based on Non-Abelian Factorization Problems," *Security and Communication Networks*, Vol. 6, 2013, pp. 912-922.
17. H. Hong, J. Shao, L. Wang, H. Ahmad, and Y. Yang, "Public-key Encryption in Non-Abelian Groups," *arXiv Preprint*, 2016, arXiv:1605.06608.
18. L. Gu and S. Zheng, "Conjugacy Systems based on Non-abelian Factorization Problems and their Applications in Cryptography," *Journal of Applied Mathematics*, Vol. 2014, 2014.
19. I. Anshel, M. Anshel, and D. Goldfeld, "An Algebraic method for Public-Key Cryptography," *Mathematical Research Letters*, Vol. 6, 1999, pp. 287-291.
20. 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 *Proceedings of Annual International Cryptology Conference*, 2000, pp. 166-183.
21. R. Alvarez, L. Tortosa, J. Vicent, and A. Zamora, "A Non-abelian group based on block upper triangular matrices with Cryptographic applications," in *Proceedings of International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, 2009, pp. 117-126.
22. I. Code, " Popular account of Sarah's award winning project on Public-key cryptography, co-written with her father," *A Mathematical Journey*, by Sarah Flannery and David Flannery, 2001.
23. S. Baba, S. Kotyad, and R. Teja, "A Non-Abelian Factorization Problem and an associated Cryptosystem," *IACR Cryptology ePrint Archive*, Vol. 2011, 2011, p. 48.
24. V. Shpilrain and G. Zapata, "Using the Subgroup Membership Search Problem in Public-key Cryptography," *Contemporary Mathematics*, Vol. 418, 2006, p. 169.
25. N. R. Wagner and M. R. Magyarik, "A Public-key Cryptosystem based on the Word Problem," in *Proceedings of Workshop on Theory and Application of Cryptographic Techniques*, 1984, pp. 19-36.
26. M. Garzon and Y. Zalcstein, "The Complexity of Grigorchuk groups with application to Cryptography," *Theoretical Computer Science*, Vol. 88, 1991, pp. 83-98.
27. R. I. Grigorchuk, "Degrees of growth of finitely generated groups, and the theory of invariant means," *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, Vol. 48, 1984, pp. 939-985.
28. A. Mahalanobis, "The Diffie-Hellman key-exchange protocol and non-abelian nilpotent groups," *Israel Journal of Mathematics*, Vol. 165, 2008, pp. 161-187.
29. A. Mahalanobis, "A simple generalization of the ElGamal cryptosystem to non-abelian groups," *Communications in Algebra*, Vol. 36, 2008, pp. 3878-3889.
30. A. I. S. Moldenhauer and G. Rosenberger, "Cryptosystems using automorphisms of finitely generated free groups," *arXiv Preprint*, 2016, arXiv:1603.02328.
31. S. H. Paeng, K. C. Ha, J. H. Kim, S. Chee, and C. Park, "New Public-key cryptosystem using finite Non-abelian groups," in *Proceedings of Annual International Cryptology Conference*, 2001, pp. 470-485.

32. A. Ben-Zvi, A. Kalka, and B. Tsaban, "Cryptanalysis via Algebraic Spans," in *Proceedings of Annual International Cryptology Conference*, 2018, pp. 255-274.
33. J. Ding, A. Miasnikov, and A. Ushakov, "A linear attack on a key-exchange protocol using extensions of matrix semigroups," *IACR Cryptology ePrint Archive*, Vol. 2015, 2015, p. 18.
34. V. Roman'kov, "A non-linear decomposition attack," *Groups Complexity Cryptology*, Vol. 8, 2016, pp. 197-207.
35. V. Roman'kov, "An Improved version of the AAG Cryptographic Protocol," *Groups Complexity Cryptology*, Vol. 11, 2019, pp. 35-41.
36. I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, "New key-agreement protocols in Braid group Cryptography," in *Proceedings of Cryptographers' Track at the RSA Conference*, 2001, pp. 13-27.
37. G. H. J. Lanel, T. M. K. K. Jinasena, and B. A. K. Welihinda, "Cryptographic Protocols using Semidirect Products of Finite Groups," *International Journal of Computer Science and Network Security*, Vol. 21, 2021, pp. 17-27.
38. J. H. Cheone and B. Jun, "A Polynomial time algorithm for the Braid Diffie-Hellman Conjugacy Problem," in *Proceedings of Annual International Cryptology Conference*, 2003, pp. 212-225.
39. B. Tsaban, "Polynomial-time solutions of computational problems in Noncommutative Algebraic Cryptography," *Journal of Cryptology*, Vol. 28, 2015, pp. 601-622.
40. S. K. Rososhek, "Modified matrix modular cryptosystems," *Journal of Advances in Mathematics and Computer Science*, 2015, pp. 613-636.
41. D. Kahrobaei and V. Shpilrain, "Using semidirect product of (semi) groups in Public-key Cryptography," in *Proceedings of Conference on Computability in Europe*, 2016, pp. 132-141.

Ganhewalage Jayantha Lanel is a Senior Lecturer at the Department of Mathematics, University of Sri Jayewardenepura. He has received Ph.D. in Mathematics degree from the Oakland University, Rochester, Michigan. His research interests include Computer algebra/symbolic Mathematics, Graph theory/Computational Discrete Mathematics, Queuing theory, and Operational research. His mathematical expertised areas are Computer algebra, Graph theory, Algorithmic analysis, Engineering Mathematics and Numerical analysis.

Tholka Mudalige Kasun Kosala Jinasena is a Senior Lecturer at the Department of Computer Science, University of Sri Jayewardenepura. He has received the Ph.D. degree in Computer Science from the same university and his research interests include Robotics, Embeded systems, Artificial intelligence, Image processing, Datamining, Mobile computing and Computer security.

Buddhini Angelika Kasuni Welihinda received the BS (Special) degree in Mathematics from the University of Sri Jayewardenepura, Sri Lanka. She is currently following the Ph.D. degree in Mathematics at the Department of Mathematics in the same university. Her research interests include Algebraic graph theory, Group theory and Cryptography.