Cryptographic Protocols using Semidirect Products of Finite Groups

G. H. J. Lanel[†,*], T. M. K. K. Jinasena[††] and B. A. K. Welihinda[†],

kasun@sjp.ac.lk, kasuniwe@gmail.com

[†]Department of Mathematics, University of Sri Jayewardenepura, Gangodawila, Nugegoda, Sri Lanka
[††]Department of Computer Science, University of Sri Jayewardenepura, Gangodawila, Nugegoda, Sri Lanka
*(Corresponding Author: Dr. G.H.J. Lanel ghjlanel@sjp.ac.lk)

Summary
Non-abelian group based cryptosystems are a latest research inspiration, since they offer better security due to their non-abelian properties. In this paper, we propose a novel approach to non-abelian group based public-key cryptographic protocols using semidirect products of finite groups. An intractable problem of determining automorphisms and generating elements of a group is introduced as the underlying mathematical problem for the suggested protocols. Then, 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. The applicability of Hamiltonian paths, and in fact any random path in Cayley graphs in the above cryptographic schemes and an application of the same concept to two previous cryptographic protocols based on a Generalized Discrete Logarithm Problem is discussed. Moreover, an alternative method of improving the security is also presented.

Key words:
Algebraic span cryptanalysis, Cayley graphs, Generalized Discrete Logarithm Problem, Hamiltonian Path/Cycle Problem, Non-abelian/Non-commutative, Semidirect products

1. Introduction

With a quantum computing era expected in the near future and the discovery of more and more vulnerabilities in the cryptosystems used at present, the scientists and mathematicians have drifted towards introducing various substitutes to the existing public-key cryptosystems. Cryptography using non-abelian groups is one such approach.

In fact, we have been stimulated towards non-abelian group based Cryptography, during a study on the existence of Hamiltonian cycles in Cayley graphs of finite groups. For a literature survey and a recent development related to the Hamiltonian Path Problem (HPP)/ Hamiltonian Cycle Problem (HCP) in Cayley graphs see [1]–[5]. Since the HPP/HCP in Cayley graphs over most non-abelian groups is a very difficult mathematical problem, we have developed an intuition that it will be a very suitable intractable problem to be studied in constructing novel public-key cryptosystems. Moreover, we also felt that such a system will provide higher security than the existing cryptosystems due to the non-abelian structures involved.

Thereby, after initiating the study of related literature (see [6]–[8]), we were able to identify that, indeed, many scholars have already begun to investigate regarding cryptosystems based on non-abelian groups, since the 1980's. The past researches on cryptosystems using non-abelian groups were focused on underlying problems such as Conjugacy Problem [9], Conjugacy Search Problem [10], [11], Word Problem [12], [13], Factorization and Membership Search Problems [14]–[18] etc. in non-abelian groups, but a significant application of the HCP or HPP was lacking. Most of the intractable problems are actually generalizations of some conventional cryptographic problem to non-abelian groups: e.g. the conjugacy related problems in [9], [10] show one way of generalizing the traditional Discrete Logarithm Problem (DLP) to problems over non-abelian groups.

From the numerous cryptosystems proposed in the literature based on various mathematical problems, the security of almost all were cryptanalyzed (See [8] for some of the attacks developed by various authors). Tsaban et al. [19] have presented a method of attack, generally called as the "algebraic span cryptanalysis", which can develop solutions using spans of algebras. It could break the security of most protocols which used groups that can be efficiently and faithfully represented as matrix groups. Hence, it is applicable to many cryptographic protocols. However, V. Roman'kov in [20] has introduced a method to offer resistance against this attack.

In this paper, we initially prove an intractable problem involving the homomorphisms determining the semidirect products of finite groups (i.e. related to automorphisms and generating elements of a group). Then, based on this intractable problem, we propose two novel techniques for signature protocols. We show that the HPP and HCP and in fact, the difficult problems of determining any random path or cycle in some Cayley graphs can be reduced to the same intractable problem and propose the use of Hamiltonian paths, or any random paths as a secret key in the protocols. Furthermore, we also propose an application of paths in Cayley graphs to variants of two schemes based on a Generalized Discrete Logarithm Problem (GDLP). Our proposals are resistant to the algebraic span method if the platform groups are chosen carefully and we present an improved version of the protocols integrating Roman'kov's technique, to be used otherwise.

Manuscript received: August 16, 2021
Manuscript revised: August 28, 2021
https://doi.org/10.22937/IJCSNS.2021.21.8.3

Paper outline. The rest of the paper is organized as follows. The section 2 presents a brief overview of the related fundamentals. The section 3 includes the novel signature protocol proposed. The next two sections present the use of paths in Cayley graphs, in the protocols followed by the novel application based on GDLP. In section 6, we show how the method of being secured against algebraic span cryptanalysis can be integrated to our proposals as well. Finally, the concluding remarks of our study.

2. Preliminaries

This section includes an overview of the fundamental concepts from Group Theory and Graph Theory related to this research.

Let G be the semidirect product between the normal subgroup N and the subgroup H, $G = N \rtimes_\phi H$. If both N and H are finite, $|G| = |N| \times |H|$. This follows from the fact that G is of the same order as the semidirect product of N and H, whose underlying set is $N \times H$.

For a semidirect product $G = N \rtimes_\phi H$, let $\phi: H \to \mathrm{Aut}(N)$ be a homomorphism which sends elements $h \in H$ to automorphisms ϕ_h, of N. The group law in G can be stated as, $(n,h)(n_1,h_1) = (n\phi_h(n_1), hh_1)$, where $(n,h),(n_1,h_1) \in G$. Here, $\phi_h(n) = h^{-1}nh$, for $h \in H$, $n \in N$. The (additive) identity element of G can be denoted as (e_N, e_H). The inverse of an element (n,h) is $(\phi_{h^{-1}}(n^{-1}), h^{-1})$. For a ϕ_h, $\phi_h(n) = h^{-1}nh$, and $\phi_{h^{-1}}(n) = (h^{-1})^{-1}nh^{-1} = hnh^{-1}$. Thus $\phi_{h^{-1}}(n) = \phi_h^{-1}(n)$. Furthermore, considering $\phi_h^m(n) = \underbrace{\phi_h(\phi_h(\cdots(\phi_h}_{m-times}(n)))) = h^{-m}nh^m = (h^m)^{-1}nh^m = \phi_{h^m}(n)$.

A path in a graph which visits every vertex exactly once is known as a Hamiltonian path. When there is an edge between the starting and the ending vertices, it is called a Hamiltonian cycle. Determining Hamiltonian paths or cycles in most of the graphs are considered as mathematically intractable problems and are known as the HPP and the HCP, respectively.

Definition 2.1. *A vertex-transitive graph is a graph X, where for any two vertices $v_1, v_2 \in V(X)$, there exists an automorphism of X which maps v_1 to v_2, where $V(X)$ is the set of vertices of X.*

A Cayley graph is a type of a vertex-transitive graphs. For a finite group G and a subset S of G, the Cayley graph of G with respect to S can be defined as follows, provided that $1 \notin S$ (identity element) and S is inverse closed.

Definition 2.2. *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 Hamiltonian cycle in a spanning subgraph is also a Hamiltonian cycle in the ambient graph. Hence, the identification of Hamilton paths/cycles in Cayley graphs with respect to irredundant generating sets is adequate.

Definition 2.3. *An irredundant generating set for a Cayley graph X is a generating set S such that no proper subset of S generates X.*

Let H and K be finite groups and $G = H \rtimes_\mu K$ be a semidirect product, where H is the normal subgroup. Assume that the group operations are efficiently computable.

The elements of any two groups, say $G_1 = H \rtimes_\phi K$ and $G_2 = H \rtimes_\theta K$, are the same, viz. the elements of $H \times K$. But the Cayley graphs of each with respect to their corresponding generating sets will be dissimilar. If a communication is to take place, where two communicating parties use two such groups chosen by each, both of them will perform calculations under the same $mod\ p, mod\ q$ etc., using the same elements, whereas the product of any two elements has to be computed using the two different homomorphisms ϕ and θ by each party.

When two groups like G_1, G_2 are chosen by the communicating parties, both of them as well as any third party is aware of H, K etc. and the elements of the group. But ϕ, θ defining the semidirect products, the generating sets chosen are only known by the respective party.

3. Cryptographic protocols using semidirect products

In this section, first we present the proof of the intractable problem which we consider as the basis for the proposed protocols.

Lemma 3.1. *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.*

Proof.

Premise 3.2. In a semidirect product $H \rtimes_\phi K$, $\phi: K \to \mathrm{Aut}(H)$. Suppose $K = \langle s \rangle$. Then, $\phi(e_K) = \phi_{e_K} \in \mathrm{Aut}(H)$, $\phi(s) = \phi_s \in \mathrm{Aut}(H)$, $\phi(s^2) = \phi_{s^2} \in \mathrm{Aut}(H)$, $\cdots$, $\phi(s^{|s|-1}) = \phi_{s^{|s|-1}} \in \mathrm{Aut}(H)$. Since $\phi_s \circ \phi_s = \phi_s^2 = \phi_{s^2}$, $\phi_{s^2} \circ \phi_s = \phi_s^3 = \phi_{s^3}$ etc., the entire homomorphism ϕ can be determined if ϕ_s is known (similar arguments follow for other generating sets of K, such as $K = \langle s_1, s_2, \cdots, s_n \rangle$).

Let $H = \langle h_1, h_2, \cdots, h_i \rangle$. For $\phi_{s^k} \in \mathrm{Aut}(H)$, $\phi_{s^k}: H \to H$, $1 \le k \le |s|$. If $\phi_{s^k}(h_1), \phi_{s^k}(h_2), \cdots, \phi_{s^k}(h_i)$ are known, the entire automorphism can be determined (once it is known where the generating elements are mapped, it is possible to determine the mapping of any element under any homomorphism or automorphism considered). *If generating elements and their mappings are not known the entire homomorphism or automorphism can't be determined.*

Example 3.3. *Consider the group* $(\mathbb{Z}_5 \times \mathbb{Z}_5) \rtimes_\phi \mathbb{Z}_3$. $\phi: \mathbb{Z}_3 \to \mathrm{Aut}(\mathbb{Z}_5 \times \mathbb{Z}_5)$. *Let* $\phi(\overline{0}) = \phi_0, \phi(\overline{1}) = \phi_1, \phi(\overline{2}) = \phi_2$, *and let any* $\overline{m} \in \mathbb{Z}_n$, *where* n *is a prime, be denoted by* m. *Suppose* $\phi_1(((1,0),0)) = ((a,b),0)$ *and* $\phi_1(((0,1),0)) = ((c,d),0)$.

Here, $a, b, c, d \in \mathbb{Z}_5$, $\phi_0, \phi_1, \phi_2 \in Aut(\mathbb{Z}_5 \times \mathbb{Z}_5)$ *and* $((0,0),0)$ *denote the additive identity.* $(\mathbb{Z}_5 \times \mathbb{Z}_5)$ *is the normal subgroup of* $(\mathbb{Z}_5 \times \mathbb{Z}_5) \rtimes_\phi \mathbb{Z}_3$. *Since* $\{((1,0),0), ((0,1),0)\}$ *are generating elements of* $(\mathbb{Z}_5 \times \mathbb{Z}_5)$, *once we know* $((a,b),0)$ *and* $((c,d),0)$, *the entire automorphism* ϕ_1 *can be determined. Also,* $\overline{1}$ *is a generating element of* $\mathbb{Z}_3$. *Thus, once we know* $\phi(\overline{1}) = \phi_1$, *the entire homomorphism* ϕ *can be determined.*

Premise 3.4. Since ϕ_{s^k}'s are automorphisms, there exists some $g_1, g_2, \cdots, g_i \in H$ such that, $\phi_{s^k}(g_1) = h_1, \phi_{s^k}(g_2) = h_2, \cdots, \phi_{s^k}(g_i) = h_i$, for all $\phi_{s^k} \in \mathrm{Aut}(H)$. Hence, for any $g \in H$, and $\phi_s \in \mathrm{Aut}(H)$, there exists some $f_k \in H$ such that, $\phi_s(g) = \phi_{s^k}(f_k)$, for all ϕ_{s^k}.

Moreover, for any $\phi: K \to \mathrm{Aut}(H)$, and $\theta: K \to \mathrm{Aut}(H)$, there exists automorphisms of H such that, $\phi_{s^k}(g) = \theta_{s^l}(g_l)$, for all $g \in H$ and $1 \le k, l \le |s|$. Here, $g_l \in H$ and $s, \mathfrak{s} \in K$.

Example 3.5. *When considering the automorphism* ϕ_1 *in above Example 3.3., for any* $((g_1, g_2),0) \in (\mathbb{Z}_5 \times \mathbb{Z}_5), \phi_1(((g_1, g_2),0)) = (((g_3, g_4),0))$ *for some* $((g_3, g_4),0) \in (\mathbb{Z}_5 \times \mathbb{Z}_5)$. *Also, there exists* $((g_5, g_6),0) \in (\mathbb{Z}_5 \times \mathbb{Z}_5)$ *such that* $\phi_1(((g_3, g_4),0)) = ((g_5, g_6),0)$ *as well. Therefore, for all the ordered pairs* $((g_x, g_y),0) \in (\mathbb{Z}_5 \times \mathbb{Z}_5)$, *there is some ordered pair* $((g'_x, g'_y),0) \in (\mathbb{Z}_5 \times \mathbb{Z}_5)$, *such that,* $\phi_1(((g_x, g_y),0)) = ((g'_x, g'_y),0)$.

In $\phi_1(X) = Y$, *since* ϕ_1 *is an automorphism, the set of values taken by* X *is same set as the set of values taken by* Y. *When considering,* $\phi_2(X_1) = Y_1$, *again the set of values for* X_1 *as well as for* Y_1, *is same as the set of values for* X *and* Y. *i.e. the set of elements of* $(\mathbb{Z}_5 \times \mathbb{Z}_5)$. *It follows similarly for* ϕ_0 *as well.*

Therefore, it is clear that, for any $((h,k),0) \in (\mathbb{Z}_5 \times \mathbb{Z}_5)$, *there are three elements,* $((h_0, k_0),0), ((h_1, k_1),0)$ *and* $((h_2, k_2),0) \in (\mathbb{Z}_5 \times \mathbb{Z}_5)$ *such that,* $\phi_0(((h_0, k_0),0)) = ((h,k),0), \phi_1(((h_1, k_1),0)) = ((h,k),0)$ *and* $\phi_2(((h_2, k_2),0)) = ((h,k),0)$.

Furthermore, if there are several semidirect products defined by different homomorphisms, say, $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_\phi \mathbb{Z}_q$ *defined by* $\phi: \mathbb{Z}_q \to \mathrm{Aut}(\mathbb{Z}_p \times \mathbb{Z}_p)$, $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_{\phi'} \mathbb{Z}_q$ *defined by* $\phi': \mathbb{Z}_q \to \mathrm{Aut}(\mathbb{Z}_p \times \mathbb{Z}_p)$, $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_{\phi''} \mathbb{Z}_q$ *defined by* $\phi'': \mathbb{Z}_q \to \mathrm{Aut}(\mathbb{Z}_p \times \mathbb{Z}_p)$, *etc., where* p *and* q *are distinct primes, the above property is satisfied by each and every homomorphism. That is, the set of values for* X, Y, X', Y', X'', Y'' *in,* $\phi_s(X) = Y, \phi_{s'}(X') = Y', \phi_{s''}(X'') = Y''$ *is equal to the set of elements of* $(\mathbb{Z}_p \times \mathbb{Z}_p)$. *Here,* s, s', s'' *are elements generating* $\mathbb{Z}_q$ *in each case.*

Premise 3.6. Suppose H has different sets of generating elements. For an example, $H = \langle h_1, h_2, \cdots, h_i \rangle$ or $H = \langle h'_1, h'_2, \cdots, h'_i \rangle$ or $H = \langle h''_1, h''_2, \cdots, h''_i \rangle$ etc. (the number of elements in any arbitrary generating set might be $\ge$ the number of elements in an irredundant generating set for H. That is, the number of elements in each generating set can be different). Let ϕ_s be defined for $a_1^k, a_2^k, \cdots, a_i^k \in \mathbb{Z}$, $1 \le k \le i$ as, $\phi_s(h_1^{x_1} h_2^{x_2} \cdots h_i^{x_i}) = s^{-1}(h_1^{x_1} h_2^{x_2} \cdots h_i^{x_i})s$ $= h_1^{a_1^1 x_1 + a_1^2 x_2 + \cdots + a_1^i x_i}\, h_2^{a_2^1 x_1 + a_2^2 x_2 + \cdots + a_2^i x_i} \cdots h_i^{a_i^1 x_1 + a_i^2 x_2 + \cdots + a_i^i x_i}$

That is, $\phi_s(x_1, x_2, \cdots, x_i)$

$= (a_1^1 x_1 + a_1^2 x_2 + \cdots + a_1^i x_i,\ a_2^1 x_1 + a_2^2 x_2 + \cdots + a_2^i x_i,\ \cdots,\ a_i^1 x_1 + a_i^2 x_2 + \cdots + a_i^i x_i)$

(compare with the definition of T in Example 4.1.).

Considering the generating elements in, $\{h_1, h_2, \cdots, h_i\}$ or $\{h'_1, h'_2, \cdots, h'_i\}$ or $\{h''_1, h''_2, \cdots, h''_i\}$ etc. whichever the elements chosen to be in the generating set for a group $H \rtimes_\phi K$, the entire group can be generated following the above definition of ϕ_s.

And whichever the elements chosen to be in the generating set for the Cayley graph of a group $H \rtimes_\phi K$, a Cayley graph having a similar structure is generated, following the above definition of ϕ_s.

Example 3.7. *An irredundant generating set for* $\mathbb{Z}_p \times \mathbb{Z}_p$ *is* $\{(1,0), (0,1)\}$. *Nevertheless, sets such as* $\{(1,0), (0,3)\}, \{(1,1), (0,2)\}$ *etc. also generate* $\mathbb{Z}_p \times \mathbb{Z}_p$ *(always generating a torus structured Cayley graph), and so it is clear that the subgroup* $\mathbb{Z}_p \times \mathbb{Z}_p$ *of* $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes \mathbb{Z}_q$ *has many generating sets. That is, sets like* $\{((1,0),0), ((0,1),0)\}, \{((1,0),0), ((0,3),0)\}, \{((1,1),0), ((0,2),0)\}$ *etc.. Out of the elements such as* $\{((1,0),0), ((0,1),0)\}, \{((1,0),0), ((0,3),0)\}, \{((1,1),0), ((0,2),0)\}$ *etc. whatever the suitable pair chosen as* $\{h_1, h_2\}$ *for* $\mathbb{Z}_p \times \mathbb{Z}_p$, *under the particular* ϕ , *the vertices get mapped following the same definition* $\phi_s(x_1, x_2) = (-x_2, x_1 - x_2)$ *(see definition of* $T(x_1, x_2) = \phi_s(x_1, x_2)$ *in Example 4.1.). Thus, a Cayley graph having a similar structure will be generated independent of the choice of the generating elements (if the vertex labels are ignored).*

- If a random pair of generating elements was used, for defining ϕ_s as in Premise 3.6. (by Premise 3.6. it is clear that random generating elements can be used for the same ϕ), *as an example, instead of* $\{((1,0),0), ((0,1),0)\}$, *if one of the other pairs of generating elements was chosen such as,* $\{((1,0),0), ((0,3),0)\}, \{((1,1),0), ((0,2),0)\}$ *etc. in* $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_\phi \mathbb{Z}_q$, then, the determination of ϕ_s nor the generating elements chosen, by using only values of the form $\phi_{s^k}(h)$, where $h \in H$ (which will be made public in the proposed cryptographic protocols: e.g. see the sequences of elements made public in steps 1,2 of the protocols in subsections 3.1., 3.2.), is difficult.

The reasons are, the impossibility of determining a homomorphism or an automorphism without the knowledge

on which generating elements have resulted in the corresponding values in the public-key sequence and the presence of many choices for $g_l \in H, l$ and $\theta: K \to \mathrm{Aut}(H)$ such that, $\theta_{s^l}(g_l) = \phi_{s^k}(h)$ (as shown by Premises 3.2. – 3.4.).

- But if the protocols were designed to use a fixed and obvious set of generating elements like $\{((1,0),0),((0,1),0)\}$, and an eavesdropper also gets knowledgeable regarding it, he can check all the possible ϕ's and choose the ϕ which gives the values of the public-keys under the respective generating elements.

  Then, even the presence of a sequence of public-key values will not help to maintain security, since the eavesdropper can attempt on assuming the values in the sequence by choosing a pair of values as corresponding to $\phi_{s^k}(((1,0),0))$ and $\phi_{s^k}(((0,1),0))$ and thereby identify possible set of private-keys.
- Hence, it is proven that, not using an irredundant generating set like $\{((1,0),0),((0,1),0)\}$ but rather using a random generating set, makes it impossible to make any identification using only the public-key values of the form $\phi_{s_j^k}(h)$, where $h \in H$, H and K, even if a third party try to check all the possible values defining each ϕ_{s_j}'s (a third party can check for all the possible homomorphisms corresponding to $H \rtimes_\phi K$, if he knows H and K). And this mathematical hardness is due to the presence of many elements getting mapped to the same value (by Premises 3.2. – 3.4.).
- Moreover, determining whether the communicating party has used exactly ϕ_{s_j}'s or any other $\phi_{s_j^k}$'s for some k, to compute the public-key sequences they make public, becomes a very difficult problem, if the communicating parties make use of an automorphism like $\phi_{s_j^k}$ for random k to compute the public-keys instead of utilizing ϕ_{s_j}'s only. Then the robustness of the protocol is further increased.

Therefore, determining $(h,0)$ *or* $\phi_{s_j^k}$*, by only using values like* $(h',0)$ *such that,* $\phi_{s_j^k}((h,0)) = (h',0)$*, H and K, is a very difficult problem/mathematically intractable, for any* s_j, k, ϕ *and* $(h,0),(h',0) \in H$.

■

Let the communicating parties be Bob and Alice. Suppose Bob communicate using $H \rtimes_\theta K$ whereas Alice uses $H \rtimes_\phi K$. Both parties choose an irredundant generating set for H, say, $H = \langle h_1, \cdots, h_{i_b} \rangle$, $H = \langle h'_1, \cdots, h'_{i_a} \rangle$ respectively.

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

Suppose Bob needs to sign his message M. First he has to compute a hash of the message using a suitable hash function.

Assumption 3.8. *Assume that the messages are from a suitable domain, where they can be mapped to a hash value which is an element of* H*, using a suitable hash function.*

Let $hash(M) = h$ such that $h \in H$.

3.1 Signature protocol 1

1. Bob chooses an element, say (h_H, s_H) and makes it public together with the sequence, $\{\theta_{s_{j_b}}(h_1), \theta_{s_{j_b}}(h_{i_b+1}), \cdots, \theta_{s_{j_b}}(h_{i_b+3}), \cdots, \theta_{s_{j_b}}(h_{i_b})\}_{j_b=1}^{n_b}$, where $\theta_{s_{j_b}}(h_{i_b+1}), \theta_{s_{j_b}}(h_{i_b+2}), \theta_{s_{j_b}}(h_{i_b+3}), \cdots$ are additions of some random elements at random places known only by Bob.
2. Alice sends Bob, $\{\phi_z(\theta_{s_{j_b}}(h_1)), \phi_z(\theta_{s_{j_b}}(h_{i_b+1})), \cdots, \phi_z(\theta_{s_{j_b}}(h_{i_b+3})), \cdots, \phi_z(\theta_{s_{j_b}}(h_{i_b}))\}_{j_b=1}^{n_b}$, for an arbitrary ϕ_z secretly chosen by her.

($z = (s'_1)^{z_1}(s'_2)^{z_2} \cdots (s'_{n_a})^{z_{n_a}}$ for some $z_1, z_2, \cdots, z_{n_a} \in \mathbb{Z}$).

3. Bob first compute, $\{\phi_z(h_1), \cdots, \phi_z(h_{i_b})\}$, by discarding the random additions and utilizing only the mapped values of the generating elements, $\{h_1, \cdots, h_{i_b}\}$.

   Then, $\phi_z(w^{-1}) = Sig$ is computed, where $h \cdot w = h_H$ and he attach it to the message as the signature.
4. Alice multiplies, $\phi_z(h_H) \cdot Sig = \phi_z(h_H) \cdot \phi_z(w^{-1}) = h$, utilizing Bob's public-key.

Together with the signature, Alice receives an encrypted message from Bob. She can decrypt the message, obtain M and apply the same hash function to it. If it is equal to the above h value, then the signature is verified.

In step 3 of the protocol, Bob identify $g_y \in H$ such that $\theta_{s_{j_b}}(g_y) = h_y$, for all j_b, y in order to obtain $\phi_z(\theta_{s_{j_b}}(g_y)) = \phi_z(h_y)$; $1 \le y \le i_b$.

The Lemma 3.1., shows that the public-key sequence of Bob doesn't compromise security. By Premises 3.2. – 3.6., it is proven that an eavesdropper can not identify the generating elements nor the θ_{s_j}'s, even if he attempts to check all possible θ's and generating elements. There exists

a $\phi: K \to Aut(H)$, such that, $\phi_{s^l}(f_l) = h_y$, for any $\theta_s(g) = h_y$, making it a mathematically intractable problem to determine θ. This is satisfied by all the possible homomorphisms, ϕ's, for each automorphism represented by each value of l. Furthermore, for any $\theta_s(g) = h_y$, there exists $\theta_{s^k}(f_k) = h_y$, for all k, which creates further hardness in determining which generating elements, and θ might have been chosen by Bob.

In obtaining the public-key sequence, Bob can use $\theta_{s^k_{j_b}}$'s for some random k, instead of $\theta_{s_{j_b}}$'s to make the protocol more secure. However, the above protocol is practical only if it is not possible to distinguish between the extra additions $\theta_{s_{j_b}}(h_{i_b+1}), \theta_{s_{j_b}}(h_{i_b+2}), \theta_{s_{j_b}}(h_{i_b+3}), \cdots$ and the values obtained from the generating elements, i.e. $\theta_{s_{j_b}}(h_1), \cdots, \theta_{s_{j_b}}(h_{i_b})$.

The reason is that, since $h_1, \cdots, h_{i_b}$ are generating elements, $\theta_{s_{j_b}}(h_1), \cdots, \theta_{s_{j_b}}(h_{i_b})$ are also generating elements of H. Then, once an eavesdropper finds $\{\phi_z(\theta_{s_{j_b}}(h_1)), \cdots, \phi_z(\theta_{s_{j_b}}(h_{i_b}))\}$, he can use it to determine ϕ_z. Furthermore, if an eavesdropper attempt to guess values corresponding to generating elements, by thinking of all possible generating sets and perform different trials to identify a correct choice, the protocol is still vulnerable even if additional random elements are added to the sequence. The following suggestion proposes a method which can overcome this restriction as well.

3.2 Signature protocol 2

1. Bob selects a public element, say (h_H, s_H). Suppose $h \cdot w = h_H$ and, $w = h_1^{x_1} h_2^{x_2} \cdots h_{i_b}^{x_{i_b}} = (h_1^{A_1} h_2^{B_1} \cdots h_{i_b}^{I_1}) \cdot (h_1^{A_2} h_2^{B_2} \cdot \cdots h_{i_b}^{I_2}) \cdots (h_1^{A_L} h_2^{B_L} \cdots h_{i_b}^{I_L})$, for some $A_j, B_j, \cdots, I_j \in \mathbb{N}$, $1 \le j \le L$.
   Let $h_1^{A_1} h_2^{B_1} \cdots h_{i_b}^{I_1} = elt_1^{r_1}$, $h_1^{A_2} h_2^{B_2} \cdots h_{i_b}^{I_2} = elt_2^{r_2}$, $\cdots$, $h_1^{A_L} h_2^{B_L} \cdots h_{i_b}^{I_L} = elt_L^{r_L}$.
2. He makes (h_H, s_H) and, $\{elt_1, elt_2, elt_{L+1}, elt_3, \cdots, elt_6, elt_{L+3}, \cdots, elt_L\}$ public, where $elt_{L+1}, elt_{L+2}, elt_{L+3}, \cdots$ are extra additions at random positions, to the sequence $\{elt_1, \cdots, elt_L\}$, which includes the elements whose product contribute in forming w.
3. Alice sends Bob, $\{\phi_z(elt_1), \phi_z(elt_2), \phi_z(elt_{L+1}), \phi_z(elt_3), \cdots, \phi_z(elt_6), \phi_z(elt_{L+3}), \cdots, \phi_z(elt_L)\}$ for an arbitrary ϕ_z keeping the same order of elements as Bob has given.
   ($z = (s'_1)^{z_1}(s'_2)^{z_2} \cdots (s'_{n_a})^{z_{n_a}}$ for some $z_1, z_2, \cdots z_{n_a} \in \mathbb{Z}$).
4. Bob first compute, $\phi_z(w)$.

$$\underbrace{\phi_z(elt_1) \cdot \phi_z(elt_1) \cdots \phi_z(elt_1)}_{r_1-times} \cdots \underbrace{\cdot \phi_z(elt_L) \cdot \phi_z(elt_L) \cdots \phi_z(elt_L)}_{r_L-times}$$
$$= \phi_z(elt_1^{r_1}) \cdots \phi_z(elt_L^{r_L}) = \phi_z(elt_1^{r_1} \cdots elt_L^{r_L}) = \phi_z(w)$$

5. Then, $\phi_z(w^{-1}) = Sig$ is computed, and he attach it to the message as the signature.
6. Alice multiplies, $\phi_z(h_H) \cdot Sig = \phi_z(h_H) \cdot \phi_z(w^{-1}) = h$, utilizing Bob's public-key.

Since a signature will be sent along with an encrypted message, Alice will have the decrypted message M. She can obtain the hash of the decrypted message using the same hash function and compare with the above h value. If they are equal, the signature is verified.

In the sequence, $\{elt_1, \cdots, elt_L\}$, some elements elt_j might represent elements of the set $\{h_1, \cdots, h_{i_b}\}$. It is necessary to choose elt_j's such that $\{elt_j\}_{j=1}^{L}$ will not include all the elements in $\{h_1, \cdots, h_{i_b}\}$ because if it does, since they are generating elements, $\phi_z(elt_j)$'s will also be generating H, this will facilitate an eavesdropper to identify the generating elements from the public sequence and their mappings $\phi_z(elt_j)$'s under ϕ_z. Then, the eavesdropper can determine ϕ_z (recreating the weakness mentioned under protocol 1 again).

In step 4, Bob can choose to compute $\phi_z(w^{-1})$ directly by choosing $elt_j^{r_j}$'s to be elements whose products will form w^{-1}.

Using $\phi_z(elt_j^{r_j})$ values to compute w, while making only elt_j's public gives additional security since the r_j's are known only by Bob. Moreover, the presence of additional elements such as $elt_{L+1}, elt_{L+2}, elt_{L+3}, \cdots$ in the public-key sequence assist in hiding which values will be used by Bob in his computation. By Lemma 3.1., it is not possible to determine ϕ_z, using only elt_j's and $\phi_z(elt_j)$'s, even if an eavesdropper tries to check all possible homomorphisms and generating sets, since the generating sets are chosen secretly by Alice and Bob, unless the platform group allows efficient and faithful matrix representations and are vulnerable to cryptanalyses such as the algebraic span cryptanalysis.

3.3 Security Analysis and Discussion

This section includes further analysis on the security aspects and discussions in addition to the explanations given above under each protocol suggestions.

The automorphisms $\phi_Z, \theta_{Z'}$, where $Z, Z' \in K$ are applicable on elements of H. Thus, we have chosen to utilize the x-coordinates in illustrating the signature protocols.

Consider the following attacks commonly applicable for digital signature schemes.

a) Key-only attack: the eavesdropper has only the public-key of message sender.
b) Known message attack: the eavesdropper possesses a set of valid signatures for some known messages, which were not chosen by him.
c) Adaptive chosen message attack: the eavesdropper has the ability to obtain the signatures for arbitrary messages chosen by him.

It is evident that the proposed signature scheme offers thorough security against the Key-only attack (if the values transmitted by Alice after application of ϕ_z are not taken in to account when considering the information known by the eavesdropper, since, then only the public-keys of the sender of the message are known to any third party). To prevent impacts from the Known message attack and the Adaptive chosen message attack, we suggest that for each communication between two parties, groups and keys should be generated each time and discarded after the communication as one-time keys. This also eliminates the insecurity created by the fact that Alice also can generate Bob's signatures using ϕ_z and h_H.

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 even though they were used in examples to explain the ideas clearly, because they offer efficient representations as matrix groups. Since ϕ_z is an action by conjugation, when matrix representations are feasible, an eavesdropper can form a system of linear equations using $z^{-1}elt_j z = f_j$ (or $z^{-1}h_y z = f_y$) which can be solved since elt_j's and f_j's (or h_y's and f_y's) are known (by algebraic span cryptanalysis).

We will also note another weakness with respect to protocol 1 as it will facilitate analysis when checking for suitable platform groups in future studies. These groups are especially not suitable when considering protocol 1, since all the elements except the identity in $\mathbb{Z}_p \times \mathbb{Z}_p$ can be used as generating elements. Then, in $\{\theta_{s_{j_b}}(h_1), \theta_{s_{j_b}}(h_{i_b+1}), \cdots, \theta_{s_{j_b}}(h_{i_b+3}), \cdots, \theta_{s_{j_b}}(h_{i_b})\}$, an eavesdropper can attempt to arbitrarily choose a suitable set as a generating set and use it to determine ϕ_z.

4. Using Hamiltonian paths or any random path in Cayley graphs

The above schemes 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. Although 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 expectation that it will be useful for any scholar where the utilization of paths and cycles has advantages and an inception for further research.

In Cayley graphs which are proven to have Hamiltonian cycles or paths, usually the cycles or paths can be written using mathematical formulae.

Example 4.1. *Consider* $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_\phi \mathbb{Z}_3$. *A set* $\{s, h_1\}$, *with* $|s| = 3, |h_1| = p$ *is an irredundant generating set for the group, where the action of* s *on* $\mathbb{Z}_p \times \mathbb{Z}_p$ *can be defined as below* [2].

Let a linear transformation T *on* $\mathbb{Z}_p \times \mathbb{Z}_p$ *be defined by* $T(h) = s^{-1}hs$. *Let* $m(x)$ *be the minimal polynomial of* T *and* $h_2 = T(h_1) = s^{-1}h_1 s$. *As,* $|s| = 3, T^3 = I$, *so* $m(x)$ *divides* $x^3 - 1 = (x-1)(x^2 + x + 1)$ *(Refer* [2], *for a complete argument on determining that* $m(x) = x^2 + x + 1$*).* T *can be defined with respect to a basis* $\{h_1, h_2\}$ *of* $\mathbb{Z}_p \times \mathbb{Z}_p$ *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, since* $\mathrm{Aut}(\mathbb{Z}_p \times \mathbb{Z}_p) \cong GL_2(\mathbb{Z}_p)$, *where* $GL_2(\mathbb{Z}_p)$ *is the general linear group of* 2×2 *matrices over* $\mathbb{Z}_p$. *Identify by comparing the coefficients* a_y^k*'s in Premise 3.6. in the proof of Lemma 3.1., with the coefficients of* x_1 *and* x_2 *in* $T(x_1, x_2) = (-x_2, x_1 - x_2)$ *(that is,* $\phi_s(x_1, x_2) = (-x_2, x_1 - x_2)$*). In particular,* $a_1^1 = 0, a_1^2 = -1, a_2^1 = 1, a_2^2 = -1$. *A Hamiltonian cycle in the corresponding Cayley graph of* $(\mathbb{Z}_p \times \mathbb{Z}_p) \rtimes_\phi \mathbb{Z}_3$ *is* [2],

$$([({h_1}^{-1})^{3j-1}, s, (h_1)^{-3j-1}, s^{-1}]_{j=1}^{\frac{(p-1)}{2}}, [(h_1)^{\frac{(p-5)}{2}}, s^{-1}, {h_1}^{\frac{(p+1)}{2}}, s], [{h_1}^{p-1}, s^{-1}], [{h_1}^{3j-1}, s, ({h_1}^{-1})^{-3j-1}, s^*]_{j=\frac{-(p-1)}{2}}^{-k-1}, [({h_1}^{-1})^{3j-1}, s^{-1}, {h_1}^{-3j-1}, s^{-1}]_{j=-k}^{-2}, [({h_1}^{-1})^{p-4}, s^{-1}, t^2, s])$$

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 the Hamiltonian cycle can be written using ϕ, p, q, s and h_1 via the mathematical proof, without generating the graph. Also, any random cycle or path of any length can be mathematically written once ϕ, p, q, s and h_1 are known. Therefore, for the cryptographic protocols, we are proposing to obtain the Hamiltonian paths (or any random path) using the mathematical proofs without generating the Cayley graphs.

The difficulty of determining the Hamiltonian paths or cycles (and any random 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 4.2. 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 $\phi_{s_j^k}$'s). The vertex obtained as the

ending vertex of a Hamiltonian path is different for different choices of the generating elements.

Therefore, by arguments in Lemma 3.1. and Premise 4.2., it is proven that making the ending vertex public does not compromise the security of the protocol and reveal the Hamiltonian path chosen, if θ and the generating elements are kept as secrets. If any random path was used in the protocol, the same argument follows. In fact, the determination of a random path is even harder, due to the presence of many random paths in a graph ending at the same vertex (more than Hamiltonian paths).

Let $X_a = Cay(H \rtimes_\phi K, S_a)$ and $X_b = Cay(H \rtimes_\theta K, S_b)$ be the Cayley graphs of Alice and Bob, with respect to the generating sets S_a, S_b defined above. As the public element Bob chooses the ending vertex of a Hamiltonian path or any random path in the Cayley graph, (h_H, s_H) (starting at the identity vertex) and the remaining steps of the protocols; both under protocol 1 and protocol 2, can be followed similarly (if the path doesn't start at the identity vertex, the product of the generating elements representing the path will not be equal to (h_H, s_H) and it will have to be taken in to account when performing computations).

The generation of Hamiltonian paths and cycles using mathematical proofs (as in Example 4.1.) is possible once H, K and ϕ are known, instead of generating the graph. This is an advantage, for both efficiency and security of the protocol (if paths in graphs are to be used). However, a naïve attack is possible, which is, checking all possible homomorphisms (ϕ's) for known H, K, and attempting to identify the ending vertices of all possible Hamiltonian paths in Cayley graphs. The ending vertices will be unique and identifiable if there is to be only one generating set, rather than a randomly chosen one. Thus, the random choice of the generating sets and keeping the generating elements and automorphisms as secrets makes the cryptographic scheme secure against this attack.

5. Applications to Generalized DLP

In this section, we propose operation of variants of the Generalized Diffie-Hellman key exchange and the El-Gamal encryption schemes introduced in [21], using Hamiltonian paths (or any random path) in Cayley graphs of finite groups.

5.1 Discrete Logarithm Problem and Generalized Discrete Logarithm Problem

The traditional DLP has been the foundation of many practical public-key cryptosystems used up to date. For a finite cyclic group G, generated by an element α, the DLP is to find a non-negative integer x, such that $\alpha^x = \beta$, where $\beta \in G$.

The authors of [22], had generalized the concept of DLP to a problem over non-abelian groups. They had named it as the GDLP, which can be identified as one of the numerous attempts to extend the notion of the traditional DLP to a problem over non-abelian groups.

Definition 5.1. [22] *Let G be a finite group generated by* $\alpha_1, \cdots, \alpha_t$. *i.e.* $G = \langle \alpha_1, \cdots, \alpha_t \rangle$. *Denote by* $\alpha = (\alpha_1, \cdots, \alpha_t)$, *the ordered tuple of generators of the group* G. *For a given* $\beta \in G$, *the GDLP of* β *with respect to* α *is to determine a positive integer* k *and a* (kt) *- tuple of non-negative integers* $x = (x_{11}, \cdots, x_{1t}, \cdots, x_{k1}, \cdots, x_{kt})$ *such that,* $\beta = \prod_{i=1}^{k} (\alpha_1^{x_{i1}} \cdots \alpha_t^{x_{it}})$.

This can be expressed using the notation $\beta = \alpha^x$. The (kt)-tuples $(x_{11}, \cdots, x_{1t}, \cdots, x_{k1}, \cdots, x_{kt})$ are known as the *generalized discrete logarithms* of β with respect to $\alpha = (\alpha_1, \cdots, \alpha_t)$. In 2010, I. Ilić [21] (see [23] for a detailed description) had conducted further studies on this GDLP, and had proposed generalized versions of the Diffie-Hellman key exchange and the El-Gamal encryption schemes based on it. In order to overcome the inequality, $(\alpha^x)^y \neq (\alpha^y)^x$ associated with the non-abelian elements, the operation of conjugation by elements was introduced to commute with the exponentiation by integers.

Theorem 5.2. [23] *Let* $G = \langle \alpha_1, \cdots, \alpha_n \rangle$ *be a finite non-abelian group. Let* $(\alpha_1, \cdots, \alpha_n)^x$ *denote the operation of exponentiation by integer* x *and for* $g \in G$ *let* $(\alpha_1, \cdots, \alpha_n)^g$ *denote the operation of conjugation:* $(\alpha_1, \cdots, \alpha_n)^g = (\alpha_1^g, \cdots, \alpha_n^g) = (g^{-1}\alpha_1 g, \cdots, g^{-1}\alpha_n g)$. *Then,* $((\alpha_1, \cdots, \alpha_n)^x)^g = ((\alpha_1, \cdots, \alpha_n)^g)^x$.

Let $(h, k), (h', k') \in H \rtimes_\phi K$. Let us consider the conjugate of an element in this group: $(h, k)^{-1}(h', k')(h, k) = (\phi_k^{-1}(h^{-1}), k^{-1})(h', k')(h, k) = (\phi_k^{-1}(h^{-1}) \cdot \phi_{k^{-1}}(h'), k^{-1} \cdot k')(h, k) = (\phi_k^{-1}(h^{-1}) \cdot \phi_{k^{-1}}(h') \cdot \phi_{k^{-1}.k'}(h), k^{-1} \cdot k' \cdot k) = (\phi_k^{-1}(h^{-1}) \cdot \phi_{k^{-1}}(h') \cdot \phi_{k^{-1}.k'}(h), k')$.

If the platform group used is a group which doesn't have an efficient and a faithful representation as a matrix group (algebraic span method is easily applicable if it has matrix representations) , then following the explanations in Lemma 3.1. it is clear, that conjugacy search problem is a difficult problem in most of these semidirect products, even if (h', k') is known, since there are many triplets of values possible to be occupied by $\phi_k^{-1}(h^{-1}), \phi_{k^{-1}}(h')$ and $\phi_{k^{-1}.k'}(h)$, which can simplify to give the same result, making it impossible to determine exact h and k values.

5.2 A Generalized Diffie-Hellman key exchange protocol using paths in Cayley graphs

Let Alice and Bob be communicating using $G = H \rtimes_\phi K$ (the same group instead of using different groups). With respect to a Hamiltonian path (starting at the identity vertex), or any random path in a Cayley graph of G, let the secret value x denote a tuple of elements representing the exponents of each generating element along the path.

1. Let $X_a = Cay(H \rtimes_\phi K, S_a)$, where $S_a = \{s, t\}$ chosen by Alice.

   Let $\alpha = \left(\underbrace{s, t, s, t, \cdots, s, t}_{n-terms}\right)$ be the sequence of elements falling along the Hamilton path (or any random path), $s^{x_1}, t^{x_2}, s^{x_3}, t^{x_4}, \cdots, s^{x_{n-1}}, t^{x_n}$ be the Hamiltonian path (or any random path) represented in terms of the generating elements of the Cayley graph and, $\beta = s^{x_1}t^{x_2}s^{x_3}t^{x_4}\cdots s^{x_{n-1}}t^{x_n}$. Then, $x = (x_1, x_2, x_3, x_4, \cdots, x_n)$.

   Alice's secret keys are $\{s, t\}$ and x. Public-keys are, $(\{g_1, g_2, \cdots, g_j, s, t, g_{j+1}, \cdots, g_i\}, \beta)$ for some $g_1, g_2, \cdots, g_i \in H \rtimes_\phi K$, placed at arbitrary positions in the sequence $\{g_1, g_2, \cdots, g_j, s, t, g_{j+1}, \cdots, g_i\}$ (positions of the generating elements s, t are to be known only by Alice and the same arguments follow for $|S_a| > 2$).

2. Bob chooses a secret key, $g = (h, k) \in H \rtimes_\phi K$, and compute, $\beta' = (g^{-1}g_1g, g^{-1}g_2g, \cdots, g^{-1}g_jg, g^{-1}sg, g^{-1}tg, g^{-1}g_{j+1}g, \cdots, g^{-1}g_ig)$, keeping the same order of elements as in Alice's public-key sequence. The public-key of Bob is β'.

3. Alice identify $g^{-1}sg, g^{-1}tg$, by their positions in β' (which is known by her only) and compute,

$$(\beta')^x = (g^{-1}sg)^{x_1}(g^{-1}tg)^{x_2}(g^{-1}sg)^{x_3}(g^{-1}tg)^{x_4} \cdots (g^{-1}sg)^{x_n-1}(g^{-1}tg)^{x_n}$$
$$= (g^{-1}s^{x_1}g)(g^{-1}t^{x_2}g)(g^{-1}s^{x_3}g)(g^{-1}t^{x_4}g)\cdots (g^{-1}s^{x_n-1}g)(g^{-1}t^{x_n}g)$$
$$= g^{-1}s^{x_1}t^{x_2}s^{x_3}t^{x_4}\cdots s^{x_n-1}t^{x_n}g,$$

   and Bob computes, $\beta^g = g^{-1}(s^{x_1}t^{x_2}s^{x_3}t^{x_4}\cdots s^{x_n-1}t^{x_n})g$, establishing a shared secret key between the two communicating parties.

5.3 Security Analysis and Discussions

- Based on the choice for S_a, the order of vertices visited and the ending vertex of the Hamiltonian path (or any random path) changes. This enhances the security of the scheme.

  It is mentioned to place the elements $g_1, g_2, \cdots, g_i$ at arbitrary positions in the sequence $\{g_1, g_2, \cdots, g_j, s, t, g_{j+1}, \cdots, g_i\}$, in order to hide which elements were chosen as $\{s, t\}$. Even the placement of s, t can be consecutive or apart from each other.
- As x, if Alice chose a sequence of exponents corresponding to a random path rather than a Hamiltonian path, β will represent the ending vertex of the particular path, and it amplifies the security of the protocol through creation of further ambiguity to an eavesdropper in tracing the secret key x.
- The proposed scheme is more secure if there are several Hamiltonian paths which end at the same vertex (since this gives several possibilities for x and uncertainty in determining $\{s, t\}$ and x based on the ending vertex if a Hamiltonian path is to be used).

  In fact, in the proposed protocol, if the ending vertex of the Hamiltonian path is to be chosen as β, it is necessary to assume that,

Assumption 5.3. *There exists several Hamiltonian paths ending at the same vertex either for a fixed choice of S_a or randomized different choices of S_a or both cases.*

5.4 A Generalized El-Gamal encryption scheme using paths in Cayley graphs

1. *Key generation*: Let $H \rtimes_\phi K$ be the non-abelian group over which the communication processes. Let the Cayley graph X_a, the choice of the generating set S_a, α and all public and private-keys of Alice and Bob be same as in the preceding protocol (the same arguments follow for $|S_a| > 2$).
2. *Encryption*: To send a message $m \in H \rtimes_\phi K$ to Alice, Bob computes, $C = m\beta^g$ and send the encrypted message C to her.
3. *Decryption*: To decrypt the message from Bob, Alice uses her secret key x to compute,

$$(\beta')^x = (g^{-1}sg)^{x_1}(g^{-1}tg)^{x_2}(g^{-1}sg)^{x_3}(g^{-1}tg)^{x_4} \cdots (g^{-1}sg)^{x_n-1}(g^{-1}tg)^{x_n}$$
$$= g^{-1}s^{x_1}t^{x_2}s^{x_3}t^{x_4}\cdots s^{x_n-1}t^{x_n}g = (\alpha^g)^x,$$

   and $((\beta')^x)^{-1}$, and multiplies on the right by C she received from Bob.

$$C \cdot (\beta')^{-x} = m\beta^g(\alpha^g)^{-x} = m(\alpha^x)^g(\alpha^g)^{-x} = m.$$

The discussions on security for the previous scheme also apply to this protocol as well. Alice's secret key can be chosen to represent a Hamiltonian path or any random path. The schemes are secure only if s, t are indistinguishable from the other elements of the public sequence. Both the protocols can be evaluated by choosing any element of the group directly, even though use of paths from a Cayley graph is illustrated due to the academic interest and novelty.

6. Resistance to Algebraic span cryptanalysis

In [19], B. Tsaban et al. introduced a method to obtain provable polynomial time solutions of problems in non-abelian group based Cryptography. Known as the "algebraic span cryptanalysis", this technique had evolved founded upon previously proposed methods such as Cheon-Jun method [24] and the Tsaban's linear centralizer method [25]. Tsaban's span method is applicable to problems where a system of linear equations can be obtained from matrix pairs g and $f = g^z$, when g and f are known (notation: $g^z = z^{-1}gz$ for a matrix z). Thus, it can be easily used on

non-abelian groups that have efficient and faithful representations as matrix groups and is a very influential attack.

By integrating a notion of marginal sets, V. Roman'kov [20] has suggested the creation of a system with the matrix pairs g and $f = (cg)^z$, where a linear system of equations can't be obtained, hence making it resistant against this attack ($(cg)^z = z^{-1}(cg)z$). The use of a "salt" γ, in the system $X^{-1}AX = \gamma B$, to create resistance against conjugacy search in [26] also quite resembles this approach in the way of using a product of elements $cg, \gamma B$ to hide the original g and B.

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

Definition 6.1. [20] *For $n \in \mathbb{N}$, let $w = w(x_1, \cdots, x_n)$ be a group word, G be a group and $\overline{g} = (g_1, \cdots, g_n)$ be a tuple of elements of G. A tuple $\overline{c} = (c_1, \cdots, c_n) \in G^n$ is said to be a "marginal tuple" determined by w and $\overline{g}$ if, $w(c_1 g_1, \cdots, c_n g_n) = w(g_1, \cdots, g_n)$. This is denoted by $\overline{c} \perp w(\overline{g})$.*

Likewise, a set $\overline{C} \subseteq G^n$ is said to be marginal with respect to w and $\overline{g}$, denoted by $\overline{C} \perp w(\overline{g})$, if $\overline{c} \perp w(\overline{g})$ for every $\overline{c} \in \overline{C}$. The author also indicates a universal method of obtaining a marginal set $\overline{C}$ by a word w in [20]:

Let $w = w(a_1, \cdots, a_k) = a_1 a_2 \cdots a_k$, $a_i \in G$, $1 \leq i \leq k$, be any expression in straight form of a fixed element $f \in G$. The values, $a_i = a_j$ or $a_i = a_j^{-1}$ for $i \neq j$ are viable and this word is non-reducible.

Consider $x_1 a_1 x_2 a_2 \cdots x_k a_k = f \quad \rightarrow (1)$

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

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

Also a solution of (1) *can be generated by a sequence of the following random elementary insertions. Suppose we have a solution, $(c_1, \cdots, c_k)$ of* (1)*. For any i and any random element $d \in G$, we can change c_i to $c_i' = c_i a_i d a_i^{-1}$ and c_{i+1} to $c_{i+1}' = d c_{i+1}$. Then, a new solution of* (1) *is obtained. Continuing this process with random d and i, a series of new solutions of* (1) *can be computed.*

Since $w(c_1 a_1, \cdots, c_k a_k) = w(a_1, \cdots, a_k)$, $(c_1 a_1, \cdots, c_k a_k)$ could be used instead of $(a_1, \cdots, a_k)$ in an algorithm to make it secure, while not having the slightest change to the results of computations.

We have made all the above proposals independently before reading and understanding the literature [19], [20]. But if the platform group utilized for the implementation of the protocols has an efficient matrix representation they are vulnerable to the algebraic span cryptanalysis. Hence, we suggest the following modified steps integrating the proposal in [20] to overcome this attack. Let all the notations and variables be same as in the initial proposals in sections 3 and 5.

6.1 Modified signature protocol 1

1. To use the method under protocol 1, Bob first write w in terms of $h_1, \cdots, h_{i_b}$.

   $w = h_1^{x_1} \cdots h_{i_b}^{x_{i_b}} = \underbrace{h_1 \cdot h_1 \cdots h_1}_{x_1 - times} \cdots \underbrace{h_{i_b} \cdot h_{i_b} \cdots h_{i_b}}_{x_{i_b} - times}$

   and determines all marginal tuples, $\overline{c} = (c_1, c_2, \cdots, c_{x_1}, \cdots, c_{x-(x_{i_b}-1)}, c_{x-(x_{i_b}-2)}, \cdots, c_x)$

   such that, $w = c_1 \cdot h_1 \cdot \ c_2 \cdot h_1 \ \cdots c_{x_1} \cdot h_1 \cdots$

   $c_{x-(x_{i_b}-1)} \cdot h_{i_b} \ \cdot c_{x-(x_{i_b}-2)} \cdot h_{i_b} \ \cdots \ c_x \cdot h_{i_b}$
2. Then, he obtain a sequence, $\{h_1, h_1, h_{i_b+1}, \cdots, h_{i_b+3}, \cdots, h_{i_b}\}$, where $h_{i_b+1}, h_{i_b+2}, h_{i_b+3}, \cdots$ are extra additions at random positions known only by him. Bob also modifies the set $\overline{D}$ consisting of the marginal tuples, so that each marginal tuple will have extra elements added at the same positions.

That is, $\overline{c}' = (c_1, c_2, c_{x+1}, \cdots, c_{x+3}, \cdots, c_x)$ for all $\overline{c}' \in \overline{D}'$, where $\overline{D}'$ is the modified marginal set and $c_{x+1}, c_{x+2}, c_{x+3}, \cdots$ are extra elements added.

3. Bob make (h_H, s_H), the sequence $\{h_1, h_1, h_{i_b+1}, \cdots, h_{i_b+3}, \cdots, h_{i_b}\}$, and $\overline{D}'$, public.
4. Alice chooses a random $\overline{c}' \in \overline{D}'$, a secretly chosen arbitrary ϕ_z, compute, $\{\phi_z(c_1 \cdot h_1), \phi_z(c_2 \cdot h_1), \phi_z(c_{x+1} \cdot h_{i_b+1}), \cdots, \phi_z(c_{x+3} \cdot h_{i_b+3}), \cdots, \phi_z(c_x \cdot h_{i_b})\}$ and send the sequence to Bob in the same order.
5. Bob compute, $\phi_z(c_1 \cdot h_1) \cdot \phi_z(c_2 \cdot h_1) \cdots \phi_z(c_x \cdot h_{i_b}) = \phi_z(c_1 \cdot h_1 \cdot c_2 \cdot h_1 \cdots c_x \cdot h_{i_b}) = \phi_z(w)$, using the correct (hidden) values from the sequence.

   Then, $\phi_z(w^{-1}) = Sig$ is computed and he attaches it to the message as the signature.
6. Alice multiplies, $\phi_z(h_H) \cdot Sig = h$, and compare with the hash value obtained from the decrypted message same as in the original proposal.

If c_y's obtained are equal for each h_y, for particular y's Bob may include h_y and c_y only once in the public sequences. Once he acquire the mapping under ϕ_z, he can take it's product with itself, $x_y -$ times to obtain the relevant value ($1 \leq y \leq i_b$).

Example 6.2. *Suppose a marginal tuple is* $\overline{c} = (\underbrace{c_1, c_1, \cdots, c_1}_{x_1 - times},$

$\cdots, \underbrace{c_{x_{i_b}}, c_{x_{i_b}}, \cdots, c_{x_{i_b}}}_{x_{i_b}-times}$). *Then, Bob can make* $\{h_1, h_2, h_{i_b+1}, \cdots, h_{i_b+3}, \cdots, h_{i_b}\}$, *public together with the modified marginal tuples,* $\overline{c}' = (c_1, c_2, c_{x_{i_b}+1}, \cdots, c_{x_{i_b}+3}, \cdots, c_{x_{i_b}}) \in \overline{D}'$, *where* $h_{i_b+1}, h_{i_b+2}, h_{i_b+3}, \cdots$ *and* $c_{x_{i_b}+1}, c_{x_{i_b}+2}, c_{x_{i_b}+3}, \cdots$ *are extra elements added. The equal elements are not repeatedly mentioned in the sequence nor the modified marginal tuples. After Alice sends,* $\{\phi_z(c_1 \cdot h_1), \phi_z(c_2 \cdot h_2), \phi_z(c_{x_{i_b}+1} \cdot h_{i_b+1}), \cdots, \phi_z(c_{x_{i_b}+3} \cdot h_{i_b+3}), \cdots, \phi_z(c_{x_{i_b}} \cdot h_{i_b})\}$, *Bob can identify* $\phi_z(c_y \cdot h_y)$*'s and obtain the products of each element with itself* $x_y -$ *times to compute any* $\phi_z(\underbrace{(c_y \cdot h_y) \cdot (c_y \cdot h_y) \cdots (c_y \cdot h_y)}_{x_y-times})$.

6.2 Modified signature protocol 2

1. Recall that, $w = elt_1^{r_1} \cdot elt_2^{r_2} \cdots elt_L^{r_L}$. Bob compute marginal tuples $\overline{c} \in \overline{D}$ such that,
$$w = c_1 \cdot elt_1 \cdot c_2 \cdot elt_1 \cdots c_{L'} \cdot elt_L.$$
2. He modifies the sequence of public elements as well as each marginal tuple in the marginal set by including extra elements, at the same positions and make them public together with (h_H, s_H).

   The modified public sequence would be like, $\{elt_1, elt_1, elt_{L+1}, \cdots, elt_{L+3}, \cdots, elt_L\}$, where elt_{L+1}, $elt_{L+2}, elt_{L+3}, \cdots$ are extra elements and a modified tuple $\overline{c}'$ would be, $\overline{c}' = (c_1, c_2, c_{L'+1}, \cdots, c_{L'+3}, \cdots, c_{L'}) \in \overline{D}'$, where $c_{L'+1}, c_{L'+2}, c_{L'+3}, \cdots$ are extra additions.
3. Alice chooses a tuple $\overline{c}' \in \overline{D}'$ and shares the sequence,
$$\{\phi_z(c_1 \cdot elt_1), \phi_z(c_2 \cdot elt_1), \phi_z(c_{L'+1} \cdot elt_{L+1}), \cdots, \phi_z(c_{L'+3} \cdot elt_{L+3}), \cdots, \phi_z(c_{L'} \cdot elt_L)\}$$
4. Bob identify the necessary elements by their positions in the sequence and compute,
$$\phi_z(c_1 \cdot elt_1) \cdot \phi_z(c_2 \cdot elt_1) \cdots \phi_z(c_{L'} \cdot elt_L) = \phi_z(c_1 \cdot elt_1 \cdot c_2 \cdot elt_1 \cdots c_{L'} \cdot elt_L = \phi_z(w).$$

Then he obtains, $\phi_z(w^{-1}) = Sig$ and attach it to the message, which can be verified by Alice following the same steps mentioned before.

As discussed under the protocol 1's modification above, if c_j's obtained corresponding to elt_j for particular j's are equal, Bob may include elt_j and c_j only once in the public sequences. Once he acquire the mapping under ϕ_z, he can obtain it's product with itself, $r_j -$ times to obtain the relevant value ($1 \leq j \leq L$).

6.3 Modified Generalized Diffie-Hellman protocol

1. Alice compute a marginal set $\overline{C} \perp \beta(\overline{g})$, where $\overline{g}$ is the tuple $\overline{g} = (s, t)$. Then her public-keys are $(\{g_1, g_2, \cdots, g_j, s, t, g_{j+1}, \cdots, g_i\}, \beta, \overline{C}')$ and the private-keys are same as before, where $\overline{C}'$ is the modified marginal set with additional elements $c_1, \cdots, c_i$ added to each tuple $\overline{c} \in \overline{C}$ at same positions occupied by the additions $\{g_1, \cdots, g_i\}$.
2. Bob chooses a random marginal tuple $\overline{c} = (c_1, \cdots, c_j, c', c'', c_{j+1}, \cdots, c_i) \in \overline{C}$ and compute $\beta' = (g^{-1}c_1g_1\,g,\ g^{-1}c_2g_2\,g,\ \cdots, g^{-1}c_jg_j\,g,\ g^{-1}c's\ g,\ g^{-1}c''tg,\ g^{-1}c_{j+1}g_{j+1}g,\ \cdots,\ g^{-1}c_ig_i\,g)$,keeping the same order of elements as in Alice's public-key sequence. The public-key of Bob is now β'.
3. Alice identify $g^{-1}c'sg, g^{-1}c''tg$, by their positions in β' (which is known by her only) and compute,
$$(\beta')^x = (g^{-1}c'sg)^{x_1}(g^{-1}c''tg)^{x_2}(g^{-1}c'sg)^{x_3} (g^{-1}c''tg)^{x_4} \cdots (g^{-1}c'sg)^{x_{n-1}}(g^{-1}c''tg)^{x_n}$$
$$= (g^{-1}(c's)^{x_1}g)(g^{-1}(c''t)^{x_2}g)(g^{-1}(c's)^{x_3}g) (g^{-1}(c''t)^{x_4}g) \cdots (g^{-1}(c's)^{x_{n-1}}g)(g^{-1}(c''t)^{x_n}g)$$
$$= g^{-1}(c's)^{x_1}(c''t)^{x_2}(c's)^{x_3}(c''t)^{x_4} \cdots (c's)^{x_{n-1}} (c''t)^{x_n}g,$$ and Bob computes,$\beta^g = g^{-1}(s^{x_1}t^{x_2}s^{x_3} t^{x_4} \cdots s^{x_{n-1}}t^{x_n})g$, establishing a shared secret key.

This is applicable, if the component of the marginal tuple multiplying each s in the element β are equal and that multiplying each t are also equal. That is, equal to c' and c'' respectively. Otherwise, suppose, $\beta = ststst \cdots t$. Then, marginal tuples $(c_1, \cdots, c_{y_n})$ can be computed such that, $\beta = (c_1s)(c_2t)(c_3s)(c_4s)(c_5t) \cdots (c_{y_n}t)$. Let $\overline{C}$ be the set with all such marginal tuples, and $\overline{C}'$ be the set including modified marginal tuples $\overline{c}' = (c_1, d_1, c_2, \cdots, c_{y_n})$ with additional elements $d_1, d_2, \cdots$ at random positions. The set $\overline{C}'$ can then be used as a public-key in the protocol.

6.4 Modified Generalized El-Gamal scheme

The computation of marginal sets can be done as explained in the above scheme and the rest of the steps could be implemented following the original statement in section 5.

The presentation of elements in the public sequences, by taking permutations of the elements or adding extra elements, to hide the true elements which will be used in computations and the length of the words, could be further followed as proposed in [20] to achieve better security, which resembles our proposals as well to a certain extent.

7. Conclusion and Future studies

This research presents another novel approach to cryptography using an intractable problem of determining

automorphisms and generating elements of a finite group. While on the contrary many of the previous cryptographic schemes using non-abelian groups have been proven to be vulnerable, this could be a suggestion of a pathway on new further studies leading to stronger schemes. The applicability of paths in Cayley graphs in these protocols imply that the HPP and HCP in Cayley graphs of non-abelian groups, and in fact the difficulty of determining random paths and cycles in Cayley graphs can also be considered in developing public-key cryptographic protocols. This study can be regarded as the first, where the paths in Cayley graphs were obtained for cryptosystems by utilizing the abstract properties of the graphs through mathematical proofs rather than suggesting to generate the graphs and a useful inception for further such researches. Exploring suitable platform groups for the implementation of the protocols, investigating novel algorithms where the use of paths and cycles in Cayley graphs is essential and beneficial, determination of improvements to enhance the security of the protocols, investigation of protocols which can employ the y-coordinates of the elements of the semidirect products we have mentioned are some of the related future research directions.

Acknowledgements

We would like to express our thanks to the Editor, Reviewers of the open access journal IJCSNS and everyone who had given kind comments and support during our research.

References

[1] 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 Math. Algorithms Appl.*, vol. 11, no. 05, p. 1930002, 2019.

[2] K. Kutnar, D. Marušič, D. W. Morris, J. Morris, and P. Šparl, "Hamiltonian cycles in Cayley graphs whose order has few prime factors," *Ars Math. Contemp.*, vol. 5, no. 1, Art. no. 1, Oct. 2011, doi: 10.26493/1855-3974.177.341.

[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," *Eur. Mod. Stud. J.*, vol. 04, no. 03, pp. 1–19, 2020.

[4] F. Maghsoudi, *Cayley graphs of order 6pq are Hamiltonian*. University of Lethbridge (Canada), 2020.

[5] D. W. Morris, "On hamiltonian cycles in Cayley graphs of order pqrs," *ArXiv Prepr. ArXiv210714787*, 2021.

[6] G. H. J. Lanel, T. M. K. K. Jinasena, and B. A. K. Welihinda, "A Survey of Public-Key Cryptography over Non-Abelian Groups," *IJCSNS*, vol. 21, no. 4, p. 289, 2021.

[7] B. Fine, M. Habeeb, D. Kahrobaei, and G. Rosenberger, "Aspects of nonabelian group based cryptography: a survey and open problems," *JP J. Algebra Number Theory Appl.*, 2011.

[8] T. C. Lin, "A study of non-abelian public key cryptography," *Int. J. Netw. Secur.*, vol. 20, no. 2, pp. 278–290, 2018.

[9] 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," 2000, pp. 166–183.

[10] I. Anshel, M. Anshel, and D. Goldfeld, "An algebraic method for public-key cryptography," *Math. Res. Lett.*, vol. 6, no. 3, pp. 287–291, 1999.

[11] I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, "New key agreement protocols in braid group cryptography," 2001, pp. 13–27. doi: 10.1007/3-540-45353-9_2.

[12] M. Garzon and Y. Zalcstein, "The complexity of Grigorchuk groups with application to cryptography," *Theor. Comput. Sci.*, vol. 88, no. 1, pp. 83–98, 1991.

[13] N. R. Wagner and M. R. Magyarik, "A public-key cryptosystem based on the word problem," 1984, pp. 19–36. doi: 10.1007/3-540-39568-7_3.

[14] S. Baba, S. Kotyad, and R. Teja, "A non-Abelian factorization problem and an associated cryptosystem.," *IACR Cryptol EPrint Arch*, vol. 2011, p. 48, 2011.

[15] L. Gu, L. Wang, K. Ota, M. Dong, Z. Cao, and Y. Yang, "New public key cryptosystems based on non-Abelian factorization problems," *Secur. Commun. Netw.*, vol. 6, no. 7, pp. 912–922, 2013.

[16] H. Hong, J. Shao, L. Wang, H. Ahmad, and Y. Yang, "Public Key Encryption in Non-Abelian Groups," *ArXiv Prepr. ArXiv160506608*, 2016.

[17] V. Roman'kov, "Two general schemes of algebraic cryptography," *Groups Complex. Cryptol.*, vol. 10, no. 2, pp. 83–98, 2018, doi: 10.1515/gcc-2018-0009.

[18] V. Shpilrain and G. Zapata, "Using the subgroup membership search problem in public key cryptography," *Contemp. Math.*, vol. 418, p. 169, 2006, doi: 10.1090/conm/418/07955.

[19] A. Ben-Zvi, A. Kalka, and B. Tsaban, "Cryptanalysis via algebraic spans," 2018, pp. 255–274.

[20] V. Roman'kov, "An improved version of the AAG cryptographic protocol," *Groups Complex. Cryptol.*, vol. 11, no. 1, pp. 35–41, 2019.

[21] I. Ilic and S. S. Magliveras, "Weak discrete logarithms in non-abelian groups," *J. Comb. Math. Comb. Comput.*, vol. 74, p. 3, 2010.

[22] L. C. Klingler, S. S. Magliveras, F. Richman, and M. Sramka, "Discrete logarithms for finite groups," *Computing*, vol. 85, no. 1–2, p. 3, 2009, doi: 10.1007/s00607-009-0032-0.

[23] I. Ilic, "The Discrete Logarithm Problem in Non-abelian Groups," *Computing*, vol. 1, p. 1, 2010.

[24] J. H. Cheon and B. Jun, "A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem," 2003, pp. 212–225.

[25] B. Tsaban, "Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography," *J. Cryptol.*, vol. 28, no. 3, pp. 601–622, 2015.

[26] S. K. Rososhek, "Modified matrix modular cryptosystems," *J. Adv. Math. Comput. Sci.*, pp. 613–636, 2015.