IRESPUB Journal of Natural & Applied Sciences
ISSN: 2770-3649
Volume: 1 | Issue: 2 | NOV - DEC 2021 | Available Online: www.irespub.com

A review of symmetric key cryptosystem using group representations

D. Samaila[1*], G. N. Shu'aibu[2], B. A. Madu[2]

[1]Department of Mathematics, Adamawa State University, Mubi, Nigeria

[2]Mathematical Sciences Department, University of Maiduguri, Borno State, Nigeria

ABSTRACT

The use of group representations in crypto-analysis has been a long-standing problem in group theory. Due to the nebulous presentations by few authors, this paper aimed at simplifying the procedure and to make it more secure than the existing literatures. We successfully presented the most radical techniques of symmetric key crypto-system using group theoretic approach. The procedure appears to be more secured and the result shows that different techniques can be used within the limit of representation theory for key selection, encryption and decryption. The difficulty in factorizing each as a product for large n makes it hard to break, and high flexibility for key selection also makes the DLP highly resistant to attacks. There is also an added advantage of easy and fast implementation.

KEYWORDS

conjugacy; commutators; representation; symmetric group; cryptography

CORRESPONDING AUTHOR*

D. Samaila

(1) INTRODUCTION

The idea of Group arises in nature as "sets of symmetries (of an object), which are closed under composition and inverses". A good example is the Symmetric group S_n of all permutations of n - objects; the group of even permutations in S_n called Alternating group A_n; the Dihedral group D_{2n} (also called geometric group) which is the group of symmetries of regular n-gon in the plane; the Orthogonal group O(3) called the group of distance-preserving transformations in the Euclidean space that fixes the origin. From geometric point of view, questions such as "Given a geometric object X, what is its group of symmetries?" aroused while the same question is reversed in Representation theory such as "Given a group G, what objects X does it act on?" and the attempt to answer such question leads to the classification of X up to isomorphism. One of the first major triumph in representation theory was the Burnside's pq-theorem which states that "a non-Abelian group of order p^xq^y with p,q prime and $x,y \in$ N cannot be simple (is solvable)". Representation theory also played a very important role in classification of finite groups.

Before the late 1970's, symmetric key systems were the only cryptosystems used for message transmission where two users that require communicating a message must create a unique common key to cipher or decipher the message [1]. In the year 1976, Diffie and Hellman invented a strong key exchange procedure that established a new system of cryptography. This system which was called the public-key cryptosystem was based on exponentiation in finite fields and gives the important background of the cryptosystems developed in this work.

Note that an action of a finite group G on any given set X is a homomorphism ϕ: $G \to S_X$, where S_X denote the symmetric group on X, see Peter [2]. Let K be a field and G be a group, then a representation of G is the pair (ρ, V) where V is a vector space over K and ρ is a homomorphism of G, given by ρ:G$\to$GL$_K$(V).Again, the conjugate of an element x by y in a group G is defined by $y^{-1}xy$. This convention fits well with right conjugation action so that $x^y = y^{-1}xy$ or equivalently, $x_y = y^{-1}xy$ and $x^e = x_e = x$ where e is the identity element of G and $(x^y)^z = (x_y)_z = x_{yz}$ for all $x, y, z \in G$.

(2) BACKGROUND OF SYMMETRIC-KEY CRYPTOSYSTEMS

In this section, the background of symmetric-key cryptosystem is discussed and we considered the Diffie-Hellman key exchange where a chosen finite field $GF(p)$ and a generator $g \in GF(p)$ are made public. Now, suppose that two users tagged "A" and "B" wish to create a common key. Then "A" randomly selects an integer x such that $2 \leq x \leq p-1$ and transmits the element g^x to "B" over an unsecure channel. User "B" also randomly selects an integer y such that $2 \leq y \leq p-1$ and transmits the element g^y to "A". The user's "A" and "B" can now compute the common key as $(g^y)^x = g^{xy}$ and $(g^x)^y = g^{xy}$ respectively. But it is clearly that finding an efficient discrete logarithm algorithm would make this system insecure since g, g^x and g^y are all known and made public. Other procedures that can be used to break this scheme is to compute the element g^{xy} from g^x and g^y without even computing x or y. This leads to a problem called Diffie-Hellman problem whose difficulty is equivalent to computing the discrete logarithms. Though, it remains unproven. Some partial results that were given about the equivalence of these problems were presented by Maurer *et al*, [3], Boneh*et al*, [4].

Again, public-key cryptosystem which was essentially a variant of Diffie-Hellman scheme was introduced by El-Gamal, [5] and Rivest, [6]. The algorithm can be described as follows: Supposed $GF(q)$ is known by public. Then user "A" randomly selects a generator $g \in GF(q)$ and an integer n, publishes (g, g^n) as the public-key while n remains secret. User "B", who requires to send a message $m \in GF(q)$ to "A", randomly selects an integer k such that $2 \leq k \leq q-1$, computes $m.(g^n)^k = m.g^{nk}$ and then sends the pair $(g^k, m.g^{nk})$ to "A". User "A" recovers m by computing $m.g^{nk}.(g^k)^{-n} = m.g^{nk}.g^{-nk}$.

Another public-key cryptosystem which was based on knapsack problem or subset sum problem has been invented by Hellman *et al*, [7]. The generated problem in this case was given as follows: Let $\{m_1, m_2, \ldots, m_n\}$ be a set of positive integers and let $w \in Z$ be arbitrary. Then find an n-bit integer $N = (b_n b_{n-1} \ldots b_1)_2$ such that $\sum_{i=1}^{n} b_i m_i = w$. This was also broken by Shamir, [8].

Generally, the above public-key cryptosystem presented the knapsack problem that is easy to solve [1]. It was successfully transformed to an instance of general knapsack problem that was difficult to solve, where the later knapsack set served as the public key. Its generality was however, been contradicted. Another algorithm for the knapsack problem which was polynomial in n was presented by Shamir, [8]. In 1988, Chor and Rivest [9], also presented another type of knapsack cryptosystem which was again broken by Vaudenay, [10]. Another important public key cryptosystem is the *Elliptic Curve Cryptosystem* (*ECC*) which was first proposed independently by Koblitz, [11] and Miller, [12]. The systems are based on group of points on an elliptic curve defined over a finite field. Some systems can be modified to work in these groups such as Diffie-Hellman key exchange or El-Gamal system, and is similar to a cryptosystem using p-groups [13]. But group of transformations gives more general and secure crypto-system that cannot be broken easily.

2.1 The Discrete Logarithmic Problem (DLP)

The security strength of any modern cryptosystems totally depends on the intractability of the corresponding Discrete Logarithm Problem (DLP) [14]. Now, let G be a cyclic group whose order is n and $g \in G$ be a generator. Let $\gamma \in G$ be arbitrary. Then the discrete logarithm of γ to base g, denoted by $\log_g \gamma$ is an integer x, $0 \leq x < n$ such that $\gamma = g^x$ [15]. We state the discrete logarithm problem as follows: Given an element $\gamma \in G$, find an integer x that satisfies $\gamma = g^x$. The discrete logarithm has some properties that are similar to any ordinary logarithm as described by Andre, [16], in describing any given finite group by short first-order sentence. For example $\log_g a.b = \log_g a + \log_g b$ and $\log_g a^k = k.\log_g a$. In the Generalized Discrete Logarithm Problem (GDLP), G is an arbitrary group and $g \in G$ is not necessarily a generator. In this case, the integer x may not exist. In last two decades, there have been substantial improvements in discrete logarithm algorithms. However, the problem still appears to be intractable. We therefore, discussed the complexity of some algorithms for solution of discrete logarithm using group of transformations.

(3) SYMMETRIC CIPHER

A cipher can be described as a method of making a message unreadable either by rotating, reflecting or transposing it so as to look meaningless to the general public. The most popular and simplest cipher is the *Caesar cipher*, used by Julius

Caesar during the Gallic Wars. In this case, each letter of the alphabet is shifted (forward or backward) to a fixed number of places (Caesar normally used a shift of three places) [17]. The pattern for the arrangement of the alphabets is cyclic so that the letter following Z is A. The two known procedures for any cipher are the encryption procedure and the decryption procedure.

In this section, we built on the work of Javad et al, [1], on Cryptosystem based on the group of symmetries, S_n. If S is a non-empty set and G = {σ:S→S|σ is a bijection}, then G is called the group of transformations on S (or Symmetric group on S), denoted by S_s. If S is a finite set of order *n*, then G is called a permutation group of degree *n* denoted by S_n. Let S = $\{a_1, a_2, \ldots, a_n\}$. For any $\alpha \in S_n$, let $\alpha(a_i) = a_j, 1 \le i, j \le n.$ Thus, each $\alpha \in S_n$ may be regarded as a permutation of $\{a_1, a_2, \ldots, a_n\}$. This leads to a notion of symmetric cipher.

Supposed that the set $K_n = \{1,2,\ldots,n\}$, then a function *f* defined by $f : K_n \to K_n$ on K_n such that $1 \le f(i) \le i$ for all $1 \le i \le n$ is called a sub-exceedant function (Javad et al, 2008). The method in this section is that if E_n is the set of all sub-exceedant functions defined on K_n and the function *f* is represented by the word $f(n)f(n-1)\ldots f(1)$, then for *n* = 5, $f = 41023$ is a sub-exceedant function over K_5 in which $f(5) = 4,\ f(4) = 1,\ f(3) = 0,\ f(2) = 2$ and $f(1) = 3$. Note that in this literature, $CardE_n = (n+1)!$ and we have the following result:

Lemma 3.1: Let $\varphi : E_n \to S_{n+1}$ be a mapping that is associated with the sub-exceedant function *f* and let σ_f be a permutation on product of transpositions defined by $\sigma_f = (1f(1))(2f(2))\ldots(nf(n))$. Then φ is a bijection mapping E_n onto S_{n+1} [1].

We therefore present in this work, some basic techniques for which group theory can be used to construct variants of the Diffie–Hellman key agreement protocol. Since the protocol uses cyclic subgroups of finite presentation, the approach in this work is to use finite groups (not necessarily Abelian) that can be efficiently represented and manipulated, and possesses cyclic subgroups. As seen in Section two above, the shift cipher is one of the symmetric ciphers in which the encryption procedure consists of shifting the letters in the message by a fixed number of steps. The fixed number is called the key, k∈Z^+, also called the length of permutation applied to the positions of the letters [18]. Message will be more secure by splitting the 26 English alphabets into two (or more) subsequences (see Table 1) of prime order in the ratio G_1:G_2= 13:13. In this case, we have to assign two different keys k_1 and k_2 (or more) for the corresponding subsequences.

With a little twist of the alphabets (through $\frac{2\pi}{13}$ radian), the following table is obtained.

TABLE 3.1: The two Subsequences of the 26 Alphabets

G_1	A	C	E	G	I	K	M	O	Q	S	U	W	Y	A
G_2	B	D	F	H	J	L	N	P	R	T	V	X	Z	B

Note that the procedures for the encryption and decryption are essentially the same. Hence, the named Symmetric Cipher [19].

3.1 Discrete Logarithmic Problem Using Finite Group

Let G be a cyclic group and let $g \in G$ be a generator. Let $q \in G$ be arbitrary. Then there exists an integer *n* such that $g^n = q$. As discussed by Simon, [15], if the Discrete Logarithmic Problem (DLP) were easy, then so is the Diffie– Hellman Problem (DHP). This in turn implies that Diffie– Hellman key agreement protocol is not secure. This paper aimed at finding more difficult instances and procedures for the DLP in order to make the cryptosystem more secure. It is observed that difficulty of the DLP only depends on the representations of the group rather than its isomorphism class.

According to McCurley, [14], the most efficient algorithm for solving the discrete logarithmic problem is the index-calculus algorithm which was initially introduced by Kraitchik. If G is a finite group such that |G| = *n* and H is a subgroup of G where large subsets of elements of G can be expressed as the product of elements $\alpha \in H$, then we construct the algorithm for the elements of H as follows:

i. Select a random integer *j* such that $1 \le j \le n-1$ and compute α^j;

ii. Express each α^j as

$$\alpha^j = \prod_{i=1}^{|H|} N_{ij}^{\sigma_i} \text{ where } \sigma_i \ge 0 \text{ and } N_i \in H \qquad 3.1$$

Taken the logarithm of both sides of equation 3.1 to base α, we have

$$j = \prod_{i=1}^{|H|} \sigma_i \log_\alpha N_{ij} \qquad 3.2$$

By repeating steps (i) and (ii), one can obtain a set of equations of the form 3.2, then solve it in order to find the logarithm of elements of H and for any $\beta \in H$, $\log_\alpha \beta$ can be computed as follows:

i. Randomly select an integer *r* such that $1 \le r \le n-1$ and compute $\beta \cdot \alpha^r$;

ii. Express $\beta \cdot \alpha^r$ as

$$\beta \cdot \alpha^r = \prod_{i=1}^{|H|} N_{ir}^{\rho_i} \text{ where } \rho_i \ge 0 \text{ and } N_i \in H \qquad 3.3$$

Then taken the log of both sides of equation 3.3, we have

$$\log_\alpha \beta = \prod_{i=1}^{|H|} \rho_i \log_\alpha N_{ij} - r .$$

Now, if $\{H_1, H_2, \ldots, H_r\}$ are subgroups of *G* as specified above such that $e = H_1 < H_2 < \ldots < H_r = G$, and K_i is the set of all cosets representatives of H_{i-1} in H_i, then $\varphi = \{K_1, K_2, \ldots, K_r\}$ is a logarithmic signature (also called a cover for *H*) called transversal logarithmic signature for *G*, see also in Ayan, [13]. This is certainly hard to break in general since given an element $h \in H$ and a cover φ for *H*, it is often difficult to obtain a factorization $h = \phi_1 k_1 \phi_2 k_2 \ldots \phi_r k_r$ associated with φ which is sometimes a discrete logarithmic problem.

(4) RESULT AND DISCUSSION

4.1 Symmetric Group in Crypto-Analysis

In this section, the idea of encryption and decryption is analyzed and used to derive new techniques for crypto-analysis using group presentations. Let $\sigma \in S_m$. Then the *Sign* of σ is define as

$$Sign\sigma = \begin{cases} -1 \ \ if \ \ \sigma \ \ is \ \ odd \\ 0 \ \ \ if \ \ \sigma \ \ is \ \ not \ \ a \ \ permutation \\ +1 \ \ if \ \ \sigma \ \ is \ \ even \end{cases}$$

and if $\varphi : [m] \to [m]$ is a function on $[m] = \{\pm 1, \pm 2, \ldots, \pm m\}$ such that $\varphi(i) = i$ and $\varphi(-i) = -\varphi(i) = -i$ for all $i \in [m]$, then φ is called a sub-exceedant function (see Section 3). We denote the set of all such functions by $\aleph_m$ and

each function φ on $[m]$ can be represented by the word $\varphi(1)\varphi(2)...\varphi(m)$. Here, the set $\aleph_m$ can be derived as

$\aleph_1 = \{1\}$; $\aleph_2 = \{11,12\}$; $\aleph_3 = \{111,112,113,122,123,133\}$; and so on.

Hence, $Card\aleph_m = m!$ and $\aleph_m$ is a group of transformations. Now, define a function $\theta : \aleph_m \to S_m$ from $\aleph_m$ to S_m such that $\theta(\varphi) \mapsto \sigma_\varphi = (1\varphi(1))(2\varphi(2))...(m\varphi(m))$ (Lemma 3.1). Then $\theta : \aleph_m \to S_m$ is a bijection. To see this, since $Card\aleph_m = |S_m| = m!$, it suffices to show that θ is injective. Let $\varphi, \pi \in \aleph_m$. Our aim is to show that $\theta(\varphi) \neq \theta(\pi)$ whenever $\varphi \neq \pi$. Suppose $\theta(\varphi) = \theta(\pi)$, then for some integer r, $\sigma_\varphi = \sigma_\pi$ so that $(1\varphi(1))(2\varphi(2))...(r\varphi(r)) = (1\pi(1))(2\pi(2))...(r\pi(r))$. In particular, $\sigma_\varphi(r) = \sigma_\pi(r)$ and since $\sigma_\varphi(r) = \varphi(r)$ and $\sigma_\pi(r) = \pi(r)$ by definition, we have $\varphi(r) = \pi(r)$.

Now, let $r = m-1$. Then

$(1\varphi(1))(2\varphi(2))...((m-1)\varphi(m-1)) = (1\pi(1))(2\pi(2))...((m-1)\pi(m-1))$

and $\sigma_\varphi(m-1) = \sigma_\pi(m-1)$ implies that $\varphi(m-1) = \pi(m-1)$. Hence, by induction on m, $(1\varphi(1))(2\varphi(2))...(m\varphi(m)) = (1\pi(1))(2\pi(2))...(m\pi(m))$ for all m so that $\varphi(i) = \pi(i)$ for all $i \in [m]$ if and only if $\varphi = \pi$ as required.

Supposed now that $\sigma_1, \sigma_2, ..., \sigma_m$ are elements of S_m such that $\sigma_i \neq \sigma_j$ for all $i \neq j$, then $\sigma_i\sigma_j \neq \sigma_j\sigma_i$ and $\sigma_i\sigma_j = \sigma_j\sigma_i$ if and only if σ_i and σ_j are disjoint. Note that except for identity permutation, each $\tau \in S_m$ can be expressed as a product of disjoint cycles as $\tau = \prod_{i=1}^{m} \sigma_i$, where each σ_i is an m_i-cycle and

$$|\sigma| = lcm(m_1, m_2, ..., m_m).$$

Let $\tau = \sigma_1\sigma_2...\sigma_m$ be isomorphic to an m-digit number in $[m]$, $0 < \varphi_i \leq i$ and $0 < \sigma_m \leq m$. Then we interpret $\tau = \sigma_1\sigma_2...\sigma_m$ as a sub-exceedant function representation given by $\varphi(1)\varphi(2)...\varphi(m)$ in which $\varphi(1) = \sigma_1$, $\varphi(2) = \sigma_2$, ..., $\varphi(m) = \sigma_m$.

Now, we present the process of key selection and encryption-decryption techniques from the above construction as follows:

Key Selection:

- Select a large integer m such that $|S_m| \geq 100!$;
- Select an element $\gamma \in S_m$ such that $G_\gamma = \langle\gamma\rangle$;
- Randomly select an integer $q \in [m], 1 < q < |G_\gamma|$ such that $\gamma^q = \sigma \in S_m$;
- Publish (γ, σ) as a public key whereas the key q is kept private.

Encryption: Romeo encrypt the message $g \in [m]$ as follows:

- Transform $g \in [m]$ to $g' \in S_m$ using the function $\theta : \aleph_m \to S_m$;
- Randomly select an integer $\eta \in [m]$ such that $1 < \eta < m$;
- Compute and transmit the pair $(\gamma^\eta, g' \cdot \sigma^\eta)$ to Juliet;

Decryption: Juliet decrypt the message as follows:

- Compute $(\gamma^{\eta})^{q} = \gamma^{\eta q}$ so that $(\gamma^{\eta})^{q} = \sigma^{\eta}$ and $(\gamma^{-\eta})^{q} = \sigma^{-\eta}$;
- Compute g' from $(\gamma^{\eta} \cdot (\gamma^{-\eta})^{q}, g' \cdot \sigma^{\eta} \cdot \sigma^{-\eta})$;
- Recover *g* by computing the integer representation of $g' \in S_m$.

4.2 Computing a Common Commutator

Let *G* be a non-Abelian two generator group of order *m*. Then $G \cong D_n$ if and only if $m = 2n$. Supposed $G = \{\alpha_1, \alpha_2, ..., \alpha_n, \beta_1, \beta_2, ..., \beta_n\}$ is public, then

- Romeo randomly picks a private word $x \in \{\alpha_1, \alpha_2, ..., \alpha_n\}$ and then sends $\beta_1^x, \beta_2^x, ..., \beta_n^x$ to Juliet;
- Juliet randomly picks a private word $y \in \{\beta_1, \beta_2, ..., \beta_n\}$ and then sends $\alpha_1^y, \alpha_2^y, ..., \alpha_n^y$ to Romeo;
- Romeo computes x^y and $x^{-1}x^y$ and Juliet computes y^x and $(y^{-1}y^x)^{-1}$;
- The computed secret key is $[x, y] = x^{-1}y^{-1}xy$.

We now provide a generalized logarithmic signature for finite groups that are generally not Abelian. Let $G \cong D_n$ be a finite non-Abelian group, $H \subset G$ and $r \in [m]^+$. From Section 3,Suppose $H_i = \{\varphi_{i1}, \varphi_{i2}, ..., \varphi_{ik_i}\}$ is a finite sequence consisting of elements of *G* of length $k_i > 1, 1 \le i \le r$ and $\varphi = \{H_1, H_2, ..., H_r\}$ represent the ordered sequence of H_i. Then the representative φ is a cover for $H \subset G$ in *G* if each $h \in H$ can be decomposed into a unique product $h = h_1 h_2 ... h_r$ such that each $h_i = \varphi_{ik_i}$ in H_i. Such unique decomposition is called a logarithmic signature for *H*.

Again if*g*, *h*∈*G*,then the conjugation of *g* by *h* is given by $g^h = h^{-1}gh$.Now, let *M* and *N* be cyclic subgroups of *G*, *M*≠*N* and $\varphi : G \to G$ be a homomorphism from *G* to *G*defined by $g^{\varphi(x)} = \varphi^{-1}(x)gx$ for all *g*∈*G*. Then for all *m*∈*M*, *n*∈*N*, [*m*, *n*] = *e*.

Now, if Romeo and Juliet want to generate a common secrete key, then the procedure is as follows:

- Romeo randomly selects an element *m*∈*M*, computes $g^{\varphi(m)} = \varphi^{-1}(m)gm$, and sends it to Juliet;
- Juliet randomly selects an element *n*∈*N*, computes $g^{\varphi(n)} = \varphi^{-1}(n)gn$, and sends it to Romeo;
- Romeo then computes $K_m = (g^{\varphi(n)})^{\varphi(m)}$, while Juliet computes $K_n = (g^{\varphi(m)})^{\varphi(n)}$.

But $\varphi(m)\varphi(n) = \varphi(n)\varphi(m)$ from the definition of *M* and *N*.This in turn means that $K_m = K_n$ as group elements whose representations might be different. The secret key *K*is then the normal form of K_h and K_m.

4.3 Public Key Exchange

The major problem for the symmetric cipher is the privacy of the key. Also, changing the key often is a good idea. But an existing problem is how to accomplish an exchange of a new key if the present key has been broken. Now, let *p* be a prime number and *g* be a primitive root modulo *p*. In practice, *p* has to be large but for demonstration purposes, let *p* = 13 and *g* = 2.

Then with *p* and *g* above, Romeo and Juliet can perform a secure key exchange over an unsecure channel as follows:

- Romeo randomly selects a *secret* number say *m*, e.g. *m* = 23 and calculate
  - X = 2^m (mod 13); $2^{23} \equiv 7$ (mod 13).
  - He sends the integer X = 7 to Juliet over an unsecure channel.

- Juliet randomly selects a *secret* number say n, e.g. $n = 31$ and calculate
  - $Y = 2^n \pmod{13}$; $2^{31} \equiv 11 \pmod{13}$.
  - She sends the integer Y = 11 to Romeo over an unsecure channel.
- Romeo now receives the integer Y from Juliet and calculates the key modulo 13, i.e.
  $K = Y^m \equiv 6 \pmod{13}$
- Juliet also receives the integer X from Romeo and calculates the key modulo 13, i.e.
  $K = X^n \equiv 6 \pmod{13}$.

The calculated key K must be the same for both Romeo and Juliet since $mn = nm$, that is, $(2^m)^n = (2^n)^m \pmod{13}$ for all $m,n \in Z$. With the secret key at hand, the two friends can now exchange message safely.

4.4 Mathematical Model for Symmetric Cipher

Let G be a finite group which acts on a non-empty set X and let δ be a representation of G. Then the set of all symmetry operations considered on the elements of X can be modeled as an action $\delta: G \times X \to X$ that satisfies $\delta(g, x) = g \cdot x$ for all elements $g \in G$ and $x \in X$. The operation g for which $g \cdot x = x$ formed the symmetric group of x, a subgroup of G such that if for some g, $g \cdot x = y$, then the elements x and y are symmetrical. If each $g \in G$ is a bijection, then $g: X \to X$ acts on the set of functions $\varphi: X \to V$ to a vector space V by $(g\varphi)(x) = \varphi(g^{-1}(x))$.

Now from Table 3.1, the following bijection describe the encryption and decryption procedures of a symmetric cipher with respect to the assigned keys $i = K_1$ and $j = K_2$.

$$\varphi : G_1 \to G_1 \text{ by } \varphi(g_n^1) = g_{(n \pm i) \bmod m}^1 ;\ g_0^1 = g_m^1,\ 1 \le n \le m, \qquad 4.1$$

$$\rho : G_2 \to G_2 \text{ by } \rho(g_n^2) = g_{(n \pm j) \bmod m}^2 ;\ g_0^2 = g_m^2, 1 \le n \le m, \qquad 4.2$$

for all $g^1 \in G_1$ and $g^2 \in G_2$.

Again, if $\varphi : [m] \to [m]$ is a sub-exceedant function such that $\theta : \aleph_m \to S_m$ is a bijection from $\aleph_m$ to S_m, then

$$\theta_\varphi(i) \mapsto \sigma_\varphi(i) = \prod_{r=1}^{m} (r_i \varphi(r)) \text{ for each } i \qquad 4.3$$

Note that in this case, the representations φ, ρ and θ are isomorphism and the collection of all such bijections formed a group with respect to composition of functions.

4.5 Conclusion

The fact that there exists an isomorphism between every finite group and some group of permutations, many cryptosystems based on finite groups can be translated to systems using the finite group S_n. The results in this paper were described by illustrating the symmetric groups that are analog of the Generalized El-Gamal system. The defined binary operation in the group S_n, called composition of functions can be implemented using n assignments. The derived scheme can be generated in a very easy way and is obviously resistant to attacks. Also, the key selection and procedures presented in this paper has very low complexity among other known schemes in representation theory. One of the disadvantage to be encounter in this case is the relatively large memory requirement and the organization of permutations. Despite these disadvantages, the procedure preserves high security and strength in the abstract and theoretical aspects. This is because the group S_n can generate a cryptosystems with more unconditional security than other existing cryptosystems. This proofs theoretically that the structure of the symmetric group S_n is appropriate for cryptosystems.

To achieve the aim of making a message more secured in Symmetric cipher, one can split the English alphabets into two or more subsequences. From Table 3.1, there are 13!×13! possible arrangements and the concept requires two different

keys K_1 and K_2 for the corresponding subsequences. For example, let $K_1 = 3$ and $K_2 = 7$ for the subsequences G_1 and G_2 respectively. Then with these values, the message

CUT YOUR COAT ACCORDING TO YOUR SIZE

can be encrypted as

IAH EUAF IUGH GIIUFROBM HU EUAF YONK

Equivalently, if an encrypted message is received, such as

HVK PUE OY FABBOBM VUSK

then using the keys provided for each subsequence in a reverse form (as an inverse operator), the message is decrypted as follows:

THE BOY IS RUNNING HOME

Hence, the encryption and decryption procedures are elements of Cyclic group.

REFERENCES

[1] Javad N. D., Ehsan M. and Ali Z.: A Cryptosystem Based on the Symmetric Group S_n. International Journal of Computer Science and Network Security, 2008; VOL. 8(2), pp226-234.

[2] Peter Mayr: Group Action and applications; 2019.

[3] Maurer U. and Wolf S.: The relationship between breaking the Diffe-Hellman protocol and computing discrete logarithms, SIAM Journal on Computing,1999; 28(5) pp.1689-1731.

[4] Boneh D. and Lipton R.: Algorithms for black-box fields and their applications to cryptography, Advances in Cryptology-CRYPTO '96, Lecture Notes in Computer Science, Springer-Verlag,1996; (1109) pp.283-297.

[5] El-Gamal T.: A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, 1985;(31) 469-472.

[6] Rivest R., Shamir A. and Adleman L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Comm. ACM,1978;(21) pp.120-126.

[7] Hellman M. E. and Merkle R. C.: Hiding information and signatures in trapdoor knapsacks, IEEE Transactions on Information Theory, 1978;(24) pp.525-530.

[8] Shamir A.: A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem, Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press,1982; pp.145-152.

[9] Chor B. and Rivest R.: A knapsack-type public key cryptosystem based on arithmetic in finite fields, IEEE Transactions on Information Theory,1988;(34) pp.901-909.

[10] Vaudenay S.: Cryptanalysis of the Chor-Rivest Cryptosystem, Advances in Cryptology-CRYPTO '98, Lecture Notes in Computer Science, 1462 Springer Verlag, 1998; pp.243-256.

[11] Koblitz N.: Elliptic curve cryptosystems, Mathematics of Computation, 1987;(48) pp.203-209.

[12] Miller V.: Uses of elliptic curves in cryptography, Advances in Cryptology- CRYPTO '85, Lecture Notes in Computer Science, Springer-Verlag, 1986;(218) pp.417-426.

[13] Ayan M.: The MOR Cryptosystems and Finite p-groups, Contemporary Mathematics,2015; Volume 633, pp.:81-95.

[14] McCurley K.: The discrete logarithm problem, Cryptology and Computational Number Theory, volume 42 of Proceedings of Symposia in Applied Mathematics, American Mathematical Society,1990; pp.49-74.

[15] Simon R. B., Carlos C. and Ciaran M.: Group theory in Cryptography, Egham, Surrey TW20 0EX, United Kingdom, 2010.

[16] Andre N. and Katrin T.: Describing Finite Groups by Short First-Order Sentences, Israel Journal of Mathematics, 2014; pp.1-13.

[17] Robert C.: Codes and Ciphers; Julius Caesar, the Enigma and the Internet, Cambridge University Press, Cambridge, 2002.

[18] Michal S.: New Results in Group Theoretic Cryptology. Ph.D. Thesis, Florida Atlantic University, Boca Raton, 2006.

[19] Douglas R. S.: Cryptography; Theory and Practice, 2nd ed, CRC Press,New York, NY, 2002.