International Journal of Latest Research in Science and Technology **ISSN (Online):2278-5299**
Volume 7, Issue 6: Page No.18 -22 , November – December 2018
https://www.mnkpublication.com/journal/ijlrst/index.php

THE PLACE OF GROUP THEORY ON RECENT TRENDS IN INFORMATION TRANSMISSION- ENCRYPTION AND DECRYPTION

Samuel Hwere Tsok[1], Augustine Pwasong[2] , Naphtali B. Jelten[3]

Faculty of Natural and applied Science Department of Mathematics Plateau State University Bokkos[1]
Faculty of Natural Sciences Department of Mathematics University of Jos [23]
tsoksam@gmail.com

Abstract* : *Progression in computing powers and parallelism technology are creating obstruction for credible security especially in electronic information swapping under cryptosystems, this led to this paper presenting the several ways in which group theory can be used to construct various key agreement protocols and finally presented the secure signatures with RSA, which enable identity verification during encryption and decryption of information.

Keywords **– Encryption, Decryption, Information transmission, Group Theory, Key agreement protocols and Secure signatures**.

I. INTRODUCTION

Cryptography can be categories as a branch of mathematics and computer science which further relates with information security and computer engineering. Krptos (“hidden”) is a Greek word gives birth to English word called cryptography- an art of changing the actual face look of information as well as converting it into unreadable form. Cryptography further relies on encryption techniques (symmetric and asymmetric) to encode the actual text message (Plain Text) with the use of secret code called key. The process of encoding or encrypting the plain text is referred as enciphering or encryption and the vise versed process is called deciphering or decryption. Symmetric encryption requires a single shared secret code known as private key and asymmetric encryption is based on two key(s); private key and public key where private key remains secret and public key is publicly available. In asymmetric encryption public key is used to encrypt the message and private key is used to decrypt the same message. Group based cryptosystems have not yet led to practical schemes to rival Rivest Shamir Adlenman (RSA) and Diffie-Hellman, however the ideas are interesting and different perspective leads to some worthwhile group theory. As asserted by (Laurence, 2008) on cryptography using elliptic curves is excellent follow-up, elliptic curve-based cryptography is becoming the norm for the current generation of public key cryptosystems. However, our paper is mainly on mathematical aspect of cryptography consciously aim to consider group theory in encryption and decryption of information transmission.

A. GROUP THEORY

Group theory is the branch of pure mathematics which emanate from abstract algebra. Due to its abstract nature, it was considered as an arts subject rather than a science subject. In fact, it was seeming to be pure abstract and not practical (Tsok, 2013).

Modern group theory is a very active mathematical discipline which studies groups in their own rights. To explore groups, mathematicians have defined certain terms that are analogous to those of sets for a better understanding of the concept. These terms include subgroups, quotient groups and simple groups. In addition to their abstract properties, group theory also studies the different ways in which a group can be represented in such a way that it can be appreciated by those who dread it (Tsok, 2013). This is called group representation. The study of groups arose early, in the nineteenth century in connection with the solution of equations. Originally a group was a set of permutations with the property that the combination of any two permutations again belongs to the set. Subsequently this set, not necessarily of permutations, together with a method of combining its elements that is subject to a few simple laws. The theory of groups occupies a central position in mathematics. Modern group theory arose as an attempt to find the roots of a polynomial in terms of its coefficients. Groups now play a central role in number of apparently unconnected subjects as in Crystallography and Quantum Mechanics, Coding theory and Cryptography, Physics, Chemistry, Biology as well as non- sciences like Games and Sociology (Anthropology). Although groups arose in connection with other disciplines, the study of groups is so exciting. Currently there is vigorous research in the subject, and it attracts the interest of many great Mathematicians. For this very reason this work wishes to explore how group theory can benefit the field of Cryptography in the area of

Publication History
Manuscript Received : 14 December 2018
Manuscript Accepted : 10 January 2019
Revision Received : 24 January 2019
Manuscript Published : 29 January 2019

encryption and decryption of information transmission. One of the main methods of encrypting data is the RSA encryption system. The algebraic structure that is at the heart of this method is that of a group. To motivate the definition of a group, let discuss the main terms used in the RSA encryption system. Let R be a ring. Recall that a unit of R is an element having a multiplicative inverse. Recall also that if *a,b* are units of R, then so is *ab*, since *ab* has $b^{-1}a^{-1}$ as its multiplicative inverse. Then R^* is the set of all units of R. In other words. $R^* = \{a \in R : there\ is\ a\ c \in R\ with\ ac = ca = 1\}$. By the statement above, if two elements are multiply which are of R^*, the result is another element of R^*. Therefore, multiplication induces a binary operation on the set, R^*. Note three properties of this binary operation multiplication on R^* is associative, $1 \in R^*$, so R^* has an identity; and each element of R^* has an inverse in R^*. It is these properties that make up the definition of a group.

Definition 1. Let G be a nonempty set together with a binary operation $*$ *on* G. Then the pair (G, $*$) is said to be a group if;
(i). $a * (b * c) = (a * b) * c$ for all $a, b, c \in G$
(ii).
$There\ is\ an\ e \in G\ such\ that\ e * a = a * e\ for\ all\ a \in G$
(iii).
$For\ each\ a \in G\ there\ is\ a\ an\ element\ b \in G\ with\ a * b = b * a = e$

B. ENCRYPTION AND DECRYPTION

Encryption is a mechanism by which a message is transformed so that only the sender and recipient can see. For instance, suppose that Alice wants to send a private message to Bob. To do so, she first needs Bob's public-key; since everybody can see his public-key, Bob can send it over the network in the clear without any concerns. Once Alice has Bob's public-key, she encrypts the message using Bob's public-key and sends it to Bob. Bob receives Alice's message and, using his private-key, decrypts it.

Cryptography is about communication in the presence of an adversary. It encompasses many problems (encryption, authentication, key distribution to name a few). The field of modern cryptography provides a theoretical foundation based on which might be understand what exactly these problems are, how to evaluate protocols that purport to solve them, and how to build protocols in whose security one can have confidence.

II. METHOD

A. PUBLIC-KEY ENCRYPTION

The idea of a public-key cryptosystem (PKC) was proposed by Diffie and Hellman in their pioneering paper (Whitefield, 1976). Their revolutionary idea was to enable secure message exchange between sender and receiver without ever having to meet in advance to agree on a common secret key. They proposed the concept of a trapdoor function and how it can be used to achieve a public-key cryptosystem. Shortly thereafter Rivest, Shamir and Adelman proposed the first candidate trapdoor function, the RSA. The story of modern cryptography followed.

The set up for a public-key cryptosystem is of a network of users u_1, u_n rather than a single pair of users. Each user *u* in the network has a pair of keys $<P_u; S_u>$associated with, the public key P_u which is published under the user's name in a "public directory" accessible for everyone to read, and the private-key S_u which is known only to *u*. The pairs of keys are generated by running a key-generation algorithm. To send a secret message *m* to *u* everyone in the network uses the same exact method, which involves looking up P_u, computing $E(P_u, m)$ where *E* is a public encryption algorithm, and sending the resulting ciphertext *c* to *u*. Upon receiving ciphertext *c*, user *u* can decrypt by looking up private key S_u and computing $D(S_u, c)$ where *D* is a public decryption algorithm. Clearly, for this to work there is need that $D(S_u, E(P_u m)) = m$.

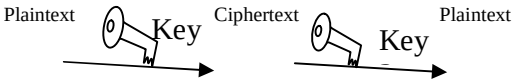

Figure 1: Public key (Asymmetric) cryptography, PKC uses key one for encryption while key two for decryption.

A particular PKC is thus defined by a triplet of public algorithms. This is the key "generation, encryption and decryption" algorithms (*G,E,D*). Now let formally define a public-key encryption scheme. For now, the definition will say nothing about "security" of a scheme.

Definition 2 A public-key encryption scheme is a triple, (*G,E,D*), of probabilistic polynomial-time algorithms satisfying the following conditions

(i) key generation algorithm: a probabilistic expected polynomial-time algorithm *G*, which, on input 1^k (the security parameter) produces a pair (*e, d*) where *e* is called the public key, and *d* is the corresponding private key. (Notation: $(e, d) \in G(1^k)$). Let also refer to the pair (*e, d*) a pair of encryption/decryption keys.

(ii). An encryption algorithm: a probabilistic polynomial time algorithm *E* which takes as input a security parameter 1^k, a public-key *e* from the range of $G(1^k)$ and string $m \in \{0,1\}^k$called the message and produces as output string $c \in \{0,1\}^k$called the ciphertext. the notation $c \in E(1^k, e, m)$ is used to denote *c* being an encryption of message *m* using key *e* with security parameter *k*. When clear, shorthand is also use as $c \in Ee(m)$, or $c \in E(m)$.

(iii) A decryption algorithm: a probabilistic polynomial time algorithm *D* that takes as inputs a security parameter 1^k, a private-key *d* from the range of $G(1^k)$, and a ciphertext *c* from the range of $E(1^k, e, m)$,and produces as output a string $m' \in \{0,1\}^*$ such that for every pair (*e,d*) in the range of $G(1^k)$, for every *m*, for every $c \in D(1^k, e, m)$, the $prob(D(1^k, d, c) \neq m')$ is negligible.

(iv). Furthermore, this system is "secure"

To use a public-key encryption scheme *(G, E, D)* with security parameter 1^k, user A runs $G(1^k)$ to obtain a pair *(e, d)*

of encryption/decryption keys. User A then "publishes" *e* in a public file, and keeps private *d*. If anyone wants to send A a message, then need to lookup *e* and compute $E(1^k, e, m)$. Upon receipt of $c \in E(1^k, e, m)$, A computes message $m = D(1^k, d, c)$.

B. PRIVATE-KEY ENCRYPTION

The symmetric setting considers two parties who share a key and will use this key to imbue communicated data with various security attributes. The main security goals are privacy and authenticity of the communicated data.

Figure 2: Secret key (Symmetric) cryptography, SKC uses single key for both encryption and decryption.

In single or private key cryptosystems, the same key is used for both encryption and decryption messages. To encrypt a plaintext message, then apply to the message some function which is kept secret say f. This function will yield an encrypted message. Given the encrypted form of the message, this can recover the original message by applying the inverse transformation f^{-1}. The transformation *f* must be relatively easy to compute, as must f^{-1}; however, f must be extremely difficult to guess at, if only examples of coded messages are available.

Example 1: One of first and most famous private key cryptosystems was the shift code used by Julius Caesar. First Digitize the alphabet by letting *A = 00, B = 01, . . ., Z = 25*. The encoding function will be $f(p) = p + 3 mod 26$ that is $A \to D, B \to E, \ldots, Z \to C$. The decoding function is the $f^{-1}(p) = p - 3 mod 26 = p + 23 mod 26$.

Suppose that one received the encoded message DOJHEUD. To decode this message, first digitize it as 3,14,9,7,4,20,3 next apply the inverse transformation to get 0, 11, 6,4, 1, 17,0, or ALGEBRA. Notice here that there is nothing special about either of the number 3 or 26. One can use a larger alphabet or a different shift.

III. RESULTS

Schemes base on Group - Based Cryptography

Considering other several ways in which group theory can be used to construct various key agreement protocols. Since Diffie–Hellman key agreement protocol uses a cyclic subgroup of a finite group G, one approach is to search for examples of groups that can be efficiently represented and manipulated, and that possess cyclic subgroups with a DLP that seems hard. Various authors have suggested using a cyclic subgroup of a matrix group in this context, but some basic linear algebra shows that this approach is not very useful: the DLP is no harder than the case when G is the multiplicative group of a finite field; as cited by Menezes and Vanstone (Alfred, 1992) and details by Biggs, (Norman,2007) has proposed representing an abelian group as a critical group of a finite graph; but Blackburn has shown that this protocol is insecure (Blackburn, 2010). An approach (from number theory rather than group theory) that has had more success is to consider the group of points on an elliptic curve, or Jacobians of hyper elliptic curves. According to Galbraith and Menezes (Steven,2005) has survey of this area. All the works discussed use representations of abelian (indeed, cyclic) groups. What about non-abelian groups? The first work to use non-abelian groups that we are aware of is due to (Neal,1985). (Gonz´alez Vasco and Steinwandt (Maria, 2004)

A. Diffie–Hellman Key Agreement Protocol

(Whitfield, 1976) asserted that. Let G be a cyclic group, and g a generator of G, where both g and its order d are publicly known. If Alice and Bob wish to create a shared key, they can proceed as follows:

(i) Alice selects uniformly at random an integer $a \in (2\ d - 1)$, computes g^a, and sends it to Bob.

(ii) Bob selects uniformly at random an integer $b \in (2\ d - 1)$, computes g^b, and sends it to Alice.

(iii) Alice computes $k_a = (g^b)^a$, while Bob computes $k_b = (g^a)^b$.

(iv) The shared key is thus $k = k_a = k_b \in G$.

The security of the scheme relies on the assumption that, knowing $g \in G$ and having observed both g^a and g^b, it is computationally infeasible for an adversary to obtain the shared key. This is known as the Diffie–Hellman Problem (DHP). The Diffie–Hellman problem is related to a better-known problem, the Discrete Logarithm Problem:

Discrete Logarithm Problem (DLP). Let G be a cyclic group, and g a generator of G. Given $h \in G$, find an integer t such that $g^t = h$. Clearly, if the DLP is easy then so is the DHP and thus the Diffie–Hellman key agreement protocol is insecure. So, as a minimum requirement interested in finding difficult instances of the DLP. Difficulty of the DLP depends heavily on the way the group G is represented not just on the isomorphism class of G. For example, the DLP is trivial if G = Z/dZ is the additive group generated by $g = 1$. However, if G is an appropriately chosen group of large size, the DLP is considered computationally infeasible

B. Ko–Lee–Cheon–Han–Kang–Park Key Agreement Protocol

Let G be a non-abelian group. For $g, x \in G$ then write g^x for $x^{-1}gx$, the conjugate of g by x. The notation suggests that conjugation might be used instead of exponentiation in cryptographic contexts. So, we can define an analogue to the discrete logarithm problem:

Conjugacy Search Problem. Let G be a non-abelian group. Let g, $h \in G$ be such that $h = h = g^x$ for some $x \in G$. Given the elements g and h, find an element $y \in G$ such that $h = g^y$. If we can find a group where the conjugacy search problem is hard (and assuming the elements of this group are easy to store and manipulate), one can define cryptosystems that are analogues of cryptosystems based on the discrete logarithm problem. Ko et al. proposed the following analogue of the Diffie–Hellman key agreement protocol.

As cited by (Ki, 2000). Let G be a non-abelian group, and let g be a publicly known element of G. Let A, B be

commuting subgroups of G, so (a, b) = 1 for all a∈A, b∈B. If Alice and Bob wish to create a common secret key, they can proceed as follows:

(i). Alice selects at random an element a∈A, computes $g_a = a^{-1}ga$, and sends it to Bob.

(ii). Bob selects at random an element b∈B, computes $g_b = b^{-1}gb$ and sends it to Alice.

(iii). Alice computes $k^a = (g^b)^a$, while Bob computes $k_b = (g^a)^b$.

(iv). Since ab = ba, we have $k_a = k_b$, as group elements (though their representations might be different). For many groups, we can use k_a and k_b to compute a secret key.

For example, if G has an efficient algorithm to compute a normal form for a group element, the secret key k could be the normal form of k_a and k_b. The interest in the paper of Ko et al. (Ki, 2000) centred on their proposal for a concrete candidate for G and the subgroups A and B, as follows. Take G to be the braid group B^n on n strings ((Emil, 1947), for example) which has presentation

$$B^n = \left\langle \sigma_1, \sigma_2, \ldots, \sigma_{n-1} \;\middle|\; \begin{array}{l} \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ for } |i-j| = 1 \\ \sigma_i\sigma_j = \sigma_j\sigma_i \text{ for } |i-j| \geq 2 \end{array} \right\rangle$$

Let l and r be integers such that l + r = n. Then we take $A = \langle \sigma_1, \sigma_2, \ldots, \sigma_{n-1} \rangle$ and $B = \langle \sigma_{l+1}, \sigma_{l+2}, \ldots, \sigma_{l+r-1} \rangle$.

The braid group is an attractive choice for the underlying group (a so called 'platform group') in the Ko et al. key agreement protocol: there is an efficient normal form for an element; group multiplication and inversion can be carried out efficiently; the conjugacy problem looks hard for braid groups. Note that we have not specified the cryptosystem precisely. Of course, chosen the values of n, l and r have not being chosen. But have also not specified how to choose the element g ∈G (it emerges that this choice is critical). Finally, since the subgroups A and B are infinite, it is not obvious how the elements a∈A and b∈B should be chosen.

C. Anshel–Anshel–Goldfeld Key Agreement Protocol

This beautiful key agreement protocol, due to Anshel, Anshel and Goldfeld has an advantage over the Ko et al. protocol: commuting subgroups A and B are not needed, Iris, (1999), According to (Iris,1999). Let G be a non-abelian group, and let elements $a_1, \ldots, a_k, b_1, \ldots, b_m \in G$ be public.

(i) Alice picks a private word x in $a_1, \ldots, a_k$ and sends $b_1^x, \ldots, b_m^x$ to Bob.

(ii) Bob picks a private word y in $b_1, \ldots, b_m$ and sends $a_1^y, \ldots, a_m^y$ k to Alice.

(iii) Alice computes x^y and Bob computes y^x.

(iv) The secret key is $(x, y) = x^{-1}y^{-1}xy$.

Note that Alice and Bob can both compute the secret commutator: Alice can premultiply x^y by x^{-1} and Bob can premultiply y^x by y^{-1} and then compute the inverse: $(x, y) = (y^{-1}yx)^{-1}$.

The Anshel et al. protocol is far from well specified as it stands. In particular, nothing has being said about the choice of platform group G. Like Ko et al., Anshel et al. proposed using braid groups because of the existence of efficient normal forms for group elements and because the conjugacy search problem seems hard. See (Alexei, 2008) for a discussion of some of the properties a platform group should have; they discuss the possibilities of using the following groups as platform groups: Thompson's group F, matrix groups, small cancellation groups, solvable groups, Artin groups and Grigor chuck's group.

D. The Stickel Key Agreement Protocol

Replacing conjugation: The Ko et al. scheme used conjugation in place of exponentiation in the Diffie–Hellman protocol, but there are many other alternatives. For example, one can define $g^a = \phi(b)ga$ and $g^b = \phi'(b)gb$ for any fixed functions $\phi : A \to A$ and $\phi' : B \to B$ (including the identity maps) and the scheme would work just as well. More generally, one may replace a and $\phi(c)$ by unrelated elements from A: there are protocols based on the difficulty of the decomposition problem, namely the problem of finding $a_1, a_2 \in A$ such that $h = a_1 g a_2$ where g and h are known. See Myasnikov et al. (Alexei, 2008) for a discussion of these and similar protocols; one proposal we find especially interesting is the Algebraic Eraser. As an example of such a protocol, will be briefly describe as a scheme due to Stickel, (Arkdius, 2006).

As cited in (Eberhard, 2005). Let G = GL(n, F_q), and let g ∈G. Let a, b be elements of G of order n_a and n_b respectively and suppose that ab ≠ ba. The group G and the elements a,b are publicly known. If Alice and Bob wish to create a shared key, they can proceed as follows:

(i) Alice chooses integers l, m uniformly at random, where $0 < l < n_a$ and $0 < m < n_b$. She sends $u = a^l g b^m$ to Bob.

(ii) Bob chooses integers r, s uniformly at random, where $0 < r < n_a$ and $0 < s < n_b$. He sends $v = a^r g b^s$ to Alice.

(iii) Alice computes $k_a = a^l v b^m = a^{l+r} g b^{m+s}$. Bob computes

$k_b = a^r u b^s = a^{l+r} g b^{m+s}$.

(iv) The shared key is thus $k = k_a = k_b$.

E. Secure Signatures with RSA

One issue of data transmission is the ability to verify a person's identity. If one sends a request to a bank to transfer money out of an account, the bank wants to know if the person who sends the request is really the owner of the account. If the request is made over the internet, how can the bank check the owner's identity? The RSA encryption system gives a method for checking identities, which is one of the important features of the system.

As cited, in David R. (2007). Suppose that person A transmits data to person B, and that person B wants a method

to check the identity of person A. To do this, both person A and B get sets of RSA data; person A has a modulus n_A and an encryption exponent e_A. These are publicly available. That person also has a decryption exponent d_A that remains private. Person B similarly has data n_B, e_B, and d_B. In addition, person A has a signature, a publicly available number S. To convince person B of his identity, person A first calculates $T = S^{e_A} mod n_A$ and then $R = T^{e_B} mod n_B$. He then transmits R to person B. Person B then decrypts R with her data, recovering $T = R^{d_B} mod n_B$. Finally, she encrypts T with person A's data, obtaining $T^{e_A} mod n_A$. By seeing that this result is the signature of person A, the identity has been validated. For example, suppose that the data for person A is n_A = 2673157 e_A= 23 d_A= 2437607 S = 837361 and the data for person B is n_B= 721864639 e_B= 19823 d_B= 700322447 Person A then calculates $837361^{2437607}$ mod2673157 = 1216606; and then 1216606^{19823} mod721864639 = 241279367: Person A then transmits 241279367 to person B. When person B receives this, she calculates $241279367^{700322447}$ mod721864639 = 1216606; and finally recovers S as $S = 1216606^{23}$ mod2673157.

To explain why this works, let denote by $encrypt_A(M)$ and $decrypt_A(M)$ the integers M^{eA} modn_A and M^{dA} modn_A, respectively. Similarly, one would have $encrypt_B(M)$ and $decrypt_B(M)$.

The validity of the RSA system says that $decrypt_A(encrypt_A(M)) = M$; $encrypt_A(decrypt_A(M)) = M$ Similar equations hold for B. With this notation, person A calculates

$R = encrypt_B(decrypt_A(S))$ and then person B calculates $encrypt_A(decrypt_B(R))$: Therefore, person B will calculate $encrypt_A(decrypt_B(encrypt_B(decrypt_A(S))))$ = $encrypt_A(decrypt_A(S)) = S$ as the consequence of the two immediate equations above. Therefore, person B does recover the signature of person A. The reason that this method validates the identity of person A is because only person A can calculate $decrypt_A(S)$. If another person tries to claim he is person A, tries to substitute a number F in place of $decrypt_A(S)$, he will transmit $encrypt_B(F)$ to person B. Person B will then calculate $encrypt_A(decrypt_B(encrypt_B(F))) = encrypt_A(F)$: However, in order to have $encrypt_A(F) = S$, we must have $decrypt_A(S) = decrypt_A(encrypt_A(F)) = F$; Which means that this person must have the correct decrypted number $decrypt_A(S)$; he cannot send any other number without person B realizing it is a fake number.

IV. CONCLUSION

If RSA algorithm is implemented correctly then the algorithm is effective and useful in cryptography. But there are some other issues also related to fear of security of RSA algorithm used in Cryptography. A well known attack on RSA is discovered by Paul Kocher. He demonstrates that it is possible to discover the decryption exponent by carefully timing the computation times for a series of decryptions. Efforts can be made to use finite group based cryptography. Although finite group based cryptography have many difficulties during implementation, but it has the more advantageous than the infinite group cryptography.

ACKNOWLEDGMENT

The authors of this paper wish to acknowledge the contribution of each other to the success of this work and wish also to acknowledge the work of typing done by the corresponding author.

REFERENCES

[1] Laurence C. Washington, Ellptic Curves. (second, Ed.) CRC Press, Boca, 2008.

[2] Samuel Hwere Tsok and Patrick N. Okolo, Application of Group Theory to a Local Game Called "Tsorry Checkerboard" (A Case of Klein Four- Group). IOSR-JM, Volume 7, Issue 3, 04-06, 2013. www.iosrjournals.org

[3] Whitefield Diffie; Martin Hellman. "New directions in cryptography". Information Theory, IEEE Transactions 22, 644-654, 1976

[4] Alfred J. Menezes and Scott A. Vanstone, A., A note on cyclic groups finite fields and the discrete logarithm problem, Application Algebra in Engineering, communication and computing, 3, 67-74, 1992

[5] Norman Biggs, The critical group from a cryptographic perspective, Bull. London Math. Soc. 39, 829–836, 2007

[6] Simon R. Blackburn, Carlos CID and Clarian Mullan. Group Theory in Cryptography a paper presented to the Department of Mathematics, Royal Holloway, University of London Egham, Surrey TW20 OEX, United Kingdom, 2010 {s. blackburn, carlos.cid, c.mullan}@rhul.ac.uk

[7] Steven Galbraith and Alfred Menezes, Algebraic curves and Cryptography, Finite Fields and Application 11, 544-577, 2005.

[8] Neal R. Wagner and Marianne R. Magyarik, A public key cryptosystem Based on the word problem, in Advances in Cryptology- CRYPTO' 84 (G.R. Blakely and David Chaum, eds), 19-36, 1985

[9] Mar´ıa Isabel Gonzalez Vasco, and Rainer Steinwandt, A reaction attack on a public key cryptosystem based on the word problem. Applicable Algebra In Engineering, Communication and computing 14, 335-340, 2004 .

[10] Ki, H. Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo H. Ju-Sung Kang and Choonsik Park, New public-key cryptosystem using braid group, in advances in Cryptology-CRYPTO 2000 (M. Bellare, ed.). Lecture notes in computer 1880, 166-183, 2000.

[11] Iris Anshel, Micheal Anshel, Dorian Goldfeld, An algebraic method for public-key cryptography. Math. And Lightweight cryptography. Contemp. Math 418, 1-34, 1999

[12] Alexei M., Shipilrain, V., Ushakov, A., Group-Based Cryptography Birkhauser, 2008,

[13] Arkdius Kalka, Mina Teicher and Boaz Tsaban, Cryptanalysis of the Algebra Eraser and short expressions of permutations as products' preprint. http://arxiv.org/abs/0804.0629., 2006.

[14] Eberhard S, A New method for Exchanging secret keys. Third international conference on Information Technology and Application (ICITA'05) pp 426-430. Piscataway: IEEE, Computer Society, 2005.

[15] David R. Finson and Patrick J. Morandi, An Introduction to Abstract Algebra Via Application, Department of Mathematical Sciences New Mexico State University Las Cruces NM 88003-8001, 2007.