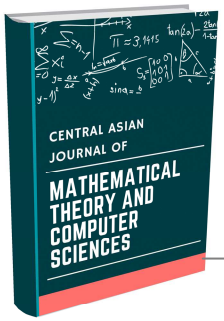

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

http://cajmtcs.centralasianstudies.org/index.php/CAJMTCS

Volume: 01 Issue: 01 | Sep 2020

Some Aspects of Group-Based Cryptography

Prof. (Dr.) Deepak Raj

Professor, P.I.E.T. Samalkha (Panipat), Haryana, India

ABSTRACT:

Over the past few years, many papers have proposed cryptosystems based primarily on institutional principles. Institutional-based cryptosystems now have not brought strong opposition to RSA and Diffie-Hellman, but the ideas are interesting and a different attitude concludes with a helpful staff perspective. Cryptographic texts are full size and varied, and it is very difficult for a young person to find the right information resources. How group theory can be applied to encryption is explained in this paper. The main reason for encryption is to make sure that the gadget designed for communication is secure. Device security depends on how well the set of rules is based. In this paper we discuss strategies based entirely on organizational theory.

ARTICLEINFO

Article history:
Received 20 July 2020
Revised form 22 August 2020
Accepted 28 September 2020

Key words: Non-abelian Group; Cryptosystem; Public Key; Private Key.

INTRODUCTION

More recently there have been a number of cryptographic protocols based on group theory concepts. Such agreements have not yet led to effective plans to compete with the likes of RSA and Diffie-Hellman, but the ideas are interesting and lead to a beneficial group theory.

Recognizing that quantum computers can successfully solve both Comprehensive Integration Problems and standard Discrete Logarithm Problems, searching for other cryptosystems has become increasingly important. Cryptosystems, which include group-based models that are not at risk for quantum enemies have become known as post-quantum cryptosystems. A well-known example is the McEliece cryptosystem based on the difficulty of recording error correction codes. Other examples include lattice-based cryptosystems and cryptosystems based on large systems of multivariate polynomial equations.

This paper deals with group-based cryptography: the design and analysis of cryptographic systems based on non-sized groups. Specifically, we explore the role of matrix groups as a platform for group-based cryptosystems. If one is going to use a team-based cryptosystem, you need an effective way to represent, maintain and decipher the team features. For these reasons matrix teams are an attractive source for non-nonabelian groups: matrices are easy to maintain and are not computer-generated, and line algebra provides an effective tool for deceptive elements. But linear algebra is also a powerful tool for the cryptanalyst. Indeed, the focus of this thesis is on cryptanalysis. We show the insecurity of a few group-based cryptosystems that use matrix groups as a platform, making them unsuitable for use in the real world. We manage a few setup protocols, a public key cryptosystem, and a group-based hash function. Outside of the hash function (where we will deal with collision resistance) we are working on an idle enemy model. So we

assume that the enemy has the power of the only listener who knows everything about the system used except the secret keys and random selections made by individual groups.

CRYPTOGRAPHY USING GROUPS

The first proposal to use non-nonabelian groups in the secret code public record was made by Wagner and Magyarik in 1985. The cryptosystem is based on the complexity of the Word Problem (or more precisely the Word Choice Problem) in the randomly defined groups. However, the program is a theory with a few unresolved issues: criticism is given to Gonzalez Vasco and Steinwandt [39] and Levy-dit- ´ Vehel and Perret. The significance of Wagner and Magyarik's plan lies in its innovation, which initiated the interaction between cryptography and the theory of a cohesive group. Let G be the group given the limited presentation. In 1911, Max Dehn posed the following problems:

- The Word Problem: given a word won the generators of G, decide if w = 1 in G.
- The Conjugacy Problem: given words u, v on the generators of G, decide if u and v represent conjugate elements in G.
- The Isomorphism Problem: given two finite presentations, decide if they present isomorphic groups.

KEY AGREEMENT PROTOCOL BY DIFFIE–HELLMAN

Assuming that G be a cyclic group, and g a generator of G, where both g and its order d are publicly known. If A and B wishes to create a shared key, they can proceed as follows:

1. On uniform basis A chooses at random an integer a ϵ [2, d − 1], computes g^a, and transfer it to B.
2. Again, on uniform basis B chooses at random an integer b ϵ [2, d − 1], computes g^b and transfer it to A.
3. A computes ka = $(g^b)^a$ while B computes kb = $(g^a)^b$
4. The shared key is thus k = ka = kb ∈ G.

In this Scheme security depends on the assumption that, knowing g ∈ G and having observed both g^a and g^b it is computationally infeasible for an outsider 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)

To describe the Discrete Logarithm Problem (DLP) let G be a cyclic group, and g a generator of G. Given h ϵ G, find an integer t such that g^t = h. It is clear that if DLP is easy then DHP is also easy. Hence, the Diffie–Hellman key agreement protocol is insecure. So if we are using DLP, we have to take care for use of difficult instances of it to make the system more secure. It is clear that difficulty level of the DLP depends heavily on the way the group G is represented, not just on the isomorphism class of G. The security of the system depends on the keys. So it is useful to choose groups of larger order. For example, the DLP is trivial if G = Z/dZ is the additive group generated by g = 1. Turning from the Diffie–Hellman scheme to the more general model, Here the main two points which are required to be emphasized are as follows:

1. A AND B ARE COMPUTERS

A protocol is needed to create a system that will show us how to use it. In particular, a well-defined plan should explain how group components are stored and used; the program description must contain an algorithm to generate any system width parameters; should also state how random selection is made. (Here the last point is very important when we choose features from the endless set, as a free team!) In spite of all these things, the rule should work well; the calculation time required to apply the rule is very important, but also the number of bits that need to be exchanged between A and B; number of domains (information exchange) required for a protocol; key size; system boundary sizes.

2. CRITICAL ASPECT REGARDING SECURITY

We use cryptosystems for public keys where the attacker knows the system i.e. Attacker is best known for how the system works without the secret keys and random selections made by individual groups.

But modern security often requires much more. For example, in the commonly readable IND-CCA2 model, we require that the attacker be able to guess (with a greater chance of success than 0.5) which of the two

messages is encrypted, when presented with a cipher text with one challenge and Encryption in one of the messages. This should also hold when the attacker is unable to select two messages, and is allowed to request coding of any cipher text that does not match the cipher text challenge. Note that cryptographer writers often have an interest in complexity in a general story (in other words, which happens often). Safety measures in extreme cases may not be as helpful in operation, as the worst case scenario may be very rare; even moderate case rates can be unnecessarily distorted by rare but complex cases.

From the above facts presented, we can now easily say that modern cryptography is much broader than the traditional forms of communication of the two groups we have discussed here: there is a developing society that develops the theory of multi-group communication, using positive concepts such as zero knowledge.

GROUPS BASSED CRYPTOGRAPHY

By using Group theory we can construct variants of the Diffie–Hellman key agreement protocol. Since the 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. In this context, various authors have suggested using a cyclic subgroup of a matrix group, 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; Biggs has proposed representing an abelian group as a critical group of a finite graph; but Blackburn has shown that this proposal is insecure. A different 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 hyperelliptic curves.

All the proposals discussed above use representations of abelian (indeed,cyclic) groups. What about non-abelian groups? The first proposal to use non abelian groups that we are aware of is due to Wagner and Magyarik in 1985. (See Gonz´alez Vasco and Steinwandt for an attack on this proposal; see Levy-dit-Vehel and Perret for more recent related work.) But interest in the field increased with two high-profile proposals approximately ten years ago.

CONJUGACY AND EXPONENTIATION

To explain the concept related to Conjugacy and exponentiation let G a non-abelian group is G. For g, x ϵ G we write g^x for x^{-1}gx, the conjugate of g by x. This 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

Another aspect with regard to Conjugacy Search problem is that let G be a non-abelian group. Let g, h ϵ G be such that h = g^x for some x ϵ G. Given the elements g and h, find an element y ϵ G such that h = g^y

Assuming that 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 similar to cryptosystems based on the discrete logarithm problem. Ko et al. proposed the following analogue of the Diffie–Hellman key agreement protocol.

KEY AGREEMENT PROTOCOL BY KO–LEE–CHEON–HAN–KANG–PARK

To discuss the **KAP** 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 ϵ A, b ϵ B.

If A and B wish to create a common secret key, they can proceed as follows:

1. A chooses at random an element a ϵ A, computes $g^a = a^{-1}$ga, and sends it to B.
2. B chooses at random an element b ϵ B, computes $g^b = b^{-1}$gb, and sends it to A
3. A computes ka = $(g^b)^a$ while B computes kb = $(g^a)^b$

4. Since ab = ba, hence we have ka = kb, as group elements. For many groups, we can use ka and kb 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 ka and kb.

The interest in the paper of Ko et al. centred on their proposal for a concrete candidate for G and the subgroups A and B, as follows. We take G to be the braid group Bn on n strings which has presentation

$Bn = \langle \sigma_{1,}\sigma_{2,}.....,\sigma_{n-1} \mid \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ for } |i-j| = 1,\ \sigma_i\sigma_j = \sigma_j\sigma_i \text{ for } |i-j| \geq 2 \rangle$

Let l and r be integers such that l+r = n. Then we take

$A = \langle \sigma_{1,}\sigma_{2,}.....,\sigma_{l-1} \rangle$ and

$B = \langle \sigma_{l+1,}\sigma_{l+2,}.....,\sigma_{l+r-1} \rangle$

Here the braid group is an attractive choice for the underlying group (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, we have not chosen the values of n, l and r. But we have also not specified how to choose the element g ϵ 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 ϵ A and b ϵ B should be chosen.

COMPUTING A COMMON COMMUTATOR

We will discuss another protocol due to Anshel, Anshel and Goldfeld has an advantage over the Ko et al. protocol: commuting subgroups A and B are not needed.

Anshel–Anshel–Goldfeld Key Agreement Protocol [1]. Let G be a non-abelian group, and let elements a1.. . ak, b1,. . ., bm ϵ G be public.

1. A select a private word x in a1,. . ., ak and sends $b_1^{\,x}$,.....$b_m^{\,x}$ to B.
2. B select a private word y in b1,. . ., bm and sends $a_1^{\,y}$,.....$a_k^{\,y}$ to A.
3. A computes x^y and B computes y^x
4. Here the secret key is [x, y] = $x^{-1}y^{-1}$xy.

Note that A and B can both compute the secret commutator: A can premultiply x^y by x^{-1} and B can premultiply y^x by y^{-1} and then compute the inverse: [x, y] = $(y^{-1}y^x)^{-1}$.

The protocol by Anshel et al. is far from well specified as it stands. In particular, we have said nothing about our 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 Myasnikov et al. 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 Grigorchuck's group.

CONJUGATION REPLACEMENT

There are many other alternatives in place of the scheme by Ko et al. used conjugation in place of exponentiation in the Diffie–Hellman protocol. For example, we could define g^a = φ(a)ga and gb = φ' (b)gb for any fixed functions

φ: A → A and φ': B → B (including the identity maps) and the scheme would work just as well. More generally, we may replace a and φ (a) by unrelated elements from A: there are protocols based on the difficulty of the decomposition problem, namely problem of finding a_1, a_2 ∈ A such that h = a_1ga_2 where g and h are known. See Myasnikov et al. 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, we briefly describe a scheme due to Stickel.

KEY AGREEMENY PROTOCOL BY THE STICKEL

To discuss the Key Agreement Protocol by the Stickel 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 A and B wishes to create a shared key, they can proceed as follows:

1. A select integers l, m uniformly at random, where 0 < l < n_a and 0 < m < n_b. She sends u = $a^l gb^m$ to B.
2. B select integers r, s uniformly at random, where 0 < r < n_a and 0 <s < n_b. He sends v = $a^r gb^s$ to A.
3. A computes k_a= $a^l vb^m = a^{l+r} gb^{m+s}$. B computes $k_b = a^r ub^s = a^{l+r} gb^{m+s}$.
4. The shared key is thus k = k_a = k_b.

LOGARITHMIC SIGNATURES

There can be more generalised form of Diffie-hellman scheme to find a more direct generalisation of the DLP for groups that are not necessarily abelian.

Let G be a finite group, S ⊆ G a subset of G and s a positive integer. For all $1 \le i \le s$, let $A_i = [\alpha_{i1},. . ., \alpha_{iri}]$ be a finite sequence of elements of G of length $r_i > 1$, and let $\alpha = [A_1,. . ., A_s]$ be the ordered sequence of A_i. We say that α is a cover for S if any h ∈ S can be written as a product $h = h_1 \cdot \cdot \cdot h_s$, where $h_i = \alpha_{iki} \in A_i$. If such a decomposition is unique for every g ∈ S, then α is said to be a logarithmic signature for S. One natural way to construct a logarithmic signature for a group G is to take a subgroup chain

$1 = G_0 < G_1 < \cdot \cdot \cdot < G_n = G,$

and let A_i be a complete set of coset representatives for G_{i-1} in G_i. Then $\alpha = [A_1,. . ., A_n]$ is a logarithmic signature (a so called transversal logarith-mic signature) for G.

Given an element h ∈ S and a cover α of S, obtaining a factorization

$$h = \alpha_1 k_1 \cdot \cdot \cdot \alpha_s k_s \qquad (1)$$

associated with α could well be a hard problem in general. Indeed, in some situations the problem is a Discrete Logarithm Problem. For example, let G be generated by an element g of large order, and define $A_{i+1} = [1, g^{2i}]$. Let $S = \{g^a \mid 0 \le a \le 2^s\}$. Then the ith bit of the discrete logarithm of h ∈ S is equal to 1 if and only if $k_i = 2$ in the factorization .

Though there are connections with the DLP, logarithmic signatures can-not be directly used in discrete logarithm based protocols, as there is no analogue of exponentiation. They were first used by Magliveras to construct a symmetric cipher known as Permutation Group Mappings (PGM). The ideas behind PGM have inspired several public key cryptosystems based on logarithmic signatures. Qu and Vanstone [28] proposed a scheme (Finite Group Mappings, or FGM) based on transversal logarithmic signatures in elementary abelian 2-groups. Magliveras, Stinson and van Trung [62] developed two interesting schemes based on finite permutation groups, MST_1 and MST_2. More recently, a public key cryptosystem based on Suzuki 2-groups (known as MST_3) has been proposed by Lempken et al.

SYMMETRIC SCHEMES

In cryptography it has been seen that the group theory main use is in public key cryptosystems and key exchange schemes, but has also been used in symmetric cryptography. We have already specified the block cipher PGM. This cipher satisfies some nice algebraic and statistical properties (such as robustness, scalability and a large key space. However, fast implementation becomes an issue, making it a rather inefficient cipher compared with more traditional block ciphers.) This subsection contains two more examples of group theory being used in symmetric cryptography.

A block cipher such as DES or AES can be regarded as a set S of permutations on the set of all possible blocks, indexed by the key. The question as to whether S is in fact a group has an impact on the ci-pher's security in some situations: if the set was a group, then encrypting a message twice over using the cipher with different keys would be no more secure than a single encryption. Other properties of the group generated by S are also of interest cryptographically and attacks have been pro-posed against ciphers that

do not satisfy some of these properties (though good group theoretic properties are not sufficient to guarantee a strong cipher. We note however that computing the group generated by a block cipher is often very difficult. For instance, it is known that the group generated by the DES block cipher is a subgroup of the alternating group A_264 , with order greater than 2^{56} (and thus S for DES is not a group, however little more is known about its structure. Block ciphers themselves are often built as iterated constructions of simpler key-dependent permutations known as round functions, and one can study properties of the permutation groups generated by these round func-tions. It has been shown, for instance, that the round functions of both DES and AES block ciphers are even permutations; furthermore it can be shown that these generate the alternating group A_264 and A_2128, respectively.

Another area of symmetric cryptography is the Hash function design where groups have been used in an interesting way. Recall it has been clearly shown that a hash function H is a function from the set of finite binary strings to a fixed finite set X. H(x) can be easily computed for any fixed string x, but it should be computationally infeasible to find two strings x and x' such that $H(x) = H(x')$. In many cryptographic proctols, Hash functions are a vital component but their design is still not well understood. The most widely used example of a hash function is SHA-1 (where SHA stands for Secure Hash Algorithm). Security flaws have been found in SHA-1; the more recent SHA-2 family of hash functions are now recommended. Z´emor proposed using walks through Cayley graphs as a basis for hash functions; the most well-known concrete proposal from this idea is a hash function of Tillich and Z´emor . We think this hash function deserves further study, despite a recent (and very beautiful) cryptanalysis due to Grassl et al. : see Steinwandt et al. and the references there for comments on the security of this hash function, and see Tillich and Z´emor for some more recent literature.

CONCLUSION

From the above discussion, we hope that the reader is already convinced that the protocols of Ko et al. and of Anshel et al. are elegant ideas, just waiting for the right platform group. Now the question here arises is that can such a platform group be found? Here, we need a candidate group whose elements can be manipulated and stored efficiently, and an associated problem that is hard in the overwhelming majority of instances. There has been a great deal of attention on infinite groups (such as braid groups) that can be defined combinatorially, but we feel that finite groups deserve a much closer study; many difficulties disappear when we use finite groups. Note that groups with small linear representations are often problematic, as linear algebra can be used to attack such groups; groups with many normal subgroups (such as p-groups, for example) are often vulnerable to attacks based on reducing a problem to smaller quotients; groups with permutation

We would like to point out that group-based cryptography motivates some beautiful and natural questions for the pure group theorist. Most obviously, the cryptosystems above motivate problems in computational group theory, especially combinatorial group theory.

REFERENCES

1. Iris Anshel, Michael Anshel and Dorian Goldfeld, An algebraic method for public-key cryptography, Math. Res. Lett. 6 (1999), 287-291.
2. Iris Anshel, Michael Anshel, Dorian Goldfeld and Stephane Lemieux, Key agreement, the Algebraic Eraser^{TM}, and lightweight cryptography, Contemp. Math. 418 (2006) 1–34.
3. Norman Biggs, The critical group from a cryptographic perspective, Bull. London Math. Soc. 39 (2007) 829–836.
4. Simon R. Blackburn, Cryptanalysing the critical group Cryptology 8 (1995), 157–166.
5. D. Coppersmith, The Data Encryption Standard (DES) and its strength against attacks, IBM Research Report RC 18613 (IBM, 1992)
6. Mar´ıa Isabel Gonz´alez Vasco, Spyros Magliveras and Rainer Stein-wandt, Group-theoretic cryptography, Chapman & Hall / CRC Press, to appear.

7. Mar´ıa Isabel Gonzalez Vasco, Martin R¨otteler and Rainer Steinwandt, On minimal length factorizations of finite groups, J. Exp. Math. 12 (2003), 1–12.
8. Burton S. Kaliski Jr, Ronald L. Rivest, and Alan T. Sherman, Is the Data Encryption Standard a group? (Results of cycling experiments on DES), J. Cryptology 1 (1988), 3–36.
9. Arkadius Kalka, Mina Teicher and Boaz Tsaban, 'Cryptanalysis of the Algebraic Eraser and short expressions of permutations as products', preprint. See http://arxiv.org/abs/0804.0629.
10. Spyros S. Magliveras and Nasir D. Memon, The algebraic properties of cryptosystem PGM, J. Cryptology 5 (1992), 167–183.
11. S. S. Magliveras, D. R. Stinson and Tran van Trung, New approaches to designing public key cryptosystems using one-way functions and trap-doors in finite groups, J. Cryptology 15 (2002) 167–183.
12. Sean Murphy, Kenneth Paterson, and Peter Wild, A weak cipher that generates the symmetric group, J. Cryptology 7 (1994) 61–65.
13. Encryption Standard, Federal Information Processing Standards Publication (FIPS) 46, 1977.
14. National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication (FIPS) 180-1, 1995.
15. Douglas R. Stinson, Cryptography: Theory and Practice, Third Edition (Chapman & Hall, Boca Raton, 2005).
16. Lecture Notes in Computer Science, 839 (Springer, 1994), 40–49.