Group Theory

Applications of Non-Abelian Groups in Cryptography

Emma Corbett, Eoin McArdle & Gordon O'Connor

National University of Ireland, Galway

Introduction

Currently the majority of cryptographic schemes are based on commutative algebraic structures such as Abelian Groups. In terms of classical computing, these schemes are considered secure. This is because the problems which underpin their operation are considered 'hard' or intractable. This means that no solution can be found in reasonable time. For example it takes approx. $2.73 * 10^{61}$ years to crack AES-256 encryption using a home computer. However, recent advancements in quantum computing theory have shown that not all these problems are indeed intractable. The use of non-commutative algebraic structures such as non-abelian groups offer a possible solution to this security issue.

Abelian Groups in Cryptography

Abelian Platform Groups:

Many abelian groups can be used for cryptographic schemes. A simple example is the additive group $G = \mathbb{Z}/d\mathbb{Z}$. However, in practice much larger and complex groups are used. Groups of points on suitable elliptic curves are usually used. An elliptic curve is given by the equation $y^2 = x^3 + ax + b$.

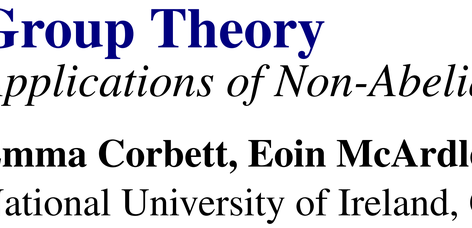

Figure 1: Examples of an Elliptic Curves

Discrete Logarithm Problem (DLP):

The DLP is one of the main problems that current crypography relies on. It is described as follows where G is a cyclic group with generator g:

$$\textit{Let } G = \langle g \rangle$$

$$\textit{Given } h \in G \textit{ find } x \textit{ such that } g^x = h$$

- Currently the DLP problem is intractable using current computing methods for certain large groups of G.
- However, Shor's quantum algorithm has been shown to solve this problem in polynomial time, therefore making the DLP tractable even for complex, large groups such as elliptic curves.
- *Fig. 2* shows the difference between tractable problems (polynomial, linear, logarithmic) and intractable problems (exponential) in terms of time.

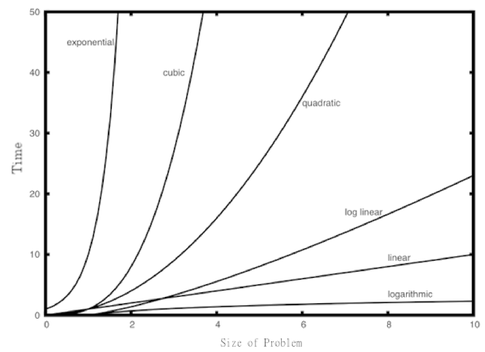

Figure 2: Time Complexity

Diffie-Hellman Key Exchange:

The Diffie-Hellman Key Exchange protocol is a fundamental method of establishing a secret shared key between two parties over an insecure connection. Suppose Alice and Bob wish to create a shared key, K using the cyclic group G where the order, d, and generator, g of G are publicly known:

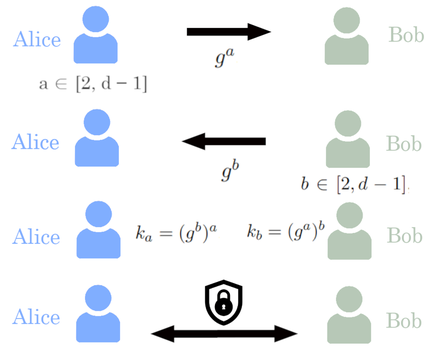

Figure 3: Diffie-Hellman Protocol

- From the above protocol we can clearly see that the Diffie Hellman Key Exchange relies on the DLP being intractable as do many other cryptographic protocols.
- This highlights the need for a more secure alternative for the quantum computing age.

Non-Abelian Platform Groups

The idea of using the complexity of non abelian (infinite) groups in cryptography dates back to the work of Wagner and Magyairk in 1985. A cryptographic platform group G must have several key requirements in order to tackle the conjugacy search problem:

- The group G must be well studied/understood.
- The word problem in G should have a linear/quadratic solution by a deterministic algorithm.
- There should be a way to disguise the elements of G so that they it is impossible to to recover individual elements from a product of elements just by inspection.
- G should be a group of super, polynomial growth. This insures that the number of elements in G of length n will grow faster than any polynomial in n.

Braid Groups:

Braid Groups were one of the first non-commutative groups to be suggested as a "good" suggestion as a cryptographic platform. There are many advantages and disadvantages of using braid groups in cryptography . It appears as if the conjugacy search problem in a braid group does not provide sufficient security unless keys are selected by narrow and yet to be determined subsets of the entire group.

A braid is obtained by laying down a number of parallel pieces if string and intertwining them without forgetting that they are essentially the same direction.

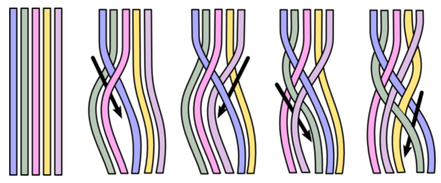

Figure 4: Five Strand Braid - Vertical direction

There are exactly $n-1$ crossing types for an n strand braid. $(x_1, ...x_{n-1}).x_i$ is a positive crossing of of the ith and i +1st strands. the set $x_1, ...x_{n-1}$ generates B_n.Each crossing is subject to the relation

$$[x_i, x_j] = 1$$

for every i, j s.t $|i-j| > 1$ and

$$x_i x_{i+1} x_i = x_{i+1} x_i x_{i+1}$$

That is , the braid group B_n is denoted :

$$B_n = \left\{ x_1, ...x_{n-1} \middle| \begin{array}{c} x_i x_j x_i = x_j x_i x_j \ \ if \ \ |i-j| = 1 \\ x_i x_j = x_j x_i \ \ if \ \ |i-j| > 1 \end{array} \right\}$$

Words and Normal Groups:

- A word is any written product of group elements and their inverses. For example if $x, y, z \in G$ then $xy,\ z^{-1}xzz,\ y^{-1}zxx^{-1}yz^{-1}$ are words in the set x,y,z.Two different words may evaluate to the same value in G
- A word is called reduced if it contains no string of the form $aa^{-1}, a \in$it's generating set
- Normal form for a free group G with generating set S is a choice of a reduced word in S for each element of G

Conjugacy Search Problem (CSP):

Let G be some platform group. For example:

$$G = B_n$$

Given words $x, y \in G$ find z such that:

$$y = zxz^{-1}$$

- It has been shown that for Braid groups and other suitable platform groups this problem is undecidable.
- This means that no algorithm can be designed such that leads to the conclusion whether z exists or not and therefore offer more security potential.

Non-Abelian Cryptographic Protocols

Many different protocols exist which rely on non-abelian platform groups to perform key exchanges, encryption/decryption and authentication. These protocols aim to provide greater security than their commutative equivalents.

Anshel–Anshel–Goldfeld Key Exchange:

- Is a non-abelian alternative to the Diffie-Hellman Key Exchange.
- Requires that the platform group used has easily computable normal forms. Because of this, braid groups are primarily used as the platform group for this protocol.

Let $G = B_n$

$a = (a_1, ..., a_k),\ b = (b_1, ..., b_m) \in G$

a and b are publicly known.

Alice selects a word in a and computes its product A:

$$A = a_{i_1}^{\epsilon 1} ... a_{i_L}^{\epsilon L},\ a_{i_k} \in a,\ \epsilon_k = \pm 1$$

Bob selects a word in b and computes its product B:

$$B = b_{j_1}^{\delta 1} ... b_{j_L}^{\delta L},\ b_{j_k} \in b,\ \delta_k = \pm 1$$

Alice sends Bob the conjugates:

$$Ab_1A^{-1}, ..., Ab_mA^{-1}$$

Bob sends Alice the conjugates:

$$Ba_1B^{-1}, ..., Ba_kB^{-1}$$

Alice computes:

$$A^{-1}(Ba_{i_1}^{\epsilon 1}B^{-1})...(Ba_{i_L}^{\epsilon L}B^{-1}) = A^{-1}B^{-1}AB$$

Bob computes:

$$(Ab_{i_1}^{\epsilon 1}A^{-1})...(Ab_{i_L}^{\epsilon L}A^{-1})B = A^{-1}B^{-1}AB$$

Conclusion

Non-abelian cryptography clearly shows promise in resisting quantum computing attacks due to the CSP being undecidable for suitable platform groups. However, with their added complexity and relative lack of research they are not often implemented at present. With this said, they are one of the main candidates for the future of cryptography in the quantum computing age.