ENCRYPTION METHODS USING FORMAL POWER SERIES RINGS

GILBERT BAUMSLAG, YEGOR BRUKHOV, BENJAMIN FINE AND GERHARD ROSENBERGER

Abstract. Recently there has been a great deal of work on noncommutative algebraic cryptography. This involves the use of noncommutative algebraic objects as the platforms for encryption systems. Most of this work, such as the Anshel-Anshel-Goldfeld scheme, the Ko-Lee scheme and the Baumslag-Fine-Xu Modular group scheme use nonabelian groups as the basic algebraic object. Some of these encryption methods have been successful and some have been broken. It has been suggested that at this point further pure group theoretic research, with an eye towards cryptographic applications, is necessary.In the present study we attempt to extend the class of noncommutative algebraic objects to be used in cryptography. In particular we explore several different methods to use a formal power series ring $R << x_1, ..., x_n >>$ in noncommuting variables $x_1, ..., x_n$ as a base to develop cryptosystems. Although R can be any ring we have in mind formal power series rings over the rationals $\mathbb{Q}$. We use in particular a result of Magnus that a finitely generated free group F has a faithful representation in a quotient of the formal power series ring in noncommuting variables.

1. Introduction

Most common public key cryptosystems and public key exchange protocols presently in use, such as the RSA algorithm, Diffie-Hellman, and elliptic curve methods are number theory based and hence depend on the structure of abelian groups. The strength of computing machinery has made these techniques theoretically susceptible to attack and hence recently there has been an active line of research to develop cryptosystems and key exchange protocols using noncommutative cryptographic platforms. This line of investigation has been given the broad title of **noncommutative algebraic cryptography**.

This paper was initiated while one of the authors (B.Fine) was a visitor at the CRM. We'd like to thank the CRM for its hospitality.

Up to this point the main sources for noncommutative cryptographic platforms has been nonabelian groups. In cryptosystems based on these objects algebraic properties of the platforms are used prominently in both devising cryptosystems and in cryptanalysis. In particular the nonsolvability of certain algorithmic problems in finitely presented groups, such as the conjugator search problem, has been crucial in encryption and decryption.

The important sources of nonabelian groups that can be used in cryptosystems are combinatorial group theory and linear group theory. Braid group cryptography, where encryption is done within the classical braid groups, is one prominent example. The one way functions in braid group systems are based on the difficulty of solving group theoretic decision problems such as the conjugacy problem. Although braid group cryptography had initial spectacular success, various potential attacks have been identified. Borovik, Myasnikov, Shpilrain [BMS] and others have studied the statistical aspects of these attacks and have identitifed what are termed black holes in the platform groups outside of which present cryptographic problems. In [BFX] and [X] potential cryptosystems using a combination of combinatorial group theory and linear groups were suggested and a general schema for the these types of cryptosystems was given. In [BFX 2] a public key version of this schema using the classical modular group as a platform was presented. A cryptosystem using the the extended modular group $SL_2(\mathbb{Z})$ was developed by Yamamura ([Y]) but was subsequently shown to have loopholes ([BG],[S],[HGS]). In [BFX 2] attacks based on these loopholes were closed.

It has been suggested that at this point further pure group theoretic research, and algebraic reserach in general, with an eye towards cryptographic applications, is necessary. In particular, although the present braid group cryptosystems may be attackable the basic group theoretic ideas are important. What is then necessary is to look at other (nonabelian) group theoretical methods as well as additional potential platform groups. Along these line in [BCFRX] an approach was followed based on a nonabelian group having either a large abelian subgroup or two large subgroups which elementwise commute. Using this idea a general key transport protocol modeled on the classical Diffie-Hellman technique but using a nonabelian group was developed. Several potential groups that could be used as platforms were described there, in particular the automorphism group of a free group.

In the present study we attempt to extend the class of noncommutative algebraic objects to be used in cryptography. In particular we explore several different methods to use a formal power series ring $R << x_1, ..., x_n >>$ in noncommuting variables $x_1, ..., x_n$ as a base to develop cryptosystems.

Although R can be any ring we have in mind formal power series rings over the rationals $\mathbb{Q}$. We use in particular a result of Magnus that a finitely generated free group F has a faithful representation in a quotient of the formal power series ring in noncommuting variables.

After describing the Magnus representation and some necessary properties we first show how to develop a Diffie-Hellman–RSA type of encryption system within the formal power series ring using the Magnus representation. In a different direction we describe the algorithm to rewrite an element of the free group in terms of the generators given its image in the power series ring. Using this we show how we can use the free groups within the formal power series ring as a platform group for the free group polyalphabetic cipher developed in [BFX]. The only previous examples were matrix groups over number rings. Under devlopment is a method to impose relations on the noncommutting variables to develop further types of secure encryptions.

2. Formal Power Series Rings and the Magnus Representation

Our encryption methods will use the ring of formal power series

$$R << x_1, ..., x_n >>$$

over a ring R in noncommuting variables $x_1, ..., x_n$. Although this can be done in an even more general context, for this study we will concentrate on rational formal power series, that is we consider the ring R to be the field of rational numbers $\mathbb{Q}$.

Throughout the rest of this paper we let

$$H = \mathbb{Q} << x_1, ..., x_n >>$$

be the formal power series ring in noncommuting variables $x_1, ..., x_n$ over $\mathbb{Q}$. One of our primary tools for developing encryption methods will be based upon a faithful representation of a finitely generated free group within a quotient of H. This representation was developed was introduced by W.Magnus [M] and is now known as the **Magnus representation**. If $n \geq 2$ this then provides free subgroups of all countable ranks within this quotient of H. Further by imposing additional relations we can obtain representations of free nilpotent groups.

We first describe the Magnus representation and give a proof. The proof will lead us to two algorithms for describing when certain polynomials lie in the image of this representation. Further we describe the unit group of this quotient.

First let $d > 1$ be an integer and impose the relations

$$x_1^d = x_2^d = ... = x_n^d = 0$$

on H. We call the resulting quotient $\overline{H}$.

Notice that the elements of $\overline{H}$ are polynomials of degree $< d$ in the noncommuting variables $x_1, ..., x_n$. The faithful representation of a free group is given in terms of the monomials

$$\alpha_1 = 1 + x_1, \alpha_2 = 1 + x_2, ..., \alpha_n = 1 + x_n.$$

Notice that in the formal power series ring H we have the well known expansion

$$\frac{1}{1+x_i} = 1 - x_i + x_i^2 - x_i^3 + ...$$

Therefore each α_i is invertible within H and hence invertible in $\overline{H}$. Within $\overline{H}$ however the inverse is a polynomial of degree $< d$ and so within $\overline{H}$

$$\frac{1}{1+x_i} = 1 - x_i + x_i^2 - x_i^3 + ... + (-1)^{d-1} x_i^{d-1}.$$

Therefore each α_i is in the unit group $U(\overline{H})$ of $\overline{H}$ and therefore the set $\{\alpha_1, ..., \alpha_n\}$ generates a multiplicative subgroup of $U(\overline{H})$. Note also that if d, the defining power, is kept secret, then inverses are unknown.

Magnus's result is the following.

Theorem 2.1. *The elements*

$$\alpha_1 = 1 + x_1, ..., \alpha_n = 1 + x_n$$

freely generate a subgroup of $U(\overline{H})$. Therefore the map given by

$$y_1 \to \alpha_1, ..., y_n \to \alpha_n$$

provides a faithful representation of the free group on $y_1, ..., y_n$ into $\overline{H}$.

We present a proof, since as mentioned the proof will lead us to an algorithm necessary for our encruption methods.

Proof. Notice from the comment above that each α_i is invertible within $\overline{H}$. Therefore each α_i is in the unit group $U(\overline{H})$ of $\overline{H}$ and therefore the set $\{\alpha_1, ..., \alpha_n\}$ generates a multiplicative subgroup of $U(\overline{H})$. We show that no nontrivial freely reduced word in the α_i can be the identity and hence the group they generate must be a free group.

From the binomial expansion we have for any non-zero integer n, positive or negative,

$$(1 + \alpha_i)^n = 1 + n\alpha_i + \text{ terms in higher powers }.$$

Now let

$$W(\alpha_1, ..., \alpha_n) = \alpha_{i_1}^{n_1} \alpha_{i_2}^{n_2} \dots \alpha_{i_k}^{n_k}$$

be a freely reduced word in the α_i with each $|n_i| \geq 1$ and $\alpha_{i_j} \neq \alpha_{i_{j+1}}$ for $j = 1, ..., k-1$. For later reference we call k the **block length**. In the ring $\overline{H}$ we then have

$$W(\alpha_1, ..., \alpha_n) = (1 + x_{i_1})^{n_1} ....(1 + x_{i_k})^{n_k}$$

and hence

$W(\alpha_1, ..., \alpha_n) =$

$= (1 + n_1 x_{i_1} + \text{higher powers in } x_{i_1}) \ldots (1 + n_k x_{i_k} + \text{higher powers in } x_{i_k})$

The variables are noncommuting, so that in analyzing this product we see that there is a unique monomial term of maximal block length k where each x_{i_j} appears to the power 1. That is there is a unique monomial term

$$n_1 n_2 ... n_k (x_{i_1} x_{i_2} ... x_{i_k}).$$

We stress here that this is of maximal block length since this will be important in the subsequent algorithm.

Since each $n_i \neq 0$ this term must appear and therefore $W(\alpha_1, ..., \alpha_n) \neq 1$. It follows that the group generated by $\alpha_1, .., \alpha_n$ is freely generated by them. □

The proof of the faithfulness of the Magnus representation leads us to several algorithms for dealing with the image in the power series ring. We will employ these algorithms in our cryptosystems. For the remainder of this section we will let $\overline{F}$ denote the free subgroup of $\overline{H}$ generated by the α_i.

The first algorithm provides a method, given a polynomial in $\overline{H}$, which is written in polynomial form, that we know to be in $\overline{F}$, to write its unique free group decomposition. That is given

$$f = f(x_1, ..., x_n)$$

a polynomial in the noncommuting variables $x_1, .., x_n$ that we know to be in $\overline{F}$ to rewrite f as

$$f = W(\alpha_1, ..., \alpha_n).$$

In general there is no factoring algorithm in $\overline{H}$.

For any monomial $x_{i_1} \cdots x_{i_k}$ in $\overline{H}$ we call k the block length of the monomial in analogy with that of a free group word.

Theorem 2.2. *(Algorithm to Recover the Free Group Decomposition of Elements in $\overline{F}$). Suppose $f = f(x_1, ..., x_n) \in \overline{H}$ and it is known that $f \in \overline{F}$. There is an algorithm that rewrites f in terms of the free generators $\alpha_1, ..., \alpha_n$, that is the algorithm uniquely expresses f as a free group word*

$$f = W(\alpha_1, ..., \alpha_n).$$

The algorithm works as follows:

Step 1: In f locate the monomial $nx_{i_1}\cdots x_{i_k}$ of maximal block length where $n \in \mathbb{Z}\setminus\{0\}$, each variable that appears in f appears in this monomial, and each variable is to the power 1. This k gives the block length for the corresponding free group word. Further the free group word must have the form

$$\alpha_{i_1}^{n_1}\cdots\alpha_{i_k}^{n_k}$$

with each n_i a divisor of n.

Step 2: For each divisor n_i of n both positive and negative sequentially form $(1+x_{i_1})^{-n_1}f$. In exactly one such product the maximal block length will be $k-1$ and there will be a unique monomial of block length $k-1$ containing each variable in f except perhaps x_{i_1} and each to the power 1. We then have

$$f = (1+x_{i_1})^{n_1} f_1$$

where f_1 is also in $\overline{F}$.

Step 3: Continue in this manner until we reach the identity. The free group decomposition of f is then

$$f = (1+x_{i_1})^{n_1}\cdots(1+x_{i_k})^{n_k} = \alpha_{i_1}^{n_1}\cdots\alpha_{i_k}^{n_k}.$$

Proof. Since we know that $f \in \overline{F}$ we know that there is a unique free group decomposition

$$f = \alpha_{i_1}^{n_1}\cdots\alpha_{i_k}^{n_k} = (1+x_{i_1})^{n_1}\cdots(1+x_{i_k})^{n_k}.$$

Hence, as in the proof that the representation is faithful, there is a unique monomial $nx_{i_1}\cdots x_{i_k}$ of maximal block length where $n \in \mathbb{Z}\setminus\{0\}$, each variable that appears in f appears in this monomial, and each variable is to the power 1. Again as in the proof of Theorem 2.2, k gives the block length for the corresponding free group word.

Now, since the free group representation is unique we have for each divisor n_i of n

$$(1+x_{i_1})^{-n_i}f = (1+x_{i_1})^{n_1-n_i}\cdots(1+x_{i_k})^{n_k}.$$

Hence only for $n_i = n_1$ will this term cancel. Hence there is exactly one such divisor such that $(1+x_i)^{-n_i}F$ will now have maximal block length $k-1$ and have a unqiue monomial of the prescribed type. It follows then that one and only one such product will reduce f to a word of shorter block length.

□

A modification of the above algorithm can be used to determine if a general element of $\overline{H}$ is actually in $\overline{F}$.

Theorem 2.3. *(Algorithm to Determine if $f \in \overline{H}$ is in $\overline{F}$). Suppose*

$$f = f(x_1, ..., x_n) \in \overline{H}.$$

There is an algorithm that determines whether or not $f \in \overline{F}$ and if it is, rewrites f in terms of the free generators $\alpha_1, ..., \alpha_n$. The algorithm works as follows:

Step 1: If the constant term of $f \neq 1$ then $f \notin \overline{F}$. Further if f has any nonintegral coefficients then $f \notin \overline{F}$.

Step 2: Assume f passes Step 1. If f does not contain a unique monomial $nx_{i_1} \cdots x_{i_k}$ of maximal block length in f where $n \in \mathbb{Z} \setminus \{0\}$, each variable that appears in f appears in this monomial, and each variable is to the power 1 then $f \notin \overline{F}$.

Step 3: Suppose f passes Steps 1 and 2. Then in f locate the monomial with the characteristics described in Step 2. If $f \in \overline{F}$ then k gives the block length for the corresponding free group word. Further the free group word must have the form

$$\alpha_{i_1}^{n_1} \cdots \alpha_{i_k}^{n_k}$$

with each n_i a divisor of n.

Step 4: For each divisor n_i of n both positive and negative sequentially form $(1 + x_{i_1})^{-n_1} f$. If in such product the maximal block becomes $k - 1$ and there is a no new monomial having the descrined characteristics above then $f \notin \overline{F}$. Otherwise continue.

Step 5: If evenutally we arrive at the identity then $f \in \overline{F}$ and the procedure yields the free product decomposition of f.

Proof. The proof follows in exactly the same manner as the proof of Theorem 2.2.

□

For certain cryptographic applications we need the full unit group $U(\overline{H})$ of $\overline{H}$. Over $\mathbb{Q}$ it can be described as those polynomials with nonzero constant term.

Theorem 2.4. *The unit group $U(\overline{H})$ over $\mathbb{Q}$ consists precisely of those polynomials with nonzero constant term.*

Proof. There are two ways to look at the proof of this. Algebraically, suppose that the defining power is $d > 1$ and $P(x) \in \overline{H}$ with nonzero constant term. Then $P(x)$ is relatively prime to the polynomials x_i^d and so is invertible in the factor ring in the standard way.

Analytically if $P(x) \in \overline{H}$ with nonzero constant term let $P^\star(x)$ be the correpsoning polynomial in H. Then $P^\star(x)$ can be made into part of a convergent power series $P^{\star\star}(x)$ in

$$\mathbb{C} << x_1, ..., x_n >> .$$

Since $P^{\star\star}(0) \neq 0$ this power series is analytic at 0 and so its inverse is analytic at) and so has a convergent power series around 0 say $Q^{\star\star}(x)$. The image of $Q^{\star\star}(x)$ in $\overline{H}$ would then be the inverse of $P(x)$.

Conversely if $P(x) \in \overline{H}$ is invertible it must have nonzero constant term. □

Before we continue we mention one final item concerning multiplication within $\overline{H}$. In general there is no factoring algorithm. However if $f \in \overline{H}$ is known and $g = fe$ with $e \in \overline{F}$ is known then we can find e. We say that e can be peeled off fe. The algorithm to do this is essentially the same as the above two algorithms. We briefly explain. Suppose we are given f and fe. Then in fe there is a unique monomial extending the monomials in f exactly as in the proof of theorem 2.2. By identifying this monomial we can find the free group decomposition of e and hence find e.

3. Cryptosystems Using the Formal Power Series Rings

We now provide several methods for developing cryptosystems using the formal power series rings and the above quotients -together with the Magnus representation. The first method is a further extension to the nonabelian group setting of the Diffie-Hellman system.

3.1. **A General Schema for Nonabelian Group Diffie-Hellman.** In [BCFRX] a group theoretic encryption protocol analogous to the standard Diffie-Hellman scheme and generalizing RSA was described in the following manner. Suppose that G is a finitely presented group that can be represented in a nice way - either as a matrix group or as words relative to a nice presentation. Further suppose that G has two large subgroups A_1,A_2 that commute elementwise. Alternatively we could use one large abelian subgroup A of G. The meaning of large is of course hazy but relative to the encryption scheme means that within G it is difficult to determine when an arbitrary element is in A_1 or A_2 (or A) and further A_1 and A_2 (or A) is large enough so that random choices can be made from them.

Now suppose that Bob wants to communicate with Alice via an open airway. The message (or the secret key telling them which encryption system to use) is encoded within the finitely generated group G with the properties given above. The two subgroups A_1, A_2 which commute elementwise are kept secret by Bob and Alice. A_1 is the subgroup for Bob and A_2 the subgroup for Alice. Bob wants to send the key $W \in G$ to Alice. He chooses two random elements $B_1, B_2 \in A_1$ and sends Alice the message

(in encrypted form) B_1WB_2. Alice now chooses two random elements $C_1, C_2 \in A_2$ and sends $C_1B_1WB_2C_2$ back to Bob. These messages appear in the representation of G and hence for example as matrices or as reduced words in the generators so they don't appear as solely concatenation of letters. Since A_1 commutes elementwise with A_2 we have

$$C_1B_1WB_2C_2 = B_1C_1WC_2B_2.$$

Further since Bob knows his chosen elements B_1 and B_2 he can multiply by their inverses to obtain C_1WC_2 which he then sends back to Alice. Since Alice knows her chosen elements C_1, C_2 she can multiply by their inverses to obtain the key W. It is assumed that for each message Bob and Alice would choose different pairs of random elements from either A_1 or A_2.

This method is a generalization of the Anshel, Anshel, Goldfeld and Ko-Lee schemes which used the classical Braid groups as platforms (see [BCFRX]). In [BCFRX] several additional potential platform groups are suggested.

We now present an alternative version of this method which can use the formal power series ring and its quotients as platforms. The general schema goes as follows:

We suppose that we have a ring R with a large unit group $U(R)$. By large we mean that $U(R)$ contains a nonabelian free subgroup so that random choices can be made from $U(R)$. We suppose further that there is no factoring algorithm in R. Suppose that Bob wants to send a message to Alice. Encoding is done within R so that elements of R represent messages. Bob wants to send the message $r \in R$ to Alice. He randomly chooses an $e \in U(R)$ and sends Alice re. Alice randomly chooses $f \in U(R)$ and sends back fre. Bob knows (but presumably an attacker can't figure out) e^{-1} so forms $free^{-1} = fr$ and sends this back to Alice. Alice applies f^{-1} to get the message r.

This method can be applied using the ring $\overline{H}$ as the platform. The power d defining $\overline{H}$ is a shared secret. Encryption is done in a polynomial in the noncommuting variables. This encryption can be done in a variety of ways. The simplest is perhaps the following. The coefficients of our polynomials are rational numbers. Codethe plaintext letters by rational numbers and then the message can be read off form the coefficients.

Bob wants to send Alice the message $T \in \mathbb{Q}[[x_1, .., x_n]]$ where $x_i^d = 0$ for all i. Let $R = T + S$ where S is an arbitrary polynomial with only powers higher than d. Bob chooses a random element of the unit group W. He sends Alice RW. Bob knows the inverse of W. Alice chooses another random V of the unit group and sends Bob back VRW. Bob multiplies by W^{-1} and sends Alice VR from which Alice recovers R. Since she knows d

she cancels all powers higher than d to obtain the message T. An attacker would need to factor RW and know the defining power d to attack the message. Notice that this scheme would not work if we restrcited W to be in the Magnus free group since if Bob sens RW and Alice sends back VRW then as explained V can be peeled off and subsequently W can be peeled off so the attacker can get the message R.

3.2. **Free Group Cryptosystems in Formal PowerSeries.**

In [BFX] the following general encryption scheme using free group cryptography was described.

We start with a finitely presented group

$$G =< X|R >$$

where $X = \{x_1, ...x_n\}$ and a faithful representation

$$\rho : G \to \overline{G}.$$

$\overline{G}$ can be any one of several different kinds of objects - linear group, permutation group, power series ring etc.

We assume that there is an algorithm to re-express an element of $\rho(G)$ in $\overline{G}$ in terms of the generators of G. That is is $g = W(x_1, ...x_n...) \in G$ where W is a word in the these generators and we are given $\rho(g) \in G$ we can algorithmically find g and its expression as the word $W(x_1, ..x_n)$.

Once we have G we assume that we have two free subgroups K, H with

$$H \subset K \subset G.$$

We assume that we have fixed Schreier transversals for K in G and for H in K both of which are held in secret by the communicating parties Bob and Alice (see [GB 1] for a description of Reidemeister-Schreier). Now based on the fixed Shreier transversals we have sets of Schreier generators constructed from the Reidemeister-Schreier process for K and for H.

$$k_1, ...k_m, ... \quad \text{for } K$$

and

$$h_1, ...h_t, ... \quad \text{for } H.$$

Notice that the generators for K will be given as words in $x_1, ...x_n$ the generators of G while the generators for H will be given as words in the generators $k_1, k_2, ...$ for K. We note further that H and K may coincide and that H and K need not in general be free but only have a unique set of normal forms so that the representation of an element in terms of the given Schreier generators is unique.

We will encode within H, or more precisely within $\rho(H)$. We assume that the number of generators for H is is larger than the set of characters within our plaintext alphabet. Let $\mathcal{A} = \{a, b, c...\}$ be our plaintext alphabet. At the simplest level we choose a starting point i, within the generators of H, and enclode

$$a \to h_i, b \to h_{i+1}, ... \text{ etc.}$$

Suppose that Bob wants to communicate the message $W(a, b, c...)$ to Alice where W ia a word in the plaintext alphabet. Recall that both Bob and Alice know the various Schreier transversals which are kept secret between them. Bob then encodes $W(h_i, h_{i+1}...)$ and computes in $\overline{G}$ the element $W(\rho(h_i), \rho(h_{i+1}), ..)$ which he sends to Alice. This is sent as a matrix if G is a linear group or as a permutation if $\overline{G}$ is a permutation group and so on.

Alice uses the algorithm for $\overline{G}$ relative to G to rewrite $W(\rho(h_i), \rho(h_{i+1}), ..)$ as a word $W^\star(x_1, ...x_n)$ in the generators of G. She then uses the Shreier transversal for K in G to rewrite using the Reidemeister-Schreier process $W^\star$ as a word $W^{\star\star}(k_1, ...k_s..)$ in the generators of K. Since K is free or has unique normal forms this expression for the element of K is unique. Once she has the word written in the generators of K she uses the transversal for H in K to rewrite again, using the Reidemeister-Schreier process, in terms of the generators for H. She then has a word $W^{\star\star\star}(h_i, h_{i+1}, ...)$ and using $h_i \to a, h_{i+1} \to b, ...$ decodes the message.

In actual implementation an additional *random noise factor* is added (see [FBX 1,2])

In [FBX 1,2] an inplementation of this process was presented that used for the base group G the classical modular group $M = PSL(2, \mathbb{Z})$. Further it was a polyalphabetic cipher which was secure.

The Magnus representation within the quotient $\overline{H}$ can now be used as the platform for this system. In particular we follow the oultine in [BFX 1,2] but now applied to $\overline{F}$ the faithful representation of a rank n free group in $\overline{H}$.

Witnin $\overline{F}$ we develop a list of finitely generated free subgroups $H_1, ...H_m$. In a practical implementation we assume that m is large. For each H_i we have a Schreier transversal

$$h_{1,i}, ...h_{t(i),i}$$

and a corresponding ordered set of generators

$$W_{1,i}, ...W_{m(i),i}$$

constructed from the Schreier transversal by the Reidemeister-Schreier process. It is assumed that each $m(i) >> l$ where l is the size of the plaintext alphabet, that is each subgroup has many more generators than the size of the plaintext alphabet. Although Bob and Alice know these subgroups in terms of free group generators what is made public are generating systems given in terms of the polynomials in noncommuting variables.

The subgroups on this list and their corresponding Schreier transversals can be chosen in a variety of ways. For example the commutator subgroup of the Modular group is free of rank 2 and some of the subgroups H_i can be determined from homomorphisms of this subgroup onto a set of finite groups. Finding a free subgroup and a representation was described in part in [6].

Suppose that Bob wants to send a message to Alice. Bob first chooses three integers (m, q, t) where

$$m = \text{ choice of the subgroup } H_m$$

$$q = \text{ starting point among the generators of } H_m$$

$$\text{for the substitution of the plaintext alphabet}$$

$$t = \text{ size of the message unit }.$$

We clarify the meanings of q and t. Once Bob chooses m, to further clarify the meaning of q, he makes the substitution

$$a \to W_{m,q}, b \to W_{m,q+1}, ...$$

Again the assumption is that $m(i) >> l$ so that starting almost anywhere in the sequence of generators of H_m will allow this substitution. The message unit size t is the number of coded letters that Bob will place into each coded integral matrix.

Once Bob has made the choices (m, q, t) he takes his plaintext message $W(a, b, ...)$ and groups blocks of t letters. He then makes the given substitution above to form the corresponding polynomials in $\overline{H}$ the restricted power series ring;

$$T_1, ...T_s.$$

We now introduce a *random noise factor*. After forming $T_1, ...T_s$ Bob then multiplies on the right each T_i by a random polynomial in $\overline{F}$ say R_{T_i} (different for each T_i). The only restriction on this random polynomial R_{T_i} is that there is no free cancellation in forming the product $T_iR_{T_i}$. This can be easily checked and ensures that the freely reduced form for $T_iR_{T_i}$ is just the concatenation of the expressions for T_i and R_{T_i}. Next he sends Alice the integral key (m, q, t) by some public key method (RSA, Anshel-Goldfeld etc.). He then sends the message as s random polynomials

$$T_1R_{T_1}, T_2R_{T_2}, ...T_sR_{T_s}.$$

Hence what is actually being sent out are not elements of the chosen subgroup H_m but rather elements of random right cosets of H_m in $\overline{F}$. The purpose of sending coset elements is two-fold. The first is to hinder any geometric attack by masking the subgroup. The second is that it makes the resulting words in the the Modular Group generators longer - effectively hindering a brute force attack.

To decode the message Alice first uses public key decryption to obtain the integral keys (m, q, t). She then knows the subgroup H_m, the ciphertext substitution from the generators of H_m and how many letters t each matrix encodes. She next uses the algorithms described in section 2 to express each $T_i R_{T_i}$ in terms of the free group generators of $\overline{F}$ say $W_{T_i}(y_1, .., y_n)$. She has knowledge of the Schreier transversal, which is held secretly by Bob and Alice, so now uses the Reidemeister-Schreier rewriting process to start expressing this freely reduced word in terms of the generators of H_m. Recall that Reidemeister-Schreier rewriting is done letter by letter from left to right. Hence when she reaches t of the free generators she stops. Notice that the string that she is rewriting is longer than what she needs to rewrite in order to decode as a result of the random polynomial R_{T_i}. This is due to the fact that she is actually rewriting not an element of the subgroup but an element in a right coset. This presents a further difficulty to an attacker. Since these are random right cosets it makes it difficult to pick up statistical patterns in the generators even if more than one message is intercepted. In practice the subgroups should be changed with each message.

The initial key (m, q, t) is changed frequently. Hence as mentioned above this method becomes a type of polyalphabetic cipher. Polyalphabetic ciphers have historically been very difficult to decode (see [H]).

4. Relations on the Variables

It was shown in [GB 2] that by imposing further relations on the variables, free nilpotent groups of all possible class size can also be embedded in quotients of H. This was used in [GB 2] to prove certain results concerning equations in free groups. This can be used further for encryption purposes. By imposing nilpotency relations on some of the variables in the power series, but not all, and keeping the relations secret a further level of seurity is imposed. This procedure is under development ([BBFGR]).

References

[AAG] I.Anshel,M.Anshel,D.Goldfeld, *An Algebraic Method for Public Key Cryptography*,Math.Res. Lett,6,1999, 287-291, Springer Verlag.

[GB 1] G.Baumslag, *Topics in Combinatorial Group Theory*, Birkhauser,1993.

[GB 2] G. Baumslag, *Residual Nilpotence and Relations in Free Groups*, J. Algebra, 2,1965, 271-285.

[BFX 1] G. Baumslag, B.Fine, and X.Xu, *Cryptosystems Using Linear Groups*, Appl. Alg. in Engineering,Communication and Computing,17,2006, 205-217.

[BFX 2] G. Baumslag, B.Fine, and X.Xu, *A Proposed Public Key Cryptosystem Using the Modular Group*, Cont. Math to appear.

[BCFRX] G. Baumslag,T.Camps, B.Fine,G.Rosenberger and X.Xu, *Designing Key Transport Protocols Using Combinatorial Group Theory*, Cont. Math. to appear.

[BBFR] G. Baumslag,Y.Brukhov, B.Fine and G.Rosenberger, *Some Suggestions for Noncommutative Algebraic Cryptography*, in preparation.

[BBFGR] G. Baumslag,Y.Brukhov, B.Fine, A.Gaglione and G.Rosenberger, *Using Nilpotent Relations In Cryptography*, under development.

[F] B.Fine, *The Algebraic Theory of the Bianchi Groups*, Marcel Dekker,1990.

[GP] D.Grigoriev and I. Ponomarenko, *Homomorphic Public-Key Cryptosystems Over Groups and Rings*, Quaderni di Matematica,2005 to appear.

[H] P.Hoffman, Archimedes Revenge, Fawcett-Crest, 1988.

[HGS] C.Hall, I. Goldberg, B. Schneier, *Reaction attaacks Against Several Public Key Cryptosystems*, Proceedings of Information and Communications Security ICICS 99, Springer-Verlag, 1999, 2-12.

[KoL] K.H. Ko,J.Lee,J.H. Cheon,J.W. Han, J.Kang, C.Park , *New Public-Key Cryptosystem Using Braid Groups*, Advances in Cryptology - CRYPTO 2000 Santa Barbara CA - Lecture Notes in Computer Science, Springer , 1880 , 166-183, Springer Verlag.

[Ko] N.Koblitz, *Algebraic Methods of Cryptography*, Springer, 1998.

[M] W. Magnus, *Rational Representations of Fuchsian Groups and Non-Parabolic Subgroups of the Modular Group* ,Nachrichten der Akad Gottingen , 1973, 179-189.

[MKS] W. Magnus, A. Karass and D. Solitar, *Combinatorial Group Theory*, Wiley Interscience,New York ,1968.

[St] R. Steinwandt, *Loopholes in two public key cryptosystems using the modular groups*, preprint Univ. of Karlsruhle,2000.

[X] Xiaowei Xu, *Cryptography and Infinite Group Theory*, Ph.D. Thesis CUNY 2006

[Y] A.Yamamura, Public Key cryptosystems using the modular groups, Lecture Notes in Comput. Sci. 1431,1998, 203-216.

Gilbert Baumslag,Department of Mathematics, City College of New York, New York, NY 10031

Yegor Brukhov,Department of Mathematics, City College of New York, New York, NY 10031

Benjamin Fine, Department of Mathematics, Fairfield University, Fairfield, Connecticut 06430, United States

Gerhard Rosenberger, Fachbereich Mathematik, Universitat Dortmund, 44227 Dortmund, Federal Republic of Germany