

# Challenge response protocols and free group definitions: Lecture 2

Gilbert Baumslag

© Draft date September 9, 2008

September 9, 2008

## 1 Challenge response protocols

Challenge response protocols are common place in computer usage.

### 1.1 Passwords and the SSG

The most familiar example of a challenge response protocol involves the use of passwords every time one logs in to another computer. Suppose Alice wants to log in to a computer called *Gauss*. *Gauss* has a record of Alice's password  $\mathbb{A}$  and when Alice attempts to login, *Gauss* challenges Alice for her password. Alice then types in  $\mathbb{A}$  and, if this is correct, *Gauss* allows Alice access. This can be annoying since Alice logs into many computers and often forgets her passwords, which for security reasons need to be changed periodically. The use of finite presentations suggests a more general simple approach. Here is the way this might work. Alice has a single finitely presented group assigned to her, in much the same way that a social security group is given to a person. We call this group, Alice's social security group, SSG for short, which we again denote by  $\mathbb{A}$ . *Gauss* has a record not of  $\mathbb{A}$  but of a finitely presented subgroup of  $\mathbb{A}$  which here we denote by  $\mathbb{A}_{Gauss}$ . Now when Alice logs into *Gauss*, *Gauss* ask Alice several questions about  $\mathbb{A}_{Gauss}$ . These questions can be made difficult to answer, requiring some group-theoretic software. We have built a package to handle questions about groups given by finite presentations called Magnus. It can be found on our web site [www.caissny.org](http://www.caissny.org). Alice has Magnus on her computer and so does *Gauss*. Access to *Gauss* is allowed provided the questions are answered correctly. Notice that  $\mathbb{A}$  can be used by Alice whether she is communicating to *Gauss* or to any other computer, say *Noether* since it has infinitely many sufficiently complicated finitely presented subgroups. In the latter instance the subgroup of  $\mathbb{A}$  on *Noether* will be denoted similarly, say by  $\mathbb{A}_{Noether}$ . So Alice does not have to remember anything. Her computer will do all the work for her and because of the complexity of the groups involved, the process is relatively secure. Each computer (actually anyone, anything, entity) has similarly a finitely presented group, its computer security group or CSG, and so *Gauss* will have a CSG. In corresponding with Alice, if extra security is needed, Alice will have a finitely presented subgroup of *Gauss*' CSG and if Alice needs to be sure that she is talking to *Gauss* she will fire off questions to *Gauss* about this subgroup. There are many challenge response protocols. This one appears to be somewhat different, making use of the complexity of finitely presented groups. I will discuss several of them in due course, in particular one using RSA called *keygen*,

which is in common use and involves the generation of keys. This provides for a “no password login” procedure which could well be replaced by a procedure using finitely presented groups.

## 1.2 Key exchange - the method of Diffie and Hellman

The basic idea behind this method can be described as follows. Alice wants to send Bob a message. She puts the message into a box and attaches a padlock to it so that no-one can open the box. The box is delivered to Bob who then puts his own padlock on the box and sends it back to Alice. Alice has the key to her padlock and so unlocks it and sends the box back to Bob which is now secured only by his padlock. He then unlocks it and is now able to read Alice’s message.

The approach by Diffie and Hellman, which was apparently discovered somewhat earlier by Malcolm Williamson of the British Intelligence service GCHQ, can be described as follows. Alice wants to send a message to Bob. She chooses a large finite group  $G$  of order  $q$ , usually a prime, to encode her message, which is recorded in  $G$  as an element  $x$ . The group  $G$  is public, i.e., known to everyone. Alice then chooses an integer  $a$  relatively prime to  $q$  and Bob also chooses another integer  $b$  also relatively prime to  $q$ . Alice then sends Bob the group element  $x^a$ , Bob sends back  $(x^a)^b$ . Since  $a$  and  $q$  are relatively prime, there is an algorithm which computes the greatest common divisor of  $a$  and  $q$  as a linear combination of  $a$  and  $q$  resulting in the equation

$$1 = \alpha a + \beta q.$$

Similarly the greatest common divisor of  $b$  and  $q$  can be expressed as a linear combination of  $b$  and  $q$ :

$$1 = \gamma b + \delta q.$$

Alice now forms

$$(x^{ab})^\alpha = x^{(a\alpha)b} = x^{(1-\beta q)b} = x^b$$

since  $x^q = 1$ . She then sends  $x^b$  back to Bob, who carries out a similar computation which yields  $x$ :

$$x^{b\gamma} = x^{(b\gamma)} = x^{1-\delta q} = x.$$

So Bob is able to receive the message  $x$  in this way. The group  $G$  is usually taken to be a group of large prime order.

This procedure can be carried over to finitely presented groups. This, in the context being discussed here, can be reworked in many different ways. One involves the use of groups with operators. Others take advantage of the complexity of the solution of the word problem, others use the complexity of multiplication in a group and still others use the complexity of the computation and identification of inverses. The first people to think of using groups to house a variation of the Diffie-Hellman procedure were Anshel, Anshel and Goldfeld.

## 1.3 RSA

The Diffie-Hellman approach involves three transmissions of information. Rivest, Shamir and Adleman invented a one-way method for secure communication, that bears their names. It’s invention was also predated in work by Clifford Cocks of the British Secret Service. In their scheme, Alice wants to provide a way for secure transmission of information to her contacts. She begins by selecting two large distinct primes, say  $p$  and  $q$  chosen so that it is hard to factor  $N = pq$ . In addition she chooses a number  $e$  which is relatively prime to  $(p-1)(q-1)$  with

$1 \leq e \leq (p-1)(q-1)$ . She is the only one who knows the number  $(p-1)(q-1)$ . She sends an email to everyone containing  $N$  and  $e$ . She then uses the Euclidean algorithm to express the greatest common divisor of  $(p-1)(q-1)$  and  $e$  as a linear combination:

$$1 = \alpha e + \beta(p-1)(q-1).$$

The number  $\alpha$  is also kept secret. Now Bob sends a message to Alice, which he first encodes as a positive integer  $m$  less than  $N$ . He then computes  $m^e$  and figures out the remainder  $r$  after division by  $N$ . We express this by writing  $m^e \equiv r \pmod{N}$ . Bob sends Alice the number  $r$ . Alice now computes  $r^\alpha$ .

We now show that

$$r^\alpha \equiv m \pmod{N}$$

which will allow Alice to obtain the message sent by Bob. To see this, observe first that

$$\alpha e \equiv 1 \pmod{(p-1)(q-1)}.$$

So

$$\alpha e \equiv 1 \pmod{p-1} \text{ and } \alpha e \equiv 1 \pmod{q-1}.$$

In other words

$$\alpha e = i(p-1) + 1 \text{ and } \alpha e = j(q-1) + 1.$$

Now if  $m$  is not a multiple of  $p$ , then  $m$  and  $p$  are relatively prime and so by Fermat's little theorem,

$$m^{p-1} \equiv 1 \pmod{p}$$

and therefore

$$m^{\alpha e} = m^{i(p-1)+1} \equiv m \pmod{p}.$$

If  $m$  is a multiple of  $p$ ,

$$m^{\alpha e} \equiv 0 \equiv m \pmod{p}.$$

So in every instance

$$m^{\alpha e} \equiv m \pmod{p}.$$

Similarly,

$$m^{\alpha e} \equiv m \pmod{q}.$$

So both  $p$  and  $q$  divide  $m^{\alpha e} - m$ . Since  $p$  and  $q$  are distinct primes,  $N = pq$  divides  $m^{\alpha e} - m$ , i.e.,

$$m^{\alpha e} \equiv m \pmod{N}$$

as claimed. So Alice can obtain the message sent by Bob.

**Exercise 1.1** *Read the article on Wikipedia about RSA.*

## 2 Free groups and their subgroups

### 2.1 Some notation, definitions and elementary results

We recall here some of material discussed in Lecture 1 and introduce some notation that will be used here and in the sequel.

To this end, let  $G$  be a group. We express the fact that  $H$  is a subgroup of  $G$  by writing  $H \leq G$ ; if  $H$  is a normal subgroup of  $G$  we write  $H \trianglelefteq G$ .

Let  $X \subseteq G$ . Then the subgroup of  $G$  generated by  $X$  is denoted by  $\text{gp}(X)$ . Thus, by definition,  $\text{gp}(X)$  is the smallest subgroup of  $G$  containing  $X$ . It follows that

$$\text{gp}(X) = \{x_1^{\varepsilon_1} \dots x_n^{\varepsilon_n} \mid x_i \in X, \varepsilon_i = \pm 1\}.$$

We call the product

$$x_1^{\varepsilon_1} \dots x_n^{\varepsilon_n} \quad (x_i \in X, \varepsilon_i = \pm 1)$$

an  $X$ -product. An  $X$ -product is termed *reduced* if

$$x_i = x_{i+1} \quad \text{implies} \quad \varepsilon_i + \varepsilon_{i+1} \neq 0 \quad (i = 1, \dots, n-1).$$

We have already defined free groups as those groups defined by an empty set of relations. It follows that a group  $G$  is free if it is generated by a set  $X$  and every non-empty reduced  $X$ -product is  $\neq 1$ ; we term  $X$  a free set of generators of  $G$ . We also say that  $X$  freely generates  $G$  or that  $G$  is free on  $X$ . Notice that if  $G$  is free on  $X$ , then two reduced  $X$ -products are equal if and only if they are identical.

**Theorem 2.1** 1. If  $G$  is free on  $X$  and also on  $Y$ , then  $|X| = |Y|$ ; this common cardinal number is termed the rank of the free group  $G$ .

2. Let  $X$  be a set. Then there exists a free group  $G$  freely generated by  $X$ , the so-called free group on  $X$ .

3. Let  $G$  be free on  $X$ . Then for every group  $H$  and every map  $\theta : X \rightarrow H$  there exists a homomorphism  $\varphi : G \rightarrow H$  such that  $\varphi|_X = \theta$

**Corollary 2.2** Every group is isomorphic to a factor group of a free group.

A group is termed an  $\alpha$ -generator group if it can be generated by a set of cardinality  $\alpha$ ; it is finitely generated if  $\alpha$  can be chosen to be finite.

**Exercise 2.3** Prove that the group of matrices generated by  $a$  and  $b$  given below is free of rank 2:

$$a = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}.$$

Let  $G$  again be a group,  $X \subseteq G$ . Then the least normal subgroup of  $G$  containing  $X$ , the so-called *normal closure* of  $X$  in  $G$ , is denoted by  $\text{gp}_G(X)$ . So

$$\text{gp}_G(X) = \text{gp}(g^{-1}xg \mid g \in G, x \in X).$$

We can now formalize the notion of a presentation of a group  $G$ . Suppose that  $F$  a free group freely generated by a set  $X$  and that  $\theta$  is a map from  $X$  into  $G$  such that

$$G = \text{gp}(X\theta).$$

Then the extension  $\varphi$  of  $\theta$  to  $F$  maps  $F$  onto  $G$  with kernel  $K$ . Suppose that

$$K = \text{gp}_F(R).$$

Then it follows, or if preferred, can be defined as such, that

$$G = \langle X; R \rangle \tag{1}.$$

As before, we term  $\langle X; R \rangle$  a presentation of  $G$ . Notice that such a presentation (1) comes equipped with an implicit map  $\theta : X \rightarrow G$  such that the extension of  $\theta$  to the free group  $F$  on  $X$  yields a homomorphism  $\varphi$  with kernel  $\text{gp}_F(R)$ .

If we identify  $X$  with its image in  $G$  then (1) implies, as indicated before, that  $X$  generates  $G$  and everything about  $G$  can be deduced from the fact that  $r = 1$  in  $G$  for every  $r \in R$ . As before, a group is finitely presentable or finitely presented if it has a finite presentation i.e. if

$$G = \langle X; R \rangle$$

where  $X$  and  $R$  are both finite.