Journal of Information Security, 2018, 9, 154-167
http://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

# Secure Passwords Using Combinatorial Group Theory

**Gilbert Baumslag[1], Benjamin Fine[2], Anja Moldenhauer[3], Gerhard Rosenberger[3]**

[1]Department of Mathematics, City University, New York, USA
[2]Department of Mathematics, Fairfield University, Fairfield, CT, USA
[3]Department of Mathematics, University of Hamburg, Hamburg, Germany
Email: fine@fairfield.edu, gerhard.rosenberger@uni-math,hamburg.de

**How to cite this paper:** Baumslag, G., Fine, B., Moldenhauer, A. and Rosenberger, G. (2018) Secure Passwords Using Combinatorial Group Theory. *Journal of Information Security*, **9**, 154-167.
https://doi.org/10.4236/jis.2018.92011

**Received:** February 7, 2018
**Accepted:** March 18, 2018
**Published:** March 21, 2018

## Abstract

Password security is a crucial component of modern internet security. In this paper, we present a provably secure method for password verification using combinatorial group theory. This method relies on the group randomizer system, a subset of the MAGNUS computer algebra system and corrects most of the present problems with challenge response systems, the most common types of password verification. Theoretical security of the considered method depends on several results in asymptotic group theory. We mention further that this method has applications for many other password situations including container security.

## Keywords

Password Security, Combinatorial Group Theory, Free Group Cryptography, Group Randomizer System

## 1. Introduction

This material essentially appeared in [1], but we feel it is an important application that should be more widely publicized and is a perfect entry for the present special volume on Cryptography and Internet security.

Secure password identification is a crucial component of modern internet security. Password verification is essential and this requires a backup system. Backup password security is handled most often by a challenge response system (see [2]) accompanying the password. In the simplest systems, this takes the form of secondary password questions such as the prover's mother's maiden name or place of birth. There are many inherent difficulties with these types of challenge response systems such as the trivial problem of the provers remembering their responses. More critical is the problem that this type of information for many people is readily available and easily found or guessed by would-be attackers or eavesdroppers. Challenge response systems are also subject to middleman attacks and replay attacks. There have been many attempts to alleviate these problems, including zero-knowledge password proofs and challenged responses somewhat based on RSA as well as timed out responses (see CRAM-MD5, Password Authenticated Key Agreement, [2] [3]).

This article presents an alternative method for challenge response password verification using combinatorial group theory. Further, this method is provably secure. It depends upon the theoretical and practical difficulty of solving the search membership problem within a given finitely presented group without knowing the presentation and the difficulty of solving systems of equations within free groups. This latter problem has been proved to be NP-hard. This alternative method uses the group randomizer system; a computer program that is a subset of MAGNUS, a much larger computer algebra system, designed to handle algorithmic problems in combinatorial group theory. MAGNUS was developed at CAISS, the Center for Algorithms and Interactive Scientific Software, a research laboratory housed at City College of the City University of New York and under the direction of the first author. The group randomizer system can be placed on a simple hand held computer device presently under development at CAISS. The system can also be used from computer to computer.

The group theoretic techniques have several major advantages over other challenge response systems. Using standard authentication terminology, the password presenter will be denoted the prover while the presentee is the verifier. The methods we present can be used for two-way authentication, that is the same method authenticates both the prover to the verifier and the verifier to the prover.

From the standpoint of cryptology, the method is a symmetric key authentication protocol. In its application, each prover has a standard password that is a common shared secret with the verifier. In addition, each prover is assigned a finitely presented group *G*. This group is called the challenge group. The total common shared secret between prover and verifier consists of $(P, G)$ where $P$ is a standard password and $G$ is the challenge group. The challenge group will provide an unlimited set of back-up challenges to the password. These challenges are in the form of group theoretical questions concerning *G*. The assignment of the challenge group to a given prover will be done randomly by the group randomizer system which we will explain. Cryptographically, we assume the adversary can steal the encrypted form of the group theoretic responses. From a security viewpoint, this does not present a problem. Each set of back-up challenges forms a virtual one time code as we will explain in the paper. Therefore, the adversary must steal three things—the original password, the challenge group and the group randomizer. Hence there is almost total security in this challenge response system. Further, there is an infinite supply of finitely presented groups to use as challenge groups and an infinite supply of challenge response questions that never have to be duplicated. These will be explained in the paper. Finally in distinction to other backup password protocols the group theoretic method is a two-way protocol between the prover and the verifier; while the verifier authenticates the prover's password, simultaneously the prover authenticates that he or she is dealing with the true verifier.

A major advantage of this technique it that it is provably secure. The proof of its security depends upon asymptotic group theory which we explain in Section 6. A result of Lysenok [4] implies that stealing the challenge group is NP-hard while a result of Jitsukawa [5] says that the asymptotic density of using homomorphisms (see Section 6) to attack the group randomizer protocol is zero.

In the next section, we provide a brief primer on combinatorial group theory and then a description of the group randomizer system. We then present several variations on how the group randomizer system can be used for secure password verification protocols. After this with we give the security model showing that it is provably secure.

Finally we describe how the group randomizer system and password methods can be used as a secure lock for container security.

This group randomizer system password security approach is part of a large program to use computational combinatorial group theory as a tool in secure data storage and data identification.

Sadly the first author, who inspired much of this work, passed away during the preparation of the paper. We thank him posthumously for his many ideas.

## 2. Finitely Presented Groups and Combinatorial Group Theory

Combinatorial group theory attempts to study groups via group presentations. A group presentation can be thought of as an encoded method to describe a given group.

A group presentation for a group $G$ consists of a set of generators $X$ for $G$ and a set $R$ of defining relators on the generators $X$. In this case we write $G = \langle X; R \rangle$. For purposes of this paper and the challenge response password authentication protocols we propose, we may assume that $G$ is a finitely presented group. By this we mean that both the set of generators $X$ and the set of defining relators are finite. The books by Baumslag [6], Lyndon and Schupp [7], Magnus, Karrass and Solitar [8] Camps, GrRebel and Rosenberger [9] are standard references for this material. Another reference for the use of combinatoiral group theory in cryptography is the book by Baumslag, Fine, Kreuzer and Rosenberger [10].

Consider a finite alphabet, $X = \{x_1, \cdots, x_n\}$ and the formal inverses $\{x_1^{-1}, \cdots, x_n^{-1}\}$. Then the set $X = \{x_1, \cdots, x_n\}$ is a set of generators for a group $G$ if every element $g \in G$ has an expression as a word in the generators and their inverses. The identity is considered the empty word. We do not assume that this expression is unique. A relator on $X$ is a word $W\left(x_1,\cdots,x_n\right)$ which represents the identity in $G$. A relator of the form $x_i x_i^{-1}$ or $x_i^{-1}x_i$ is called a trivial relator. A set $R=\left\{R_1,\cdots,R_k\right\}$ of words on $X$ is a complete set of defining relators if every relator $W$ can be transformed into the empty word by finitely many insertions and deletions of elements of $R$ and trivial relators. Two words $W_1,W_2$ on the generators represent the same group element $G$ if and only if $W_1$ can be transformed into $W_2$ by insertions and deletions of relators in $R$ and trivial relators. If $X$ is a set of generators for $G$ and $R$ is a set of defining relators we say $G$ has the presentation $G=\langle X;R\rangle$. A finite presentation will always define a group (see [8]) for which this is the presentation. A set of words $W_1,\cdots,W_m,\cdots$ in one-to-one correspondence with the elements of $G$ is said to determine a set of normal forms for $G$.

Presentations are in no way unique, however in principle all group theoretic questions about $G$ should be answerable given a presentation. For a group presentation, the word problem asks whether given any word on $X$, is there an algorithm to determine in finitely many steps whether this word represents the identity in $G$. In general the word problem is undecidable. That is there exist group presentations for which it can be proved that no such algorithm exists (see [6]). A particularly nice class of groups, which have both normal forms and solvable word problems, are the automatic groups (see [11]).

For a group $G$ and a subgroup $H$ of $G$ the membership problem (also called the generalized word problem) is the problem of determining algorithmically whether a given word $W$, written in terms of the generators of $G$, lies in the subgroup $H$. As with the word problem, the membership problem is in general undecidable. Clearly a solution to the membership problem, in a given group, implies a solution to the word problem.

Fundamental to combinatorial group theory is the concept of a free group. Let $A$ be a set. Then, a group $F$ is free on $A$ if every mapping $f:A\to G$, where $G$ is a group, can be extended to a unique homomorphism of $F$ to $G$. We denote this by $F[A]$. A group is free if it is free on some set $A$. It can be proved that given a set $A$, there exists a free group on $A$ and further if two sets $A_1$ and $A_2$ have the same size then the corresponding free groups $F[A_1]$ and $F[A_2]$ are isomorphic. If $A=\left\{x_1,\cdots,x_n\right\}$ is a finite set, we say that $F[A]$ is a free group of rank $n$ and sometimes denote this by $F_n$.

From the viewpoint of group presentations free groups are groups with a presentation with an empty set of defining relators. If $F$ is free on $X=\left\{x_1,\cdots,x_n\right\}$ then there are no nontrivial relators on $X$ and a presentation for $F$ is $F=\langle x_1,\cdots,x_n\rangle$. In this case the elements of $F$ can be considered as reduced words on the alphabet $\left\{x_1,\cdots,x_n\right\}^{\pm1}$. The identity element is considered as the empty word. Reduced means that we can cancel any occurrences of $x_i x_i^{-1}$ or $x_i^{-1}x_i$. It is clear that each word has a unique reduced form and hence the word problem for $F$ is solvable.

A well-known theorem due to Nielsen and Schreier (see [6]) says that a proper subgroup of a free group is again a free group, of course on a different set of generators. For our purposes what is important is that a finitely generated subgroup of a free group is completely determined by a finite set of words which have only trivial cancellation between them.

Essential to the group randomizer system protocols is that algorithmically both the word problem and the membership problems are solvable in free groups. That is given a free group $F = F\left(x_1, \cdots, x_n\right)$ of finite rank $n$ we can decide algorithmically whether or not a word $W = W\left(x_1, \cdots, x_n\right)$ represents the identity; given two words $W_1 = W_1\left(x_1, \cdots, x_n\right)$, $W_2 = W_2\left(x_1, \cdots, x_n\right)$ we can decide algorithmically whether $W_1 = W_2$ in $F$; given a finitely generated subgroup $H$ of $F$ and a word $W$, we can decide algorithmically whether the element defined by $W$ lies in $H$. Details on these algorithmic procedures can be found in [6] [8] [9].

## 3. The Group Randomizer System

The group randomizer system is a computer program that can handle several elementary tasks involving finitely presented groups. It is a subset of MAGNUS, a large computer algebra system, developed at CAISS—the Center for Algorithms and Interactive Scientific Software. The program MAGNUS is specifically designed to handle computations and algorithmic problems in combinatorial group theory. At present there are various versions of the group randomizer, including a portable hand held version now under development.

The scope of a particular group randomizer system will depend on the type of login protocol or cryptographic protocol desired. At the most basic level the group randomizer system has the ability to do do the following things:

1) Recognize a finite presentation of a finitely presented group with a solvable word problem and manipulate arbitrary words in the alphabet of generators according to the rewriting rules of the presentation. In particular if the group is automatic the group randomizer can rewrite an arbitrary word in the generators in terms of its group normal form.

2) Given a finite presentation of a group $G$, with a solvable word problem, recognize whether two free group words have the same value in the given group when considered in terms of the given generators of the group.

3) Randomly generate free group words on an alphabet of any finite size.

4) Recognize and store sets of free group words $W_1, \cdots, W_k$ on an alphabet $x_1, \cdots, x_n$ and rewrite words $W\left(W_1, \cdots, W_k\right)$ as the corresponding word in $x_1, \cdots, x_n$.

5) Given a free group of finite rank on $x_1, \cdots, x_n$ and a set of words $W_1, \cdots, W_k$ on $x_1, \cdots, x_n$ solve the membership problem in $F$ relative to $H = \left\langle W_1, \cdots, W_k \right\rangle$, the subgroup of $F$ generated by $W_1, \cdots, W_k$.

6) Given a stored finitely presented group or a stored set of free group words, the randomizer can accept a random free group word and rewrite it as a normal form in the finitely presented group in the former case or as a word in the

ambient free group in the latter case.

7) Evaluate a group word on a set of $n \times n$ rational matrices.

In the next section we show how this can be used for secure password verification in both directions—verifier to prover and prover to verifier.

## 4. Secure Password Verification

We now present several variations on secure password verification using the group randomizer. First we give an overall outline of the protocol.

1) General Outline of the Authentication Protocol

At a theoretical level, this protocol is a symmetric key cryptographic authentication protocol. Both the prover and verifier use a single private key to both encrypt and decrypt within the authentication process. At first, the prover and verifier must communicate directly, either face-to-face or by a public key method, to set the private shared secret. This is the model now used for most password/password back-up schemes. We assume that both the user and verifier have a group randomizer system. For security analysis, we assume that an adversary or eavesdropper has access to the encrypted form of the transmission but is passive in that the adversary will not change any transmissions.

Step (1): The prover and verifier communicate directly to set up a common shared secret $(P,G)$ where $P$ is a standard password and $G$ is a challenge group. Each prover's challenge group is unique to that prover. The challenge group is a finitely presented group with a solvable word problem and satisfying the strong generic free group property (see Section 5). The password is chosen by the prover while the challenge group is randomly chosen by the group randomizer system.

Step (2): The prover presents the password to the verifier. The group randomizer of the verifier presents a group theoretic "question" (see parts (2) and (3)) concerning the challenge group $G$ to the prover. The assumption is that this "question" is difficult in the sense that it is infeasible to answer it if the group $G$ is unknown. The question is then answered by the group randomizer. This is repeated a finite number of times. If the answers are correct the prover (and the password) is verified.

Step (3): The protocol is then repeated from the viewpoint of the prover, authenticating the verifier to the prover.

2) Free Subgroup Method

The first method we present uses a free group as the basic group theoretic object.

We assume that both the prover and the verifier has a group randomizer. Each prover has a standard password. Suppose that $F$ is a free group on $\{x_1,\cdots,x_n\}$. The prover's password is linked to a finitely generated subgroup of a free group given as words in the generators—that is the prover's password is linked to $W_1,\cdots,W_k$ where each $W_i$ is a word in $x_1,\cdots,x_n$. The group $G=\langle W_1,\cdots,W_k\rangle$ is called the challenge group. In general, $k \neq n$. The prover doesn't need to know the generators. The randomizer can randomly choose words from this subgroup and then freely reduce them. The verifier has the challenge group or subgroup also stored in its randomizer.

The prover submits his or her standard password to the verifier. This activates the verifier's randomizer to the prover's set of words. The verifier now submits a random free group word on $y_1,\cdots,y_k$ to the prover's randomizer say $W(y_1,\cdots,y_k)$. The prover's randomizer treats this as $W(W_1,\cdots,W_k)$ and then reduces it in terms of the free group generators $x_1,\cdots,x_n$ and rewrites it as $W^{\star}(x_1,\cdots,x_n)$. The verifier checks that this is correct—that is $W(W_1,\cdots,W_k)=W^{\star}(x_1,\cdots,x_n)$ on the free group on $x_1,\cdots,x_n$. If it is the verifier continues and does this three times (or some other finite number of times). There is one proviso. A challenge word or submitted word can never be reused. The prover's randomizer will recognize if a presented challenge word has been submitted previously and reject it. This is a further authentication to the prover of the verifier and directly hinders middle man attacks.

To verify that the verifier is legitimate the process is repeated from the prover's randomizer to the verifier.

An attacker only has access to the transmitted words. Given a series of free group words, reconstructing the subgroup involves solving systems of equations in free groups. Solving such systems has been shown to be NP-hard (see Section 6). To prevent an attacker using an already used word to gain access the group randomizer system allows a free group word, submitted as a challenge word, to be used only once. If an attacker gets access to the verifier and submits an already submitted word or vice versa from the prover this will red flag the attempt. We also suggest that if there is a previously used word, indicating perhaps an attack, the group randomizer should change the prover's group. The beauty of this system is that this can be done extremely easily—change several of the words for example. Essentially this presents an essential one-time keypad each time the prover presents the password. Hence there is very strong security in this back-up system. The map $y_i \to W_i$ is a homomorphism and an attacker can manipulate various equations in an attempt to solve. Presumably if there are enough equations the words $W_1,\cdots,W_k$ can be discovered. However in section 6 we will present a security proof based on several results in asymptotic group theory showing that this can not happen with asymptotic density one.

We suggest a noise/diffusion enhancement. The prover's challenge group generator words $W_1,\cdots,W_k$ are indexed. With each use the randomizer applies a random permutation $\phi$ on $\{1,\cdots,k\}$ to scramble the indices. These permutations are coded and stored both in the prover's randomizer and the verifier's. These coded permutations are set at the time of initialization of the protocol and become part of the common shared secret. This prevents a length based attack by an eavesdropper since discovering for example what $W_{37}$ is, is of no use since it will be indexed differently for the next use. The coded permutation is sent as part of the challenge.

3) General Finitely Presented Group Method

Rather than working with an ambient free group we can work with a given finitely presented group with a solvable membership problem. Let $G = \langle X ; R \rangle$ be the group. As before we assume that the group $G$ has a solvable word problem and satisfies the strong generic free group property. Further, as before, we assume that both the prover and the verifier has a group randomizer. Each prover has a standard password. Suppose that $X = \{x_1, \cdots, x_n\}$ and $F$ is a free group on $\{x_1, \cdots, x_n\}$. The prover's password is linked to a finitely generated subgroup of *G*, again given as words in the generators *X*. That is the prover's password is linked to $W_1, \cdots, W_k$ where each $W_i$ is a word in $x_1, \cdots, x_n$. As before, $k \neq n$. The randomizer can randomly choose words from this subgroup and then reduce them via the finite presentation. The verifier has the group or subgroup also stored in its randomizer.

The remainder of the procedure is exactly the same as in the free group case. The prover submits his or her standard password to the verifier. This activates the verifier's randomizer to the prover's set of words. The verifier now submits a random free group word on $y_1, \cdots, y_k$ to the prover's randomizer say $W(y_1, \cdots, y_k)$. The prover's randomizer treats this as $W(W_1, \cdots, W_k)$ and rewrites it as $W^{\star}(x_1, \cdots, x_n)$. The verifier checks that this is correct, that is $W(W_1, \cdots, W_k) = W^{\star}(x_1, \cdots, x_n)$ with equality this time in the group *G*. If it is true then the verifier continues and does this three (or some other finite number) of times. There is one proviso. The verifier submits a word to the prover only once so that a submitted word can never be reused. The prover's randomizer will recognize if it has ( this is a verification to the prover of the verifier).

To authenticate that the verifier is legitimate the process is repeated from the prover's randomizer to the verifier.

As in the free group method, an attacker has access only to the transmitted words. Given a series of group words it is infeasible to reconstruct the group. Further, as in the free group method, a given challenge response word is to be used only once. Since we assume that the group has the strong generic free group property, it follows that previous challenge words cannot be used to discover the challenge group or subgroup of the challenge group.

## 5. The Strong Generic Free Group Property

Part of the theoretical security of the group randomizer protocols depends upon the strong generic free group property and asymptotic density. Asymptotic density is a general method to compute densities and/or probabilities on infinite discrete sets where each individual outcome is tacitly assumed to be equally likely. The origin of asymptotic density lie in the attempt to compute probabilities on the whole set of integers where each integer is considered equally likely. The method can also be used where some probability distribution is assumed on the elements. It has been effectively applied to determining densities within infinite discrete finitely generated groups where random elements are considered as being generated from random walks on the Cayley graph of the group. The paper by Borovik, Myasnikov and Shpilrain [12] provides a good general description of this method in group theory. Let $\mathcal{P}$ be a group property and let $G$ be a finitely generated group. We want to determine the measure of the set of elements which satisfy $\mathcal{P}$. For each positive integer $n$ let $B_n$ denote the $n$-ball in $G$. Let $|B_n|$ denote the actual size of $B_n$ (which is an integer since $G$ is finitely generated) or the measure of $|B_n|$ if a distribution has been placed on the elements of $G$. Let $S$ be the set of elements in $G$ satisfying $\mathcal{P}$. The asymptotic density of $S$ is then

$$\lim_{n\to\infty}\frac{|S\cap B_n|}{|B_n|}$$

provided this limit exists. We say that the property $\mathcal{P}$ is generic if the asymptotic density of the set $S$ of elements satisfying $\mathcal{P}$ is one.

This concept can be easily extended to properties of finitely generated subgroups, We consider the asymptotic density of finite sets of elements that generate subgroups that have a considered property. For example to say that a group has the generic free group property we mean that

$$\lim_{m,n\to\infty}\frac{|S_m\cap B_{m,n}|}{|B_{m,n}|}=1$$

where $S_m$ is the collection of finite sets of elements of size $m$ that generate a free subgroup and $B_{m,n}$ is the collection of $m$-element subsets within the $n$-ball. We refer to the papers [12] and [13] for terminology and further definitions.

We say that a group $G$ has the generic free group property if a finitely generated subgroup is generically a free group. For example a result of Epstein [14] says that the group $GL(n,R)$ satisfies the generic free group property. A group $G$ has the strong generic free group property if given randomly chosen elements $g_1,\cdots,g_n$ in $G$ then generically they are a free basis for the free subgroup they generate. Jitsukawa [5] proved that free groups have the strong generic free group property. That is given $k$ random elements $W_1,\cdots,W_k$ in the free group on $y_1,\cdots,y_n$ then with asymptotic density one $W_1,\cdots,W_k$ are a free basis for the subgroup they generate. We compare this with the Nielsen-Shreier theorem that says that $W_1,\cdots,W_k$ generate a free group. In the context of the group randomizer protocols the strong generic free group property implies that if $V_1(y_1,\cdots,y_m),\cdots,V_k(y_1,\cdots,y_m)$ have already been presented as challenge words then the density is zero that a new challenge word $V(y_1,\cdots,y_m)$ lies in the subgroup generated by $V_1,\cdots,V_k$ and hence a homomorphism attack is nullified.

The strong generic free group property has been extended to arbitrary free products of infinite groups and many other amalgams including surface groupsaid groups and br by Fine. Myasnikov and Rosenberger [13] and Carstensen, Fine and Rosenberger [15].

## 6. Security Analysis of the Group Randomizer Protocols

For the security analysis of the group randomizer password protocols, we make the security assumption that an adversary has access to the coded group theoretic responses. The strength of the proposed protocol is that an attacker must steal three things; the original password, the group randomizer and the challenge group. There is no access without all three. This immediately nullifies middelman attacks. If the adversary pretends to be the verifier to obtain the group words the attack is thwarted by the facts that the prover can verify the verifier and further if the attacker just transmits from the middle, nothing can be stolen since each time through a new challenge word must be used. Further the group randomizer has an infinite supply of both subgroups and challenge responses that are done randomly. In addition since a challenge word can be used only once the protocol nullifies replay attacks. Since challenge responses are machine to machine there is an essential probability of zero of an incorrect response. The protocol shuts down with an incorrect response and hence repeat attacks are harmless.

These are in distinction to answer-driven challenge-response systems where a prover often forgets or misspells a response. In these systems a prover is usually permitted several opportunities to answer. This makes these systems susceptible to both middleman and repeat attacks.

There are two theoretical attacks that must be dealt with. Relative to these attacks, the security of the system, and hence a security proof for the protocol, is provided by several results in asymptotic group theory.

The most straightforward attack is for the adversary to collect enough challenge words and responses. This provides a system of equations in a free group (or a finitely presented group)

$$y_i = W_i\left(x_1,\cdots,x_n\right), i = 1,\cdots,m.$$

An adversary can then break the protocol by solving the system

$$y_i = W_i\left(x_1,\cdots,x_m\right)$$

to obtain the challenge group.

A result of Lysenok [4] shows that solving such systems of equations in free groups (and in most finitely presented groups) is NP-hard. Hence this method of attack is impractical in most cases.

A second method of attack is based on the following. The mapping $y_i \to W_i$ is a homomorphism. If a challenge word appears in the subgroup generated by previous challenge words then an attacker can use this to answer a challenge without ever solving for the challenge group. However this approach fails due to the strong generic free property. Each set of challenge words is a free basis for the subgroup they generate with asymptotic density 1. Hence as explained in the previous section the probability converges to zero that a new challenge word is in the subgroup generated by previous challenge words.

There are several enhancements that can perhaps improve security:

1) Permutation Diffusion and Noise: Both the prover's randomizer and the verifier's randomizer have a fixed coded set of permutations on $\{1,\cdots,k\}$ where $k$ is the rank of the challenge group. Each presentation of a challenge word is accompanied by a random one of these permutations. If $\phi$ is the presented permutation then the challenge word $W\left(y_1,\cdots,y_k\right)$ is evaluated on $W\left(y_{\phi(1)},\cdots,y_{\phi(k)}\right)$. As mentioned earlier this prevents hinders attacks by an eavesdropper since discovering for example what $W_{37}$ is, is of no use since it will be indexed differently for the next use. The set of coded permutations is agreed upon at initialization and becomes part of the common shared secret.

2) Short Challenge Words: In each challenge word $W\left(y_1,\cdots,y_k\right)$, we assume that not all $k$ variables are used. In actual implementation we specify that the number of variables $t$ that appear in any challenge word is small relative to $k$ the rank of the challenge group. For example, we may have $5\le t\le 8$ with $k=256$. Hence each equation that can be stolen by an attacker has only a relatively small number of variables. This increases the number of equations necessary to impact on a homomorphism solution which in turn is NP-hard to solve.

3) Frequent Reset: We recommend that the challenge group be reset relatively often. Since this protocol is symmetric the rest must be done via some sort of direct communication as in the original initialization of the secret key.

## 7. Actual Implementation of a Group Randomizer System Protocol

The actual implementation of a workable group randomizer system protocol involves several choices of parameters and subprograms. These include

1) The choice of the rank of the ambient free group in the free group method.

2) An enhancement program which takes a random choice of words $W_1,\cdots,W_k$ in a free group $F$ and finds a new set of words $V_1,\cdots,V_k$ generating the same subgroup for which the words formed in $V_1,\cdots,V_k$ have a great deal of free cancellation. This involves what is called Nielsen transformations (see [7] [8] [9]).

3) The choice of parameter sizes for the lengths of the randomly chosen words. In an actual implementation all words in the generators will have lengths between $a$ and $b$ where $a$ and $b$ are to be determined. All words used as test logins will have lengths between $c$ and $d$ with $c$ and $d$ to be determined, The optimal values for these parameters must be determined.

4) The implementation of a coded permutation system on $\{1,\cdots,k\}$ where $k$ is the rank of the challenge group and so that a coded permutation can be sent with each challenge word.

5) The development of an automatic reset protocol for the challenge group. In an ideal situation this can be done without actually communicating the changes between verifier and prover—that is each randomizer system does the same protocol automatically when reset is called for.

## 8. Alternative Methods Using Rational Matrices

Free groups have faithful representations in terms of rational and integral matrices (see [16] [17]), or integral $2\times 2$ matrices there is an algorithm to go back and forth between a free group and the corresponding matrices in its representation. This is explained in [16]. This can be used in several ways in conjunction with the group randomizer.

First the basic free group method can be enhanced with the matrix representation in the following manner. We assume that the group randomizer has been extended to include the algorithm to go back and forth between a free group and $SL(2,Z)$ mentioned above. Then what can be sent from verifier to use is an integral matirx rather than a free group word. This is then deciphered by the algorithm into a word in the free group. The prover's group randomizer then proceeds as in the standard free group method rewriting thw word in terms of the stored password subgroup. Finally, this is rewritten in terms of the matrix representation and an integral matrix sent back to the verifier. This presents a further time obstacle to an attacker.

There is a simpler variation of the whole system solely using matrices. Each prover is assigned a set of $m\times m$ invertible rational matrices $M_1, M_2, \cdots, M_k$. These are linked to the prover's standard password as before. Each matrix is assigned a free group variable $x_1 = M_1, \cdots, x_k = M_k$. As in the standard free group method when the prover presents a password the verifier sends a a free group word $W(x_1, \cdots, x_k)$. The prover's group randomizer evaluates this word on $M_1, \cdots, M_k$ to obtain a rational matrix $M = W(M_1, \cdots, M_k)$. This matrix $M$ is then sent back to the verifier. The verifier checks to see if the evaluation of the sent word is correct or not. As before to verify that the verifier is legitimate the process is repeated from the prover's randomizer to the verifier.

An attacker here sees a matrix polynomial equation $W(x_1, \cdots, x_k) = M$. This must be solved for matrices $M_1, \cdots, M_k$ in order to obtain access. For $n \geq 3$ there is no factoring algorithm or solution algorithm for such equations and hence if $k$ is large (or even moderately large) the equation is feasibly insolvable. This again present a one-time keypad type of approach. As mentioned earlier, if the matrices are over the reals *R*, the group $GL(n,R)$ has the generic free group property.

## 9. The Group Randomizer and Container Security

Another very common security problem is container safety or container security. Here a container means a large shipping unit and the fear is that some contraband material or dangerous people will be stored or shipped via a container. Our contention is that the group randomizer can be used here as a secure lock.

We make the assumption that the shipper is legitimate and that we are only interested in the main lock—that is we don't consider the situation where a terrorist group saws through the center of the container. We only want to check that the main lock has not been tampered with.

We assume that the main lock has been outfitted with a group randomizer. When it is sealed, the group randomizer is given a finitely presented group as in the password case. Since the protocol is symmetric key, the group is be transmitted via some secure key exchange to the receiver. The main lock is set so that when it is opened or tampered with the group is lost. When the container gets to its destination, its group is checked by a group randomizer at the far end. This of course can be done electronically. Without stealing the group, as in the password case, the lock cannot be tampered with.

## Acknowledgements

The authors would like to thank the referee for many helpful suggestions concerning the exposition of this paper. This is especially true in Section 4 on secure password verification.

## References

[1] Baumslag, G., Brjukhov, Y., Fine, B. and Troeger, D. (2010) Secure Password Verification Using Combinatorial Group Theory. *Groups-Complexity-Cryptology*, **2**, 67-82. https://doi.org/10.1515/gcc.2010.005

[2] Wikipedia. Challenge-Response Authentication. The Free Encyclopedia. https://en.wikipedia.org/wiki/Challenge%E2%80%93response_authentication

[3] Challenge-Response System Based on RSA. http://www.cag.lcs.mit.edu/rugina/ssh-procedures

[4] Lysenok, I. (2006) Equations over Free Groups. Private Communication.

[5] Jitsukawa, T. (2002) Malnormal Subgroups of Free Groups. In: Gilman, R., Shpilrain, V. and Myasnikov, A.G., Eds., *Computational and Statistical Group Theory*, Contemporary Mathematics, Vol. 298, 83-96. https://doi.org/10.1090/conm/298/05115

[6] Baumslag, G. (1993) Topics in Combinatorial Group Theory. Birkhäuser.

[7] Lyndon, R. and Schupp, P. (1978) Combinatorial Group Theory. Springer.

[8] Magnus, W., Karass, A. and Solitar, D. (1968) Combinatorial Group Theory. Wiley Interscience, New York.

[9] Camps, T., Rebel, V. and Rosenberger, G. (2008) Einführung in die kombinatorische und die geometrische Gruppentheorie. Berliner Studienreihe zur Mathematik Band 19, Heldermann Verlag.

[10] Baumslag, G., Fine, B., Kreuzer, M. and Rosenberger, G. (2015) A Course in Mathematical Cryptography. De Gruyter, Berlin.

[11] Fine, B. and Rosenberger, G. (1999) Algebraic Generalizations of Discrete Groups. Marcel-Dekker, New York.

[12] Borovik, A., Myasnikov, A.G. and Shpilrain, V. (2002) Measuring Sets in Infinite Groups. In: Gilman, R., Shpilrain, V. and Myasnikov, A.G., Eds., *Computational and Statistical Group Theory*, Contemporary Mathematics, Vol. 298, 21-42. https://doi.org/10.1090/conm/298/05112

[13] Fine, B., Miasnikov, A. and Rosenberger, G. (2009) Generic Properties of Amalgams. *Groups-Complexity-Cryptography*, **1**, 51-61.

[14] Epstein, D.B.A. (1971) Almost All Subgroups of Lie Group Are Free. *Journal of Algebra*, **19**, 261-262. https://doi.org/10.1016/0021-8693(71)90107-4

[15] Carstensen, C., Fine, B. and Rosenberger, G. (2010) On Asymptotic Densities and Generic Properties in Finitely Generated Groups. *Groups-Complexity-Cryptology*, **2**, 113-121. https://doi.org/10.1515/gcc.2010.008

[16] Baumslag, G., Fine, B. and Xu, X. (2006) Cryptosystems Using Linear Groups. *Applicable Algebra in Engineering, Communication and Computing*, **17**, 205-217. https://doi.org/10.1007/s00200-006-0003-z

[17] Baumslag, G., Fine, B. and Xu, X. (2006) A Proposed Public Key Cryptosystem Using the Modular Group. In: Fine, B., Gaglione, A.M. and Spellman, D., Eds., *Combinatorial Group Theory*, *Discrete Groups*, *and Number Theory*, Contemporary Mathematics, Vol. 421, 35-44. https://doi.org/10.1090/conm/421/08025