Group Cryptography Using High Performance Computing

Matthew Craven and Daniel Robertz
Centre for Mathematical Sciences, Plymouth University

Motivation

The *Anshel-Anshel-Goldfeld* key exchange protocol is based upon the *multiple conjugacy problem* (MCP) for a finitely-presented group. The hardness in breaking this protocol relies on the difficulty in solving the corresponding equations for the conjugating element in the group. Two protocols based on polycyclic groups were recently proposed [5, 7] and shown to be resistant to length-based attack. We propose a parallel approach which runs on multicore high-performance architectures. It is shown to be more efficient and also more successful than previous attempts to break these protocols. Through comprehensive analysis of experiments run with a `GAP` implementation, we demonstrate that the proposed platform is not as secure as first thought and also show that existing measures of cryptographic complexity are not optimal.

Introduction

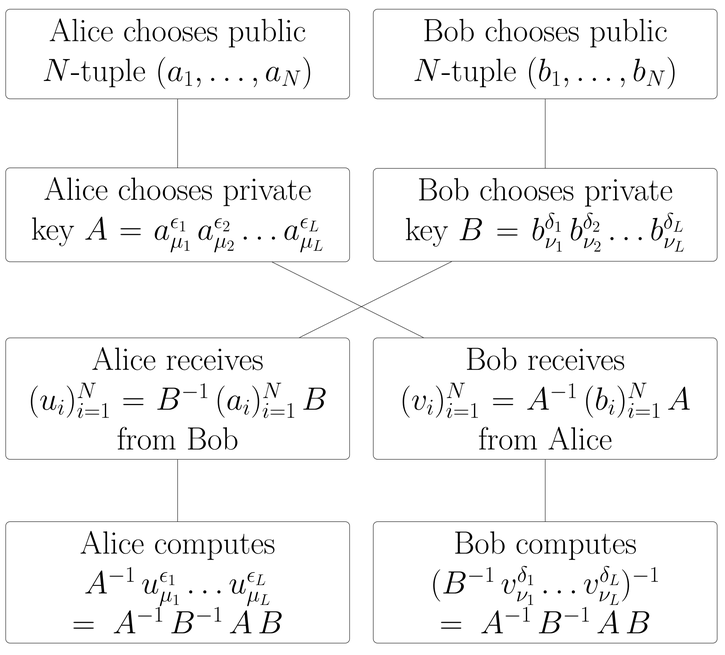
The AAG KEP is the problem of solving a system of equations in a finitely-presented group for A:

$$A^{-1}\,(b_i)_{i=1}^N\,A = (v_i)_{i=1}^N.$$

It was proposed over the braid groups in 1999, and during 2000-2006 was attacked from many directions [6]. As it became clear that the protocol was not as secure as first reported, search began for alternative platform groups to the braid groups. These comprise metabelian groups, Garside groups and polycyclic groups [1], among others.

Public key cryptography

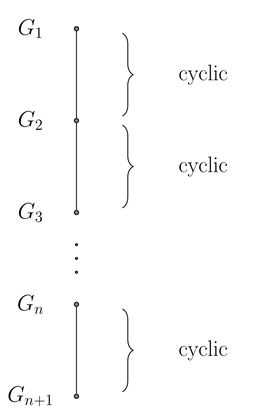
Alice and Bob wish to exchange cryptographic keys so that they may communicate in secret. Following the *AAG Key Exchange Protocol* they first choose words $a_1, \ldots, a_N$ and $b_1, \ldots, b_N$ in the generators $g_1, \ldots, g_n$ of a fixed group $G = \langle\, g_1, \ldots, g_n \mid r_1, \ldots, r_m \,\rangle$.



Now Alice and Bob are sharing the secret $A^{-1}\,B^{-1}\,A\,B$ without having disclosed their private keys!

Polycyclic groups

A group G is said to be *polycyclic* if there exists a subnormal series $G = G_1 \rhd G_2 \rhd \ldots \rhd G_{n+1} = \{1\}$ for some $n \in \mathbb{N}$ such that each factor group G_i/G_{i+1} is cyclic.



For general groups G which are given by a *finite presentation*

$$G = \langle\, g_1, \ldots, g_n \mid r_1, \ldots, r_m \,\rangle$$

in terms of generators $g_1, \ldots, g_n$ and relations $r_1, \ldots, r_m$, it is undecidable whether a word w in the generators represents the identity element in G.

Given a polycyclic group G by a finite presentation, every element has a unique normal form $g_a^{e_a} \ldots g_{n-1}^{e_{n-1}}\,g_n^{e_n}$, where $e_k \in \mathbb{Z}$, so that the word problem for polycyclic groups has an effective solution.

Evolutionary algorithms

EAs are probabilistic population-based optimisation algorithms which use the principles of evolution. An EA begins with an initial population of candidate solutions to a problem. Simulating Darwinian operations such as elitist selection, mutation and crossover, solutions breed to create new generations. Operations are performed relative to a best-fitness-first ranking of the population. In this way, evolution continues producing candidate solutions of equal or higher fitness to those found previously, terminating when an exact solution is obtained. EAs provide efficiency gains over more restrictive algorithms, navigating very large search trees of candidate solutions in parallel. Given a word $w = f_{i_1}^{e_{i_1}} f_{i_2}^{e_{i_2}} \ldots f_{i_r}^{e_{i_r}}$, where $e_j \in \mathbb{Z} \setminus \{0\}$, $\{f_{i_j}\}_{k=1}^r$ are generators of a free group F, and consecutive pairs of $i_1, i_2, \ldots, i_r \in [n]$ are distinct, length is measured by

$$\ell(w) = \sum_{k=1}^{r} |e_{i_k}|, \qquad \ell_{wt}(w) = \sum_{k=1}^{r} w_{i_k} |e_{i_k}|,$$

where the weight w_j is defined as the sum of the lengths of the normal forms of the commutators $[g_j, g_k]$ in G for $j \neq k$. The EA cost function is a metric of how close a candidate word is to an exact solution of the MCP, and is a tuple of components composed from various statistics of the above length functions of the form $\ell_*\left(\alpha^{-1}\,a_i\,\alpha\,b_i^{-1}\right)$. The aim is to minimise each component. To distinguish candidate solutions, cost tuples of individuals α are compared lexicographically. An individual α is an exact solution of the MCP if and only if any one of the components is zero.

Examples of polycyclic groups

1. For each positive integer n the group of matrices

$$\begin{pmatrix} 1 & x_1 & x_2 & \ldots & x_n & z \\ 0 & 1 & 0 & \ldots & 0 & y_1 \\ 0 & 0 & 1 & \ldots & 0 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & y_n \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix}, \qquad x_i, y_j, z \in \mathbb{Z},$$

and matrix multiplication as binary operation, is an example of a *Heisenberg group*. It is an infinite polycyclic group.

2. The additive group $\mathcal{O}$ of the ring of integers of a number field F is a finitely generated free abelian group and its group of units U is a finitely generated abelian group. By Dirichlet's unit theorem, the rank of U is $s+t-1$, where s is the number of embeddings of F into $\mathbb{R}$ and t the number of conjugate pairs of embeddings of F into $\mathbb{C}$. In this way we obtain non-nilpotent infinite polycyclic groups $\mathcal{O} \rtimes U$.

HPC and parallelism

Adopting techniques from *High Performance Computing (HPC)*, our approach distributes the work of computing normal forms of elements in polycyclic groups among several processors (see also Master-Slave Layout). The *Message Passing Interface* standard (MPI) is used for communication between the different processes.

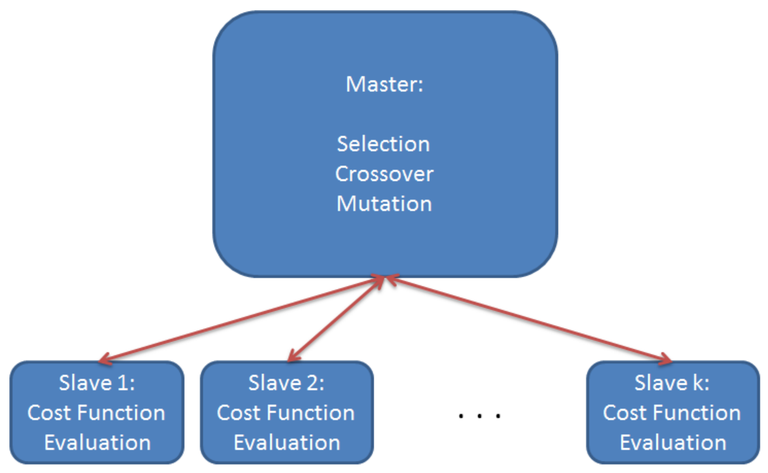
Computer algebra system GAP

The computer algebra system `GAP` (Groups, Algorithms, Programming) provides a programming language and functions for studying a variety of algebraic structures, in particular groups. Its package `polycyclic` is dedicated to computations with polycyclic groups. The `pargap` package runs `GAP` on a HPC platform using the MPI standard. We use both `pargap` and `polycyclic` to implement an attack of the AAG protocol using high performance computing (see also Master-Slave Layout).

The population (of candidate solutions) is sorted according to the cost function and updated by random mutations.

```
n := 1;
while n <= maxsteps do
  C := ParList(L, cost);
  SortParallel(C, L);
  if C[1] = 0 then
    successes := successes + 1;
    break;
  fi;
  # random mutations subj. to parameters,
  # updating the population in L
  n := n+1;
od;
```

Master-slave layout



Results

1. Heisenberg

n	[7]	EA	$\overline{g}$	$\overline{t}$	$\overline{t}/\overline{g}$	$\overline{g}_{.10}$	$\overline{t}_{.10}$
5	29%	100%	42.8	29.8	0.70	40.6	27.0
6	69%	100%	64.9	74.1	1.14	56.4	60.8
7	51%	97%	171.6	223.1	1.30	85.8	117.5
8	62%	90%	795.5	932.8	1.17	364.1	454.1

2. Number Field

d	[5]	EA	$\overline{g}$	$\overline{t}$	$\overline{t}/\overline{g}$	$\overline{g}_{.10}$	$\overline{t}_{.10}$
1	98%	100%	4.8	0.4	0.08	4.6	0.2
2	100%	100%	34.4	19.4	0.56	22.9	19.4
3	100%	98%	76.2	20.3	0.27	20.9	6.7
5	35%	74%	1469.1	1367.3	0.93	1208.1	922.4
7	8%	60%	2283.7	2508.4	1.10	2222.9	2015.4
9	5%	22%	6419.0	33101.2	5.16	6980.6	23972.2
11	5%	25%	5863.1	9989.5	1.70	6370.9	8472.1

From left to right: cryptographic parameter ($\sim$ problem hardness), success rate reported for the length attack followed by that for our approach, mean number of generations and time (sec.), trimmed means of both measures.

References

[1] B. Eick and D. Kahrobaei, Polycyclic Groups: A New Platform for Cryptology?, preprint, 2004, (`http://arxiv.org/abs/math/0411077`).

[2] M. J. Craven, An Evolutionary Algorithm for the Solution of Two-Variable Word Equations in Partially Commutative Groups, *Studies in Comp. Intell.* 153, Springer (2008), 3-19.

[3] M. J. Craven and H. C. Jimbo, An Evolutionary Algorithm Solution of the Multiple Conjugacy Search Problem in Partially Commutative Groups with Applications, *Groups, Complexity and Cryptology* 4 (2012), 135-165.

[4] M. J. Craven and D. Robertz, A Parallel Evolutionary Approach to Solving Systems of Equations in Polycyclic Groups, submitted March 2016.

[5] D. Garber, D. Kahrobaei and H. T. Lam, Length-Based Attacks in Polycyclic Groups, *J. Math. Crypt.* 9 (1) (2015), 33-43.

[6] D. Garber, S. Kaplan, M. Teicher, B. Tsaban and U. Vishne, Probabilistic Solutions of Equations in the Braid Group, *Adv. Appl. Math.* 35 (2005), 323-334.

[7] D. Kahrobaei and H. T. Lam, Heisenberg Groups as Platform for the AAG Key-Exchange Protocol, *22nd IEEE ICNP* (2014), 660-664.