Two identification protocols based on Cayley graphs of Coxeter groups*

Feliú Sagols
Mathematics Department
Cinvestav-IPN
Mexico City, Mexico

Guillermo Morales-Luna
Computer Science Department
Cinvestav-IPN
Mexico City, Mexico

September 3, 2010

Abstract

A challenge-response identification protocol is introduced, based on the intractability of the word problem in some Coxeter groups. A Prover builds his public key as the set of leaves of a tree in the Cayley graph of a Coxeter group, and the tree itself is his private keys. Any challenge posed by a Verifier consists of a subset of the public key, and the Prover shows his knowledge of the private key by providing a subtree having as set of leaves the challenge set. Any third party aiming to impersonate the Prover faces a form of the word problem in the Coxeter group. Although this protocol maintains the secrecy of the whole private key, it is disclosing some parts of it. A second protocol is introduced which is indeed a transcription of the already classical zero-knowledge protocol to recognize pairs of isomorphic graphs.

Keywords. Authentication, Coxeter groups, identification protocols, random spanning trees, word problem.
Classification. 94A62, 20F10, 20M05, 68P25

1 Introduction

In user identification and authentication several challenge-response protocols have been proposed [5]. Any *prover* is able to generate instances and corresponding solutions of computationally hard problems: the instances play the role of public keys of him, while the solutions are corresponding private keys. Any *verifier* chooses instances as challenges to the prover, and only the prover is able to submit proof of his knowledge by providing the corresponding solutions.

In the current paper we introduce a challenge-response authentication protocol based on the difficulty to solve the word problem in Coxeter groups, or

*This research is partially supported by Mexican Conacyt.

the shortest path problem in the corresponding Cayley graphs which is indeed intractable due to the graph sizes.

In section 2 the proposed identification protocol is introduced within the frame of a graph in which the shortest path problem is intractable.

The word problem is solvable for finite groups and its complexity may be polynomial with respect to the order of the group. However, this problem may be intractable in terms of the length of a presentation of the group. In section 3 we describe the Coxeter groups whose Cayley graphs are adequate for the introduced authentication protocol and we discuss its complexity and its robustness. Since the prover is replying with partial words contained within its private key this protocol is not zero-knowledge. In section 4 we introduce a zero-knowledge protocol allowing the prover to prove its knowledge of the private key without disclosing any part of it. Finally, some conclusions are drawn.

2 Identification protocol

Let K be a finite non-empty set, $k = \mathrm{card}(K)$, and let $f : \mathbb{N} \to \mathbb{N}$ be a map of growth order $\Omega(n)$. Let us fix an integer $n \in \mathbb{N}$. The set $K^{f(n)}$ is properly the collection of words with symbols in K of length $f(n)$ and its cardinality is $N_{kn} = k^{f(n)}$. Let $\mathcal{G} = (K^{f(n)}, E)$ be a graph over $K^{f(n)}$. Then the *Handshaking Lemma* states $2\,\mathrm{card}(E) = \sum_{\mathbf{x}\in K^{f(n)}} \deg(\mathbf{x})$, hence if the valency of each node has $d \in \mathbb{Z}^+$ as an upper bound, we have $\mathrm{card}(E) \leq \frac{d}{2} N_{kn}$ and the height of any tree in a spanning forest in $\mathcal{G}$ is of order $\ell = O(\log N_{kn}) = O(f(n) \log k)$. The Shortest Path Problem may become computationally expensive in $\mathcal{G}$. For instance, Dijkstra's Algorithm has average complexity

$$O(\mathrm{card}(E) + N_{kn} \log N_{kn}) = O\left((d + f(n))\, k^{f(n)}\right).$$

Let us consider an *identification scenario*: A Prover tries to convince a Verifier that he has a piece of knowledge without revealing that piece during the message exchange among them. The Verifier poses successive questions and according to the obtained replies, he decides whether the Prover knows or not the piece of knowledge. Indeed, we may realize that the piece of information is properly the private key corresponding to a public key released by the Prover.

Then, in the graph $K^{f(n)}$, we may select a subgraph $\mathcal{G}$ in which the Shortest Path Problem is intractable, mainly due to a great number of nodes in $\mathcal{G}$. For a tree, i.e. a connected subgraph with no cycles, in $\mathcal{G}$ the Shortest Path Problem whose instances are leaves in the tree is trivial for anyone knowing the tree. Thus, the Prover builds a tree within $\mathcal{G}$, publishes the set of leaves as his public key and keeps the tree itself as the corresponding private key. Any agent aiming to check whether the Prover knows the private key may challenge the Prover with a subset of nodes in the Prover's public key. Then only the Prover may submit efficiently a tree having as leaves the challenge subset providing in this way a proof of his knowledge of the private key.

Public-key protocol

Precondition. The graph $\mathcal{G}$ shall be known by both the Prover and the Verifier.

Initialization. The Prover chooses a tree T as a subgraph of $\mathcal{G}$. The tree itself is a *private key* of him. The Prover publishes the collection of leaves of T as his *public key.*

Identification protocol. Repeat

1. the Verifier chooses a proper subset N of nodes in the public key of the Prover, and sends it to the Prover as a challenge,
2. the Prover calculates a subtree T_N of T having N as set of leaves and sends it back to the Verifier.
3. the Verifier receives T_N and checks that it is a tree with N as set of leaves, if not he rejects the procedure,

until the Verifier is fully satisfied.

Since the Prover knows the tree T, he is able to succeed in the challenge at any stage of the protocol. However, due to robustness purposes, the protocol should comply with the following requirements:

- the set N at step (1) should pose an intractable instance of the Spanning Tree Problem at the graph $\mathcal{G}$, and
- the set N at step (1) should be strictly lesser than the whole set of leaves in order to prevent the disclosure of the whole private key owned by the Prover.

Indeed, any subtree of $\mathcal{G}$ having as leaves the public key will allow any intruder to impersonate successfully the Prover in the protocol. The problem to find such a subtree can be solved in linear time with respect to the number $N_{\mathcal{G}}$ of vertexes in the graph $\mathcal{G}$, but may become an expensive problem with respect to $\ell_{\mathcal{G}} = O(\log N_{\mathcal{G}})$ (which may be proportional to the diameter of $\mathcal{G}$). Of course, if the challenging set N coincides with the public key of the Prover then the Prover shall disclose his private key at step (2), and the same situation occurs if the iterative segment is repeated using the sets of a covering of the public key as challenging sets. Nevertheless, at any stage in which the public key remains uncovered, and the current instance of the Spanning Tree Problem remains intractable, any third party, trying to impersonate the Prover, may be unable to succeed in a new iteration of the protocol.

3 The word problem as warrant of the protocol robustness

Let us recall that a *presentation* of a group G is a pair (C, R), where C is a collection of *generators* and $R \subset \{\alpha = \beta |\ \alpha, \beta \in C^*\}$ is a collection of *relations* such that $G \approx F(C)/\langle\langle R \rangle\rangle$ where $F(C)$ is the free-nonabelian group generated

by C and $\langle\langle R\rangle\rangle$ is the normal subgroup in $F(C)$ spanned by the words $\alpha\beta^{-1}$, or *relators*, with $(\alpha = \beta) \in R$. The *word problem* consists in deciding for a given word $\sigma \in (C \cup C^{-1})^*$ whether it lies or not at the normal subgroup $\langle\langle R\rangle\rangle$, or equivalently in finding for a given $g \in G$ the least word, according to a well founded ordering, $\sigma \in (C \cup C^{-1})^*$ such that $g = \sigma$ in G.

It is well known the Novikov's theorem, formulated in the 50's, asserting that there exist finitely presented groups in which the word problem is unsolvable. An example of such a group [3] is the following:

$$\begin{aligned} \text{Generators: } C &= \{a, b, c, d, e\} \\ \text{Relations: } R &= \{ac = ca, ad = da, bc = cb, bd = db, ce = eca, de = edb, \\ &\qquad cdca = cdcae, caaa = aaa, daaa = aaa\}. \end{aligned}$$

Although the Cayley graph of this group may serve to realize our proposed protocol, it is infinite.

However, let us remark that the word problem may be solvable but intractable in practice, as is the case of the word problem on an Artin group [6]. Let us recall this concept in a succinct way. For two symbols x, y and an integer $\ell \in \mathbb{Z}^+$, let $(xy)^{[\ell]}$ be the prefix of length ℓ of the word $(xy)^\ell$, or in other words $(xy)^{[\ell]} = (xy)^{\lfloor \frac{\ell}{2} \rfloor}\eta$ where $\eta = x$ if ℓ is odd and η is the empty word otherwise. Let C be a finite set of generators and let $M \in (\mathbb{Z}^+ \cup \{+\infty\})^{C \times C}$ be a matrix indexed by $C \times C$ with entries which are either positive integers or an infinite value. Let us introduce the collection R_M of relations $(xy)^{[m_{xy}]} = (yx)^{[m_{yx}]}$, with $\{x, y\} \in C^{(2)}$. A Coxeter group results by adding the relations $x^2 = 1$ (i.e. $m_{xx} = 2$). The word problem in these groups has exponential complexity with respect to the number of generators, $n = \text{card}(C)$, and has served as basis of several public-key cryptosystems (e.g. [6]).

Let us consider one of these Coxeter groups $G(n, M)$. It possesses n generators and $d = \frac{1}{2}(n+1)n = O(n^2)$ relators. Let $f(n) = \lceil \log_n(o(G(n, M)))\rceil$ be the logarithm in base n of the group's order, then $f(n) = \Omega(n)$. By associating each element in the group with its minimal expression as a word over the generators, we have that the Cayley graph of the group $G(n, M)$ can be realized as a graph with vertexes in the set $K^{f(n)}$, with $K = C$, as required in the first section of the current paper.

However, let us consider the normal subgroup $H_M = \langle\langle R_M\rangle\rangle$ in the free group generated by C. Let $\mathcal{G}_M$ be the graph whose set of vertexes is H_M and the edges are pairs of the form $(\sigma\alpha\tau, \sigma\beta\tau)$ with either $(\alpha = \beta) \in R_M$ or $(\beta = \alpha) \in R_M$. The word problem here remains of exponential complexity with respect to n.

Any vertex in this graph can be expressed as a word of length at most $f(n)$ over the alphabet C, thus it can be represented by a bit string of length $O(f(n) \log n)$.

In order to use the graph $\mathcal{G}_M$ in the identification protocol introduced in section 2, the prover shall construct a suitable subgraph $\mathcal{G}_m$ of $\mathcal{G}_M$ and a spanning tree of $\mathcal{G}_m$, where $m \in \mathbb{N}$ is a parameter in the protocol. The following random

selection of a spanning tree is based on the already classical approach due to Broder [2] and Aldous [1].

Let us assume already built an undirected connected subgraph $\mathcal{G}_m = (V_r, E)$ consisting of $r = r(m) \in \mathbb{Z}^+$ vertexes. For each node $v \in V_r$ let d_v be the valency of node v. For any two vertexes $u, v \in V_r$, let p_{uv} be d_v^{-1} if $\{v, u\} \in E$ and let it be 0 otherwise. Then the matrix $P = (p_{uv})_{u,v \in V_r}$ is the transition matrix of a simple Markov chain within the graph $\mathcal{G}$. A spanning tree may uniformly be selected by the following procedure [2]:

`SpanningTreeGeneration`

1. Departing from an arbitrarily chosen vertex $v_0 \in V_r$, let $\{x_\tau\}_{\tau=0}^{t_e}$ be a random walk of shortest length covering the whole graph $\mathcal{G}$, i.e. for each $v \in V_r$ there is a minimum time $t_v \leq t_e$ such that $x_{t_v} = v$.

2. Let T be the tree consisting of the edges $\{x_{t_v - 1}, v\}$, with $v \in V - \{v_0\}$.

T is a spanning tree because it consists of exactly $r - 1$ edges. Then, one can see [2] that $t_e = O(r^3)$ in worst cases but in general one may expect $t_e = O(r \log r)$. Also, under some symmetry conditions on the graph $\mathcal{G}_m$, one may expect [1] that the ratio among the leaves and the vertexes of $\mathcal{G}_m$ is bounded by $\exp\left(-\frac{r-1}{2r}\right)$ and the diameter $\Delta(T)$ of the generated graph is $O(\sqrt{r})$.

Consequently, given an integer $m \in \mathbb{N}$, the Prover may generate a tree as a subgraph of the graph $\mathcal{G}_M$, with m as the expected of leaves in the tree, by selecting a connected graph with $r(m) = \lceil \sqrt{e}\, m \rceil$ vertexes and then by uniformly selecting one of its spanning trees:

`TreeGenerationWithExpectedNumberOfLeaves`

1. Generate a subgraph $\mathcal{G}_m$ with $r = \lceil \sqrt{e}\, m \rceil$ vertexes.

2. Using `SpanningTreeGeneration`, generate a spanning tree T of $\mathcal{G}_m$.

3. Output tree T.

At step (1) in the above procedure, it is possible to implement a node-generation according to a breadth-first traversing in order to select a fixed number of neighbors at each newly discovered node. This with the aim to fulfill, to the greatest extent, the regularity conditions required for the graphs analyzed in [1]. In this case, the expected number of leaves at the produced tree is m, and the whole representation of the tree T and its set of leaves has length of order $O(m\, f(n) \log n)$. This is the size of the messages exchanged by the Prover and the Verifier during the identification protocol.

As an alternative to the above construction, let us consider at any node v of the whole Cayley graph $\mathcal{G}_M$ a probability distribution

$$(p_{uv} |\ \{v, u\} \text{ is an edge in } \mathcal{G}_M)\,.$$

Then, $\mathcal{G}_M$ has a structure of a simple Markov chain. Given an integer $m \in \mathbb{N}$, the tree T is produced as in `SpanningTreeGeneration` with a random walk halting until $\lceil \sqrt{e}\, m \rceil$ pairwise different nodes are visited in $\mathcal{G}_M$.

4 Zero-knowledge protocol

Let C be a finite set of generators and let $M \in (\mathbb{N} \cup \{+\infty\})^{C \times C}$. Let $\mathcal{G}_M$ be the Cayley graph of the Coxeter group determined by M. According to the procedures sketched at section 3, a prover chooses a tree T in $\mathcal{G}_M$ as a private key and publishes the yield $Y(T)$ as his public key. In order to authenticate the prover, any verifier selects a subset $J \subset Y(T)$ and poses it as a challenge to the prover, who shall respond with the minimum tree S having as yield the challenge set J. Since J is a proper set of $Y(T)$, S is a proper subtree within T, thus the whole T is kept secret although a part of it has been disclosed. Obviously either by round repetition of the protocol or by the collusion of several verifiers the whole private key T may be disclosed.

Let us modify the protocol in order to obtain a zero-knowledge protocol similar to the well known protocol to recognize pairs of isomorphic graphs [4]:

The prover possesses both the public key $Y(T)$ and the private key T. The verifier knows $Y(T)$.

`ZeroKnowledgeIdentification`

Repeat

1. the verifier choose two leaves $v_0, v_1 \in Y(T)$ and he sends them to the prover,
2. the prover finds a path h connecting v_0 with v_1 within T, and chooses randomly an intermediate point v_2 on h. Let h_0 be the segment of h connecting v_0 with v_2 and let h_1 be the segment of h connecting v_2 with v_1. Then the prover chooses a path g in the Cayley graph starting at v_2, and the sends the ending point u of g to the verifier,
3. the verifier chooses a bit $b \in \{0, 1\}$ and he sends it to the prover (indeed the verifier is requiring a path from u to v_b),
4. the prover responds with $f = g \star (h_b)$ (here $\star$ is path-juxtaposition),
5. the verifier checks that f connects indeed v_b with u,

until either the prover fails or the verifier is satisfied.

With this protocol the verifier cannot identify any partial information contained in the prover's private key.

5 Conclusions

The first introduced identification protocol is robust due to the rapid growth of the involved graphs rendering intractable the shortest path problem within this graph, for any pair of vertexes, thus the public keys are not required to be too long. If they consist of just two extreme vertexes, the proposed challenge is rather difficult for any intruder. However, the number of repetitions of the protocol is bounded by the number of leaves forming the public-key. The

protocol is robust as long as the private key is just partially known. If the whole public key of the prover is posed as a challenge, then the private key is the due response. In this extreme case, an impersonating party may succeed in any further iteration of the identification protocol. The second protocol, which is indeed of zero-knowledge, is mimicking the classic recognition of pairs of isomorphic graphs.

References

[1] David Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM J. Discrete Math.*, 3(4):450–465, 1990.

[2] Andrei Z. Broder. Generating random spanning trees. In *FOCS*, pages 442–447. IEEE, 1989.

[3] Donald J. Collins. A simple presentation of a group with unsolvable word problem. *Illinois Journal of Mathematics*, 30(2):230–234, 1986.

[4] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, New York, NY, USA, 2008.

[5] Rupert J. Hartung and Claus-Peter Schnorr. Public key identification based on the equivalence of quadratic forms. In Ludek Kucera and Antonín Kucera, editors, *MFCS*, volume 4708 of *Lecture Notes in Computer Science*, pages 333–345. Springer, 2007.

[6] James Hughes and Allen Tannenbaum. Length-based attacks for certain group based encryption rewriting systems. In *Sécurité des Communications sur Internet, SECI-02*, 2002.