On public-key cryptosystems based on combinatorial group theory

Jean-Camille Birget*
Department of Computer Science,
Rutgers University - Camden,
Camden, NJ 08102, U.S.A.
birget@camden.rutgers.edu

Spyros S. Magliveras
Department of Mathematical Sciences
Florida Atlantic University, 777 Glades Road
Boca Raton, FL 33431, U.S.A.
spyros@fau.edu

Michal Sramka
Department of Mathematical Sciences
Florida Atlantic University, 777 Glades Road
Boca Raton, FL 33431, U.S.A.
sramka@math.fau.edu

Abstract

We analyze and critique the public-key cryptosystem, based on combinatorial group theory, that was proposed by Wagner and Magyarik in 1984. This idea is actually not based on the word problem but on another, generally easier, premise problem. Moreover, the idea of the Wagner-Magyarik system is vague, and it is difficult to find a secure realization of this idea. We describe a public-key cryptosystem inspired in part by the Wagner-Magyarik idea, but we also use group actions on words.

Keywords: Public-key cryptosystem, combinatorial group theory, Richard Thompson groups, \((\mathsf{NP} \cap \mathsf{coNP})\)-complete premise problems.

*JCB is supported in part by NSF grant CCR-0310793.
1 Introduction

A number of public-key cryptosystems based on combinatorial group theory have been proposed since the early 1980s, the first of which was probably the outline of Wagner and Magyarik [14]. A good overview of various other group-based systems is given in the dissertation of M.I. González Vasco [6]; see also [8].

In this paper we present a critique of the Wagner-Magyarik system, and propose a public-key cryptosystems based on finitely presented groups with hard word problem, and which are also transformation groups.

In order to make the paper more self-contained we give some basic definitions from combinatorial group theory. More details and rigor can be found in texts like [11] or [12].

Let $G$ be a group, defined by a presentation $(X, R)$, where $X = \{x_1, x_2, \ldots\}$ is a set of generators and $R = \{r_1, r_2, \ldots\}$ is a set of relators. When the sets $X$ and $R$ are both finite we say that the group $G$ is finitely-presented. A word $w$ over $X$ is a finite sequence of elements of the set $X \cup X^{-1}$. The empty word is the empty sequence, of length 0. A word which defines the identity element in the group $G$ is called a relator. We say that two words $w$ and $w'$ are equivalent for the presentation $(X, R)$ iff the following operations, applied a finite number of times, transform $w$ into $w'$:

(T1) Insertion of one of the relators $r_1, r_1^{-1}, r_2, r_2^{-1}, \ldots \in R \cup R^{-1}$, or of a trivial relator (of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$ with $x_i \in X$) at the beginning of a word, at the end of a word, or between any two consecutive symbols of a word.

(T2) Deletion of one of the relators $r_1, r_1^{-1}, r_2, r_2^{-1}, \ldots$ or of a trivial relator, if it forms a block of consecutive symbols in a word.

An application of one transformation of the form (T1) or (T2) is called a rewrite step. Every element $g$ of $G = (X, R)$ can be described by a word over $X \cup X^{-1}$, usually in many ways; the length of the shortest word that describes $g$ is called the word length of $g$. For a word $w$ over some fixed alphabet we denote the length of $w$ by $|w|$. Also, for $g \in G = (X, R)$ we denote the word length of $g$ by $|g|$. The word problem of a group with generating set $X$, as introduced by Max Dehn in 1911, is the following decision problem: For an arbitrary word $w$ over $X \cup X^{-1}$, is $w$ equivalent to the empty word?

In the 1950’s, Novikov and Boone independently showed that there are finite group presentations whose word problem is undecidable. It is an important fact that the decidability and the complexity of the word problem of a finitely generated group depend only on the group, and not on the generators or the
presentation chosen (provided that one sticks to finite generating sets). In other words, if $G$ has decidable word problem for some finite generating set $X$ then $G$ has decidable word problem for every finite generating set. Concerning complexity, a change of the finite generating set changes the complexity only linearly (see [13]). Therefore, we are allowed to talk about "the word problem of a group $G$" without referring to a specific presentation.

It was proved more recently that there are finitely presented groups whose word problem is NP-complete [15], [4], or whose word problem is coNP-complete [1].

By a group with easy word problem we will understand a group whose word problem is decidable in deterministic polynomial time. The other groups are said to have a hard word problem.

We will also deal with the following variant of the word problem, which we call the word choice problem. Let us fix a group $G$ with a finite generating set $X$, and let us fix two words $w_0$ and $w_1$ over $X \cup X^{-1}$.

**Input:** A word $w$ over $X \cup X^{-1}$.

**Premise:** $w$ is either equivalent to $w_0$ or to $w_1$.

**Question:** Is $w$ equivalent to $w_0$?

Note that this is a "premise problem"\(^1\), i.e., a problem with restrictions (pre-condition) on the input; an algorithm for solving a premise problem can assume that the pre-condition holds, and is not required to give correct answers (or any answer at all) on inputs that violate the pre-condition.

The word choice problem is rather different from the word problem. E.g., for a finitely presented group, the word choice problem is always decidable; and for a group with word problem in NP or in coNP, the word choice problem is in NP \cap coNP. One sees from these examples that the word choice problem can be much easier than the word problem.

## 2 Critique of the Wagner-Magyarik system

In 1984 Wagner and Magyarik [14] proposed a public-key cryptosystem "based on the word problem". The general scheme follows.

**Setup:** Let $X$ be a finite set of generators, and let $R$ and $S$ be finite sets of relations such that the group $G = (X, R)$ has a hard word problem, and the group $G' = (X, R \cup S)$ has an easy word problem, i.e., there exists an algorithm $A$ that solves the word problem in $G'$ in polynomial time.

Choose two words $w_0$ and $w_1$ which are not equivalent in $G'$ (and hence not equivalent in $G$ either).

\(^1\)In the complexity literature, premise problems are usually called promise problems; however, the word 'premise' is the appropriate logical term; look up 'premise' in the Merriam-Webster Dictionary http://www.m-w.com/home.htm
Public key: The presentation \((X, R)\) and the words \(w_0\) and \(w_1\).

**Encryption:** To encrypt a single bit \(i \in \{0, 1\}\), pick \(w_i\) and transform it into a ciphertext word \(w\) by repeatedly and randomly applying the transformations (T1) and (T2) for the presentation \((X, R)\).

**Decryption:** To decrypt a word \(w\), run algorithm \(A\) in order to decide which of \(ww_0^{-1}\) and \(ww_1^{-1}\) is equivalent to the empty word for the presentation \((X, R)\).

The private key is thus the set \(S\).

To make their system concrete, Wagner and Magyarik propose the following collection of finitely-presented groups: The set of generators is \(X = \{x_1, x_2, \ldots, x_m\}\) and the set of relators is any set of words of the following three types:

(R1) \(x_ix_jx_kx_{\ell}^{-1}x_i^{-1}x_j^{-1}x_{\ell}^{-1}\)

(R2) \(x_ix_jx_kx_{\ell}^{-1}x_j^{-1}x_k^{-1}\)

(R3) \(x_ix_jx_kx_{\ell}^{-1}x_k^{-1}x_{\ell}^{-1}\)

where \(x_i, x_j, x_k, \) and \(x_{\ell}\) stand for generators or inverses of generators.

For the private key \(S\) they propose any set of words of the following three types:

(S1) \(x_i\) (elimination of a generator)

(S2) \(x_ix_i^{-1}\) (collapse of two generators to one)

(S3) \(x_ix_jx_i^{-1}x_j^{-1}\) (commutator of two generators)

where \(x_i\) and \(x_j\) are any generators.

**Critique**

1. **Vagueness of the general scheme:** In its general form the Wagner-Magyarik cryptosystem is far too vague. To turn their idea into an actual cryptosystem, design steps would need to be specified: (1) How do we find appropriate presentations \((X, R)\) and \((X, R \cup S)\)? (2) How do we find appropriate words \(w_0\) and \(w_1\)? (3) How is the random application of the transformations (T1) and (T2) carried out, and when does it stop? (4) Finally, once all these design choices have been specified, how secure is this cryptosystem?

2. **Vagueness and insecurity of the concrete specification:** In their specific example, Wagner and Magyarik give an answer to question (1), albeit an unsatisfactory one. Questions (2), (3) and (4) are left open. The concrete
relators proposed for the set $S$ are inadequate. For any value of the security parameter $m$, there are at most $m + m(m - 1) = m^2$ possible choices for relators if types (S1), (S2), (S3). A reaction attack is possible, as described in [7]. Moreover, an attacker could find the private key through a chosen-ciphertext attack: For any relator $s$ of types (S1), (S2), (S3), the attacker considers the word $w_0s$, encrypts it (by applying the transformations (T1) and (T2) several times) and then observes the decryption, say $w$. If $w$ is equivalent to $w_0$ in $G$, the attacker learns that $s$ belongs to $S$ (or is implied by relators in $S$, which means that one might as well assume that it is in $S$). The complexity of the attack is $O(m^2)$.

Moreover, it is not obvious whether the word problem of the proposed presentations above is hard. It is certainly not hard for every choice of (R1), (R2), (R3) (e.g., some of the choices lead to commutative groups). It seems possible that for some choices of (R1), (R2), (R3) is hard, but this is not known at the moment.

3. Spurious keys: Another problem (already mentioned in [14]) is the existence of spurious keys. More precisely, in order to decrypt one does not explicitly need the presentation $(X, R \cup S)$. Any homomorphic image of $G$ with easy word problem will do the job; so, even if $S$ might be hard to find, one also has to prove that any homomorphic image of $G$ with easy word is hard to find; this adds to the difficulty of proving the security of any concrete cryptosystem that follows the Wagner-Magyarik approach.

4. Word choice problem: An analytical flaw in the Wagner-Magyarik paper (and subsequent papers that comment on their paper) is the claim that the system is based on the word problem. In reality, it is based on the word choice problem, that we introduced earlier. We pointed out already that the word choice problem can be much easier than the word problem. In particular, it seems unlikely that this system could ever lead to NP-completeness. Instead, $(\text{NP} \cap \text{coNP})$-completeness is more likely to be the highest difficulty that we can hope for, regarding robustness to attack. It is generally believed that $\text{NP} \cap \text{coNP}$ is a strict subclass of $\text{NP}$. Although no $(\text{NP} \cap \text{coNP})$-complete decision problem is known (see e.g., [5], page 116), it is not hard to see that for every $\text{NP}$-complete decision problem one can construct a $(\text{NP} \cap \text{coNP})$-complete premise problem. See the Appendix for details.

5. In summary: The Wagner-Magyarik cryptosystem is not a cryptosystem, but an approach towards finding new public-key cryptosystems. As a research approach it is worthwhile, however, leading to interesting (yet unsolved) problems.
3 A public-key cryptosystem based on finitely presented transformation groups

We describe a public-key cryptosystem that has some similarity with the Wagner-Magyarik system, as far as the encryption is concerned. However, we use a group whose word problem is known to be coNP-complete. The main difference is that for decryption we use the action of the group on words (instead of Wagner and Magyarik’s homomorphic image $G'$).

We pick a finitely presented group $G = (X, R)$ together with a faithful transitive action of $G$ on $\{0,1,2\}^*$ (the set of all strings over the alphabet $\{0,1,2\}$). We assume that the word problem of $G$ is coNP-complete. We conjecture that the word choice problem of $G$ is $(\text{NP} \cap \text{coNP})$-complete. The Appendix deals with a semigroup version of this question.

An example of such a group is constructed in [1], where it is called $G = \langle G_{3,1}^{(\text{mod}3)}(0,1; \#) \cup \{e_{32}\} \rangle$; it is closely related to the Higman-Thompson group $G_{3,1}$ (generalizing Richard Thompson’s infinite finitely presented simple group $G_{2,1}$). This group has the property that if two elements $g_0, g_1 \in G$ of word length $\leq n$ are different then there exists a word $z \in \{0,1,2\}^*$ of length $O(n)$ on which $g_0$ and $g_1$ act differently. Moreover, given a word $z \in \{0,1,2\}^*$ and a word $w$ over a finite generating set of $G$, the word $(z)w \in \{0,1,2\}^*$ (resulting from the action of $w$ on $z$) can be computed in time $O(|z| + |w|)$. For a definition of the Higman-Thompson groups, see also [2], [16] and [10].

**Key selection:** We first pick a word $x \in \{0,1,2\}^*$. For encrypting and decrypting 0 we choose a word $z \in \{0,1,2\}^*$ and, similarly, for 1 we choose a word $u \in \{0,1,2\}^*$; the three words $x$, $z$, $u$ should be long enough so that it is impossible to guess them. For 0, we also choose $m - 1$ “intermediary words” $z_i \in \{0,1,2\}^*$ (with $i = 1,\ldots,m - 1$); similarly, for 1 we choose $m - 1$ “intermediary words” $u_i \in \{0,1,2\}^*$ (with $i = 1,\ldots,m - 1$). The two sets $\{z\} \cup \{z_i : i = 1,\ldots,m - 1\}$ and $\{u\} \cup \{u_i : i = 1,\ldots,m - 1\}$ are required to be disjoint.

Next, we choose a “system of words” over $X \cup X^{-1}$ for encrypting a bit 0, and a system of words over $X \cup X^{-1}$ for encrypting a bit 1. A system of words (say for encrypting 0) is a sequence of $m$ finite sets $(Z_1,\ldots,Z_m)$. Each set $Z_j$ is a small set of words over $X \cup X^{-1}$ (with e.g., 4 elements). Each element $w \in Z_j$ has the property that $(z_{j-1})w = z_j$, for $j = 2,\ldots,m - 1$; also, for each element $w \in Z_1$, $(x)w = z_1$, and for each element $w \in Z_m$, $(z_{m-1})w = z$. For 1, a similar system $(U_1,\ldots,U_m)$ of sets of words is chosen, with similar properties regarding $x$, $u_j$ ($j = 1,\ldots,m - 1$), and $u$. Action diagram:

\[
\begin{array}{ccccccc}
x & \rightarrow & Z_1 & \rightarrow & Z_2 & \rightarrow & \cdots & \rightarrow & Z_{m-1} & \rightarrow & Z_m & Z_m \rightarrow & z \\
& z_1 & \rightarrow & z_2 & \rightarrow & \cdots & \rightarrow & z_i & \rightarrow & \cdots & \rightarrow & z_{m-1} & \rightarrow & z 
\end{array}
\]

The **private key** is $(x,z,u)$. (The words $z_i$ and $u_i$ are secret but are not needed for decryption.)
The **public key** consists of the presentation \((X, R)\), as well as the two set systems \((Z_1, \ldots, Z_m)\) (for 0), and \((U_1, \ldots, U_m)\) (for 1).

**Encryption:** To encrypt a bit 0, randomly choose an element \(w_j\) in each set \(Z_j\) \((j = 1, \ldots, m)\), and concatenate these elements to form the word \(w_1w_2 \ldots w_m\). Next, as in the Wagner-Magyarik system, we rewrite \(w_1w_2 \ldots w_m\) by applying the relators of \(G = (X, R)\) (as well as the trivial relators) randomly a “sufficiently large” number of times. This yields some word \(W_0\), encrypting 0.

To encrypt a bit 1, the procedure is similar, but now the set system \((U_1, \ldots, U_m)\) is used.

**Decryption:** With a ciphertext \(w\), compute \((x)w\). If \((x)w = z\), decrypt as a 0; if \((x)w = u\), decrypt as a 1.

Case of an error: If \((x)w\) is neither \(z\) nor \(u\), an error occurred (possibly as part of an attack). It is safer not to let the attacker know that the encryption is not valid, so we must produce an output in this case too. Let \(y \in \{0, 1, 2\}^*\) be a string of length similar to the lengths of \(x\), \(z\), and \(u\). We output 0 if \((x)w\) precedes \(y\) in dictionary order, and we output 1 if \((x)w\) follows \(y\) in dictionary order.

**Some design issues:**

1. The words \(x, z, u \in \{0, 1, 2\}^*\) are selected uniformly at random among words of length between \(n\) and \(2n\). In the selection process, the next letter drawn should not be the inverse of the previous letter. Here \(n\) is a security parameter; e.g., \(n = 100\). Similarly, the intermediary words are selected uniformly at random among words of length between \(n/2\) and \(4n\).

Another security parameter is \(m\); e.g., \(m = 100\).

2. How is the “system of words” \((Z_1, \ldots, Z_m)\) (and similarly \((U_1, \ldots, U_m)\)) determined? For each pair of intermediary words \((z_j, z_{j+1})\) (for 0) and \((u_j, u_{j+1})\) (for 1), we design Boolean circuits that map \(z_j\) to \(z_{j+1}\), respectively \(u_j\) to \(u_{j+1}\). These two circuits should be as similar as possible. If we want \(Z_{j+1}\) (and \(U_{j+1}\)) to have 4 elements we repeat this four times. Next, we use the correspondence between circuits and elements of the Higman-Thompson group \(G_{3,1}\) (see [1]) to construct elements of \(G\) that simulate these circuits.

3. The encryption of 0 first chooses one out of \(4^m\) elements from \(Z_1 \times \ldots \times Z_m\) (respectively from \(U_1 \times \ldots \times U_m\) for 1). The rewriting by application of relators of \(G\) then makes it hard to recognize what system of sets the chosen element \(w_1 \ldots w_m\) came from. The relators should be applied everywhere in the word, so that no local pattern from a set \(Z_j\) or \(U_j\) \((j = 1, \ldots, m)\) remains. Because of the exponential number of choices for \(w_1 \ldots w_m\), the role of the rewriting is less important than in the original Wagner-Magyarik idea. But the rewriting is nevertheless necessary, and some research is still needed to determine how (and how much of) the random rewriting should be done.

**Other groups that could be used in our public-key cryptosystem:**
The Higman-Thompson group \( G_{3,1} \) with infinite generating set \( \Delta_{3,1} \cup \{ \tau_{0,i} : i > 0 \} \), as studied in [1], could be used. This group has a finite presentation, and over this finite presentation the word problem is easy. However, over the infinite generating set \( \Delta_{3,1} \cup \{ \tau_{0,i} : i > 0 \} \) the word problem of \( G_{3,1} \) is coNP-hard. This group can be used directly to simulate circuits.

The finite symmetric group \( \mathfrak{S}_{N} \) could be used; here \( N = 2^{n} \), and \( n \) is a security parameter, e.g., \( n = 100 \). Although this group is finite, its size is exponential in the security parameter. Moreover, we believe (conjecture) that \( \mathfrak{S}_{N} \) has presentations of size linear in \( n \). We think of \( \mathfrak{S}_{N} \) as acting on bit-strings of length \( n \), hence it is natural to use elements of \( \mathfrak{S}_{N} \) for representing circuits.

4 Appendix: (NP \( \cap \) coNP)-complete premise problems

We construct an (NP \( \cap \) coNP)-complete premise problem from any NP-complete decision problem; in fact, we obtain an (NP \( \cap \) coNP)-complete word choice problem for a finitely presented semigroup. (For groups we only have a somewhat weaker result, which we will not yet present.)

Let \( S_{np} = (X,R) \) be a finitely presented semigroup with NP-complete word problem, as constructed in [3]; this presentation was derived from any nondeterministic polynomial-time Turing machine that recognizes an NP-complete language.

**Proposition.** The word choice problem of the finitely presented semigroup \( S_{np} \) above is an (NP \( \cap \) coNP)-complete premise problem.

**Proof.** Let \( L \) be any problem in NP \( \cap \) coNP. Consider a nondeterministic polynomial-time Turing machine that recognizes \( L \) and consider also a nondeterministic polynomial-time Turing machine that recognizes the complement \( \overline{L} \). Without loss of generality we can assume that these two Turing machines are actually the same Turing machine (let’s call it \( M \)), except for the accept states: \( L \) is accepted by \( M \) using accept state \( q_{1} \), and \( \overline{L} \) is accepted by \( M \) using accept state \( q_{2} \). In [3] the acceptance problem “does \( M \) accept a word \( w \) using accept state \( q_{i} \)” (for \( i = 1,2 \)) is reduced to the word problem “\( F(q_{0}w) =_{s_{np}} F(q_{i}) \)” ; here, \( q_{0} \) is the start state of \( M \), and \( F \) is a linear-time computable function from the words over the symbol set of \( M \) to the words over \( X \); \( F \) is the function that reduces the decision problem of \( M \) to word problem of \( S_{np} \). Observe that the same word \( F(q_{0}w) \) is used for both \( L \) and \( \overline{L} \). Therefore, \( w \in L \) iff \( F(q_{0}w) =_{s_{np}} F(q_{1}) \), and \( w \notin L \) iff \( F(q_{0}w) =_{s_{np}} F(q_{2}) \); hence also, \( F(q_{1}) \neq_{s_{np}} F(q_{2}) \). So, \( F \) reduces the language \( L \) to the word choice problem of the semigroup \( S_{np} \), relative to the two words \( F(q_{1}) \) and \( F(q_{2}) \). \( \square \)
5 Conclusion

The general idea for a public-key cryptosystem proposed by Wagner and Magyarik in 1984, is an interesting subject for research. The original idea is too vague to be called a cryptosystem, and it is an interesting challenge to make the idea precise in such a way as to obtain a secure system. Also, the idea needs a better analysis; in particular, it is not based on the word problem (as has been claimed so far) but on the word choice problem, which is a less difficult problem and which is related to (NP ∩ coNP)-completeness of premise problems. It seems possible to construct public-key cryptosystems based on a combination of finite presentations and transformation groups. We describe such a system, based on groups related to the Higman-Thompson groups.

References


http://www.criptored.upm.es/paginas/investigacion.htm


