On the security of two public key cryptosystems using non-abelian groups

M.I. González Vasco, D. Hofheinz, C. Martínez, and R. Steinwandt

M.I. GONZÁLEZ VASCO and C. MARTÍNEZ
Departamento de Matemáticas, Universidad de Oviedo,
c/Calvo Sotelo, s/n, 33007 Oviedo, Spain
mvasco@orion.ciencias.uniovi.es, chelo@pinon.ccu.uniovi.es

D. HOFHEINZ and R. STEINWANDT
Institut für Algorithmen und Kognitive Systeme,
Arbeitsgruppe Systemsicherheit, Prof. Dr. Th. Beth,
Universität Karlsruhe, 76128 Karlsruhe, Germany
{hofheinz, steinwan}@ira.uka.de

Keywords: Public key encryption, non-abelian groups, word problem, logarithmic signatures

Abstract

The security of two public key encryption schemes relying on the hardness of different computational problems in non-abelian groups is investigated. First, an attack on a conceptual public key scheme based on Grigorchuk groups is presented: We show that from the public data one can easily derive an 'equivalent' secret key that allows the decryption of arbitrary messages encrypted under the public key. Hereafter, a security problem in another conceptual public key scheme based on non-abelian groups is pointed out: We show that in the present form the BMW scheme is vulnerable to an attack, which can recover large parts of the private subgroup chain from the public key.

1 Introduction

During the last years, several proposals have been made to use non-abelian groups as foundation for public key schemes; however, only few of them remain unbroken. Interestingly, one of the (conceptual) schemes where still no successful attack has been published, is more than 10 years old [6]. It is due to Garzon and Zalcstein and based on the word problem in *Grigorchuk groups*. In Section 2 we demonstrate that in its present form this scheme must be considered as insecure, as it succumbs to an annoyingly simple attack, which allows to derive an 'equivalent' secret key from the public data alone. Here 'equivalent' means, that the recovered key is not necessarily

identical to the one owned by the legitimate recipient, but can nevertheless be used to decrypt arbitrary ciphertexts encrypted under the public key.

In Section 3 we analyze another interesting conceptual proposal for a public key scheme that is due to Birget, Magliveras, and Wei [1]. Their BMW-scheme is based on a special kind of factorizations of finite non-abelian groups. In contrast to earlier proposals along this line (cf. [7]), here the group structure is 'partially destroyed', and it is interesting to explore to what extent this approach helps to thwart a successful cryptanalysis. In this contribution we show that the simple method used in [1] to hide the secret subgroup chain, is not sufficient yet: We explain, how an attacker can sometimes derive large parts of the secret data from the public key.

2 A proposal of Garzon and Zalcstein

For a complete description of the cryptosystem from [6], including details about how to interpret the elements of Grigorchuk groups as permutations of the infinite complete binary tree, we refer to the original paper. Here we restrict the description to those aspects necessary for explaining our attack.

2.1 Description of the scheme

First, we recall the following facts:

- Given an infinite ternary sequence $\chi = (\chi_i)_{i\geq 1}$ over $\{0, 1, 2\}$, the *Grigorchuk group* G_χ associated with χ is an infinite 4-generator group, generated by four involutions a, b_χ, c_χ, and d_χ (the latter three of which depend on χ).

- If at least two of 0, 1, and 2 repeat infinitely often in χ, then G_χ is not finitely presentable and has subexponential growth.

- For the values of χ considered in the sequel, one can check efficiently whether a word $\mathbf{w} \in \{a, b_\chi, c_\chi, d_\chi\}^N$ represents the identity in G_χ, if the first $\lceil \log_2(N) \rceil$ positions of χ are known.

With this, the Garzon-Zalcstein public key cryptosystem can be sketched as follows:

Private key: Alice chooses an infinite ternary sequence χ such that both G_χ is not finitely presentable and the word problem in G_χ can be solved efficiently. More precisely, her private key is a *Turing machine* M_{G_χ} solving the word problem in the Grigorchuk group G_χ.

Public key: The corresponding public information consists of a finite subset of relators R of G_χ (i. e., some words in the generators $a, b_\chi, c_\chi, d_\chi$ that are equal to the identity in G_χ), and two words $\mathbf{w}_0, \mathbf{w}_1$ in the generators $a, b_\chi, c_\chi, d_\chi$ representing distinct group elements $\omega_0, \omega_1 \in G_\chi$.

Encryption: Plaintexts are encrypted bit-wise. To encrypt a single message bit $b \in \{0,1\}$, Bob starts with the word $\mathbf{c} = \mathbf{w}_b$ and repeatedly applies some of the public relators R to $\mathbf{c}$, i. e. he inserts words $\mathbf{r} \in R$ into $\mathbf{c}$ and/or removes occurrences of words in R from $\mathbf{c}$. After having done this a couple of times (in [6] no precise specification is given here), he stops and transmits the resulting ciphertext $\mathbf{c} \in \{a, b_\chi, c_\chi, d_\chi\}^*$ to Alice.

Decryption: To decrypt a ciphertext $\mathbf{c} \in \{a, b_\chi, c_\chi, d_\chi\}^*$ to a plaintext bit b, Alice uses her secret Turing maching M_{G_χ} for checking whether in G_χ the equality $c = \omega_0$ or $c = \omega_1$ holds (where c is the element of G_χ represented by $\mathbf{c}$). Due to the choice of χ, this can be checked efficiently: If $\mathbf{c}$ is the result of an encryption as described above, it is sufficient to check with M_{G_χ} whether $\mathbf{w}_0^{-1}\mathbf{c}$ represents the identity in G_χ.

2.2 Cryptanalysis

In [6] the assumption is adopted that verifying the correctness of a guess for the first few positions of χ is difficult, because of Alice's public key revealing only the finite subset R of relators of (the not finitely presentable group) G_χ. However, an attacker can exploit that only the public relators R are used during encryption:

Let $N := \max\{|\mathbf{r}| : \mathbf{r} \in R \cup \{\mathbf{w}_0^{-1}\mathbf{w}_1\}\}$ be the maximum of the lengths of the 'relational words' in Alice's public key and of the word $\mathbf{w}_0^{-1}\mathbf{w}_1$, where $\mathbf{w}_0, \mathbf{w}_1$ are Alice's two distinct public words. Then, having in mind a sensible size of the public key, we must assume that the number $B := \lceil \log_2 N \rceil$ is rather small and that a brute-force search over all 3^B words in $\{0,1,2\}^B$ is feasible. Let us denote by $\bar{\chi}$ the finite sequence consisting of the first B elements of the secret χ. Then, an attacker can use the following strategy:

1. For each possible choice of the first B positions of χ, he can check whether both each element of R vanishes and $\mathbf{w}_0^{-1}\mathbf{w}_1$ does not vanish in G_χ. By construction, at least one possible choice must satisfy these conditions (for $\bar{\chi}$ does).

2. Once such a finite sequence is found, he can complete it in an arbitrary suitable manner (e. g. by appending zeroes) to an infinite ternary sequence Ψ. The public key depends on $\bar{\chi}$ only, i. e., the public key could have been derived from Ψ as well as from χ. Thus, a Turing maching solving the word problem in G_Ψ now correctly decrypts all ciphertexts encrypted under Alice's public key.

Consequently, in its present form, the proposal in [6] must be considered as insecure. We are not aware of possible modifications of the scheme that may help to achieve greater cryptographic security, and it remains unclear

if the complexity of the word problem in Grigorchuk groups can be used for cryptographic applications.

3 A proposal of Birget, Magliveras, and Wei

Recently, another public key scheme based upon a computational problem in non-abelian groups has been introduced [1]. In contrast to the above scheme of Garzon and Zalcstein, the proposal of Birget, Magliveras, and Wei which we discuss now is based on the use of *finite* non-abelian groups. The BMW-scheme is still rather conceptual, and no concrete parameters have been proposed so far. Thus, our discussion can be understood as an attempt to recognize certain weak parameter choices.

Similarly, as in the previous section, we recall only those aspects of BMW, which are relevant for our discussion; a more complete description of the scheme can be found in the original paper [1].

3.1 Description of the scheme

Analogousy as in the MST_1 public key scheme [7]—a group based proposal of Magliveras, Stinson, and van Trung—the main tool of BMW is a certain kind of group factorization called *logarithmic signature* :

Definition 3.1 *Let G be a finite group, and $A_1, \ldots, A_s$ finite sequences over G, i. e., $A_i = [\alpha_{i,1}, \ldots, \alpha_{i,r_i}]$ with $r_i \in \mathbb{N}_0$ and $\alpha_{i,j} \in G$ ($1 \leq j \leq r_i$, $1 \leq i \leq s$). Then the sequence $A := [A_1, \ldots, A_s]$ is a* logarithmic signature *for G if and only if for each $g \in G$ there is a unique factorization*

$$g = \alpha_{g,1} \cdot \ldots \cdot \alpha_{g,s} \tag{3.1}$$

with $\alpha_{g,i} \in A_i$ ($1 \leq i \leq s$).

In BMW, permutation groups along with a special kind of logarithmic signatures are used. Namely, these logarithmic signatures are derived from a chain of subgroups, and there are efficient algorithms for computing the factorization (3.1) with respect to such a logarithmic signature (see, for instance, [8]). This is crucial for decrypting ciphertexts in BMW. To illustrate the construction, assume we are given a permutation group G along with a chain of subgroups

$$G = G_0 > \cdots > G_{s+2} = \{\mathrm{id}\} \quad (\text{where } s > 2).$$

Now fix a partition $\{1, \ldots, s+1\} = I \uplus J$, say $I = \{i_1, \ldots, i_x\}$ ($x \geq 2$) with $i_1 < \cdots < i_x$ and $J = \{j_1, \ldots, j_{s+1-x}\}$ with $j_1 < \cdots < j_{s+1-x}$. Then for each $i \in I$ we fix a right transversal R_i of G_i in G_{i-1} containing the identity id, and for each $j \in J$ we choose a left transversal L_j of G_j in G_{j-1} containing the identity id. That is, the group G can be factored as

$$G = L_{j_1} L_{j_2} \cdots L_{j_{s+1-x}} G_{s+1} R_{i_x} \cdots R_{i_2} R_{i_1}.$$

Now we obtain a logarithmic signature $A := [A_1, \ldots, A_s]$ for G by setting

- $A_k := L_{j_k} \quad (1 \leq k \leq s - x)$,
- $A_{s+1-x} := L_{j_{s+1-x}} \cdot G_{s+1}$,
- $A_k := R_{i_{s+2-k}} \quad (s + 2 - x \leq k \leq s - 1)$, and
- $A_s := R_{i_2} \cdot R_{i_1}$.

Here, the products $L_{j_{s+1-x}} \cdot G_{s+1}$ and $R_{i_2} \cdot R_{i_1}$ are actually Kronecker/tensor products of matrices when we interpret finite sequences as single-row matrices; hence, A_{s+1-x} and A_s are again finite (ordered) sequences. If the subgroup chain and the sets I, J along with the respective transversals are known, then factoring a permutation $g \in G$ with respect to such a logarithmic signature provides no algorithmic difficulties. Basically, in BMW the mentioned parameters form the

Private key: A subgroup chain with transversals as described—the additional requirements imposed in [1] are not important for our discussion. Also, variations are possible: The subgroup G_{s+1} can be *melted* with a right instead of a left transversal, and instead of melting the 'last' two transversals R_{i_2}, R_{i_1} we could melt the 'first' ones L_{j_1}, L_{j_2}. For the sequel, it is sufficient to assume that—up to two exceptions—each A_k is a transversal as described above; one 'exceptional' block is obtained by melting G_{s+1} with a transversal to its left or right, and the other 'exceptional' block is $A_1 = L_{j_1} \cdot L_{j_2}$ or $A_s = R_{i_2} \cdot R_{i_1}$ (depending on whether $j_1 = 1$ or $i_1 = 1$).

Public key: With the exception of a single block, the complete logarithmic signature A for G is made public. Namely, the public key is the sequence $B := [B_1, \ldots, B_s]$ where $B_k := A_k$ for all but one value of k: Instead of the block obtained by melting G_{s+1} with the transversal to its left or right, only the respective transversal is published. Thus, for $A_{s+1-x} = L_{j_{s+1-x}} \cdot G_{s+1}$ we set $B_{s+1-x} := L_{j_{s+1-x}}$, and analogously when G_{s+1} has been melted with a right transversal. In summary, the subgroup G_{s+1} is 'cut out', which in particular means that only a subset Γ of the elements of G can be factored with respect to B.

Encryption: Plaintexts are regarded as elements of $M := \mathbb{Z}_{|B_1|} \times \cdots \times \mathbb{Z}_{|B_s|}$, where $|B_k|$ denotes the size of B_k. Thus, a plaintext $m \in M$ can be taken for a pointer that in each block B_k determines a unique element $\beta_{k,m} \in B_k$ $(1 \leq k \leq s)$. Now, the ciphertext computes to $c := \beta_{1,m} \cdot \ldots \cdot \beta_{s,m} \in \Gamma \subseteq G$.

Decryption: Knowing the subgroup chain $G = G_0 > \cdots > G_{s+2} = \{\mathrm{id}\}$, as well as I, J, and A, factoring a ciphertext c with respect to B provides no algorithmic difficulties.

3.2 An attack

As already pointed out in [1], the removal of the subgroup G_{s+1} in the public key is crucial: Assume we had $A_{s+1-x} = L_{j_{s+1-x}} \cdot G_{s+1}$ and $B_{s+1-x} = L_{j_{s+1-x}}$. Then the attacker would know that the public key contains two consecutive blocks B_l, B_{l+1} with $B_l \cdot B_{l+1}$ being a group (namely $B_l \cdot B_{l+1} = G_{s-1}$). For permutation groups, we can (with some luck) efficiently locate these two blocks by simply computing the size of the group generated by the elements in consecutive blocks. Once B_l and B_{l+1} are identified, the attacker knows that either $B_{l-1} \cdot B_l B_{l+1}$ or $B_l B_{l+1} \cdot B_{l+2}$ is a group, which can again be checked efficiently by computing the size of the group spanned by the elements in these three blocks. Continuing in this manner, large parts of the secret sets I, J, and the subgroup chain can be revealed. To avoid this attack, the group G_{s+1} is not part of the public key. However, for certain groups G this might not be sufficient for hiding G_{s+1} from the attacker, as quite some information on G_{s+1} is public. Namely, if the attacker learns G, then he can try to exploit that

- $|G_{s+1}| = |G| / \prod_{k=1}^{s} |B_k|$
- $G_{s+1} \cap B_k = \{\mathrm{id}\}$ for $k = 1, \ldots, s$
- For some $k \in \{1, \ldots, s\}$, $B_k G_{s+1}$ or $G_{s+1} B_k$ is a group.

It can well be the case that this information suffices for identifying the group G_{s+1} and thus the attack just described applies. But even if G_{s+1} cannot be found, an adaption of the attack may work:

When dealing with permutation groups of degree n, a subgroup chain can be no longer than $\lceil 3n/2 \rceil - b(n) - 1$ where $b(n)$ is the number of 1s in the binary representation of n [4]. Thus, having in mind practical key sizes, we must assume that the attacker can guess the position of the block that contains the (right or left) transversal of G_{s+1} in G_s. Actually, we can simply apply the subsequently described steps to all blocks in parallel and discard all those where an 'impossible situation' is encountered.

Say, the block with the transversal of G_{s+1} in G_s is $B_{s+1-x} = L_{j_{s+1-x}}$. Depending on the (left) transversal used, now it may well happen that $L_{j_{s+1-x}}$ *generates* G_s. Clearly, we cannot expect this to happen in all cases. Nevertheless, it seems plausible to assume that this situation occurs 'too often', when the block B_{s+1-x} has not specifically been designed to dodge this problem: Permutation groups require quite 'few' generators. Namely, from [9] we know that a permutation group of degree $n > 3$ can always be generated with $\leq \lfloor n/2 \rfloor$ generators, and for certain groups (like the symmetric ones) this bound is far from tight. Further on, in [5] it is shown that if $\beta > 1/2$ and n is large enough, then $\lfloor \beta n \rfloor$ randomly chosen elements of a permutation group of degree n almost certainly form a system of generators. On the one hand, modeling the transversals in the public key of BMW as

randomly chosen elements is certainly not appropriate; in particular, special requirements on such a transversal will be imposed. On the other hand, it is interesting to note that the requirement of using *anticlosed* transversals (considered in [1]) supports the attacker in the sense, that it prohibits 'superfluous' generators:

Definition 3.2 *A subset S of a group G is* anticlosed *in G if and only if $u, v \in S \setminus \{1\}$ implies that $uv \notin S \setminus \{1\}$.*

Of course, once we know the subgroup G_s, we can proceed exactly as described in the first paragraph of this section to recover a large part of the secret subgroup chain and the secret sets I, J. Moreover, there is a trivial extension of this approach: If the public transversal of G_{s+1} in G_s does not generate G_s, it may still happen that two consecutive blocks form a generating system for G_{s-1}. Thus in this case much of the secret subgroup chain can leak, too.

As in [1] no concrete parameter choices are proposed, we illustrate the above attack with the setting in [1, Example 1]:

Example 3.3 *Let G be some permutation group, $s := 6$, $I := \{1, 3, 7\}$, $J := \{2, 4, 5, 6\}$, and consider the subgroup chain*

$$G = G_0 > G_1 > \cdots > G_6 > G_7 > G_8 = \{\mathrm{id}\}.$$

Thus, we have three right transversals R_1, R_3, R_7 and four left transversals L_2, L_4, L_5, L_6, which induce the following factorizations:

$$\begin{array}{ll} G_0 = G_1 R_1 & G_4 = L_5 G_5 \\ G_1 = L_2 G_2 & G_5 = L_6 G_6 \\ G_2 = G_3 R_3 & G_6 = G_7 R_7 \\ G_3 = L_4 G_4 & \end{array}$$

Combining these, we obtain

$$\begin{aligned} G &= (L_2((L_4(L_5(L_6(G_7R_7))))R_3))R_1 \\ &= (L_2)(L_4)(L_5)(L_6G_7)(R_7)(R_3R_1) \\ &=: A_1A_2A_3A_4A_5A_6. \end{aligned}$$

Basically, from the logarithmic signature $[A_1, \ldots, A_6]$, we now obtain a public key for BMW by setting $[B_1, \ldots, B_6] := [A_1, A_2, A_3, L_6, A_5, A_6]$, i. e., the subgroup G_7 is 'cut out'.

If $B_5 = R_7$ happens to form a generating system for G_6, and the attacker has (by guessing or exhaustive search) identified this block, then he can proceed as follows to reveal almost the complete secret subgroup chain:

1. *Compute the size of the subgroups generated by B_4G_6 and G_6B_6. Now check which of B_4G_6 and G_6B_6 forms a group by comparing the respective sizes of the generated subgroups to that of B_4G_6 resp. G_6B_6. This should reveal the left transversal $B_4 = L_6$ of G_6 in G_5, so that the attacker knows a generating system for $G_5 = \langle B_4, B_5 \rangle$.*
2. *Analogously, by checking which of B_3G_5 and G_5B_6 is a subgroup, the attacker should identify the left transversal L_5 of G_5 in G_4.*
3. *Next, by checking which of B_2G_4 and G_4B_6 is a subgroup, finding the left transversal L_4 of G_4 in G_3 should be possible.*

Now we expect that neither B_1G_3 nor G_3B_6 is a subgroup, and the above approach gets 'stuck'. With some luck, an exhaustive search over the subsets of B_6 allows to recover R_3—and thereafter L_2 and R_1. But even if this is not possible, a significant part of the secret key has leaked.

To hamper cryptanalysis, [1] proposes to impose additional requirements on the public blocks. For this aim, the notion of *anticlosedness* (see Definition 3.2) is introduced, and in the next section, we want to look at this concept in more detail.

3.3 On using anticlosed blocks

When constructing a logarithmic signature by simply juxtaposing left or right transversals along a subgroup chain, there is one block whose elements form a group. As factoring elements with respect to such an *exact transversal logarithmic signature* can be done efficiently, it is intuitively desirable that the public key B is 'as far away as possible' from a transversal logarithmic signature. This problem is well-known from the MST_1 public key cryptosystem, and one may ask how to recognize 'good' public keys, where effectively factoring group elements with respect to the public blocks is algorithmically hard.

As a possible criterion, in [7] the notion of being *totally-non-transversal* has been introduced. However, in [2] it has been proven that a logarithmic signature A for a permutation group G can be both totally-non-transversal and *tame*, i. e., allow for a polynomial time algorithm for factoring elements of G with respect to A. In fact, the next proposition illustrates with two examples, that for anticlosedness the situation is similar:

Proposition 3.4 *Let G be a symmetric group or a cyclic 2-group. Then there exists an exact transversal logarithmic signature $A = [A_1, \ldots, A_s]$ for G such that each block A_i ($1 \leq i \leq s$) is anticlosed in G.*

Proof. For the symmetric group S_n we can use the exact transversal logarithmic signature $A := [A_2, \ldots, A_n]$ where

$$A_m := [(m,1), (m,2), \ldots, (m,m-1), \mathrm{id}] \quad (2 \leq m \leq n)$$

is a right transversal of S_{m-1} in S_m. Verifying anticlosedness is straightforward: Let $(m, i_1), (m, i_2) \in A_m \setminus \{\mathrm{id}\}$ with $1 \leq i_1, i_2 \leq m-1$. Then $(m, i_1)(m, i_2) \in \{(m, i_2, i_1), \mathrm{id}\}$, and therefore $(m, i_1)(m, i_2) \notin A_m \setminus \{\mathrm{id}\}$.

Now let $G \simeq \mathbb{Z}/2^n\mathbb{Z}$ be a(n additively written) cyclic 2-group. W. l. o. g. we may assume $n > 0$. Then from the subgroup chain

$$G = \underbrace{\langle 2^0 \rangle}_{\simeq \mathbb{Z}/2^n\mathbb{Z}} > \underbrace{\langle 2^1 \rangle}_{\simeq \mathbb{Z}/2^{n-1}\mathbb{Z}} > \cdots > \underbrace{\langle 2^{n-1} \rangle}_{\simeq \mathbb{Z}/2^1\mathbb{Z}} > \{0\}$$

we derive the following exact transversal logarithmic signature for G:

$$A := [[0, 2^0], [0, 2^1], \ldots, [0, 2^{n-1}]]$$

For violating anticlosedness, the relation $2^m + 2^m \equiv 2^m \mod 2^n$ had to hold for some $m \in \{0, \ldots, n-1\}$. In other words, we had $2^n \mid 2^m$ for some $m \in \{0, \ldots, n-1\}$, which is impossible. ■

It should be stressed that this proposition does not imply, that factoring with respect to a logarithmic signature is easy, provided that all its blocks are anticlosed. However, it illustrates that the property of being comprised of anticlosed blocks does not imply that a logarithmic signature is 'far from being transversal'. It would be interesting to know whether there is a family of finite groups where

1. logarithmic signatures comprised of anticlosed blocks can be computed efficiently, and

2. only a negligible fraction of these logarithmic signatures is tame.

Such a family could provide a starting point for building a key generation procedure for a secure MST_1- or BMW-like scheme. At the moment, it remains unclear whether it is possible to use logarithmic signatures as a foundation of a practical and secure public key scheme. In particular, our discussion illustrates some obstacles which have to be taken into account when trying to derive secure instances of the BMW-scheme proposed in [1].

4 Conclusion

The above cryptanalysis of the Grigorchuk group based public key cryptosystem from [6] shows that, in its present form, this scheme must be considered as insecure: The described attack allows an attacker to decrypt arbitrary ciphertexts encrypted under the public key. Nonetheless, it is an inspiring research topic to try to exploit well studied instances of the word problem in finitely presented groups for cryptographic applications.

Moreover, we have pointed out a potential security problem in the BMW scheme proposed in [1]. While the described attack does not 'break' the

general scheme, it points out new obstacles that have to be dealt with for deriving concrete instances of BMW. It remains an interesting challenge to explore the potential of logarithmic signatures for finite groups as a basis for practical and secure public key encryption schemes.

References

[1] Jean-Camille Birget, Spyros S. Magliveras, and Wandi Wei. Trap doors from subgroup chains and recombinant bilateral transversals. In Santos González and Consuelo Martínez, editors, *VII Reunión Española de Criptología y Securidad de la Información, Proceedings of RECSI 2002*, pages 31–48, 2002.

[2] Jens-Matthias Bohli, María Isabel González Vasco, Consuelo Martínez, and Rainer Steinwandt. Weak keys in MST_1. IACR ePrint Archive, May 2002.

[3] Peter J. Cameron. Some measures of finite groups related to permutation bases. Unpublished, 2003.

[4] Peter J. Cameron, Ron Solomon, and Alexandre Turull. Chains of subgroups in symmetric groups. *Journal of Algebra*, 127:340–352, 1989.

[5] Eloisa Detomi, Andrea Lucchini, and Fiorenza Morini. How many elements are needed to generate a finite group with good probability? *Israel Journal of Mathematics*, 132:29–44, 2002.

[6] Max Garzon and Yechezkel Zalcstein. The complexity of Grigorchuk groups with application to cryptography. *Theoretical Computer Science*, 88(1):83–98, 1991.

[7] S. S. Magliveras, D. R. Stinson, and Tran Van Trung. New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups. *Journal of Cryptology*, 15(4):285–297, 2002.

[8] Spyros S. Magliveras and Nasir D. Memon. Properties of cryptosystem PGM. In Gilles Brassard, editor, *Advances in Cryptology, Proceedings of CRYPTO '89*, number 435 in Lecture Notes in Computer Science, pages 447–460. Springer-Verlag, 1990.

[9] A. McIver and P. M. Neumann. Enumerating finite groups. *Quarterly Journal of Mathematics*, 38(2):473–488, 1987.

[10] Neal R. Wagner and Marianne R. Magyarik. A public key cryptosystem based on the word problem. In G. Robert Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84*, number 196 in Lecture Notes in Computer Science, pages 19–36. Springer-Verlag, 1985.