Contemporary Mathematics

An attack on a group-based cryptographic scheme

Dennis Hofheinz and Dominique Unruh

Abstract. We give an attack on a public key encryption scheme suggested by Shpilrain and Zapata. Experimental evidence shows that this attack is practical and works for the proposed parameters. We give a way to repair the encryption scheme so that our attack does not work anymore. However, we also expose weak points of the scheme that do not seem to be repairable in an obvious manner.

1. Introduction

Within the last years various attempts have been made to derive cryptographic primitives from problems originating in combinatorial group theory (see, e. g., [**Wag84, WM85, GZ91, AAG99, KLC+00, AAFG01, Shp04**]). As a relatively new approach, [**Shp04, SZ04**] propose a public key cryptosystem based on metabelian groups. They claim that the security of their scheme is based on the subgroup membership problem in the considered metabelian group.

In this contribution, we show that their scheme can be broken by a very efficient heuristic attack that bypasses solving the subgroup membership problem. This attack uses that public key and ciphertext are transmitted as elements of a free group instead of the considered metabelian group. With the original scheme, this was necessary to allow for en- and decryption. We give a fix to this that allows to have at least the ciphertext transmitted as an element of a metabelian group. With this modification, our attack does not work anymore. But since even with our fix, the public key has to be transmitted as an element of a free group, a reduction to the subgroup membership problem in a metabelian group seems not directly possible. Furthermore, we also expose some additional weak points of the scheme that are also present in the fixed version and do not seem to be easily removable. At the moment, this does not constitute a complete break of the repaired scheme in the sense of a successful attack, but indicates that further research might be needed with respect to some parameters and possibly the proposed platform group.

1991 *Mathematics Subject Classification.* Primary 94A60; Secondary 20F36.

Key words and phrases. Public key cryptography, metabelian groups.

Most of this work was done while the first author was with the Institut für Algorithmen und Kognitive Systeme (IAKS), Lehrstuhl Prof. Beth at the Universität Karlsruhe.

Note. This paper refers to the version [**Shp04, SZ04**] of the Shpilrain-Zapata cryptosystem that was presented at the Canadian Mathematical Society Winter Meeting in December 2004. After presentation of our attack on that system at [**Hof05**], however, the preprint [**SZ04**] was updated (see [**SZ06**]) with improvements very similar to our suggestions, so that the original, unmodified version [**SZ04**] of the Shpilrain-Zapata system is no longer available online. Consequently, our attack from Section 3 does not apply to the updated system [**SZ06**]; however, the observation of weak points in Section 5 does.

2. The Shpilrain-Zapata Cryptosystem

Here, we present the public key cryptosystem of [**Shp04**]. To ease the things to come, we do so in a slightly different (but equivalent) form.

2.1. The System. First, we describe the system on an abstract level, and in the next subsection, we then discuss some parameter suggestions made in [**Shp04, SZ04**].

Let F_{n+m} be a free group with free generators $x_1, \ldots, x_{n+m}$. Let $R \triangleleft F_{n+m}$ be a normal subgroup of F_{n+m} that is invariant under arbitrary F_{n+m}-endomorphisms. Then $\mathcal{F}_{n+m} := F_{n+m}/R$ is called *relatively free*.

So let for fixed $n, m \in \mathbb{N}$ such $F_{n+m}, R, \mathcal{F}_{n+m}$ be given. Computationally, here any element from $\mathcal{F}_{n+m}$ is given by a free representative (i.e., a word in the free generators x_i and their inverses). Furthermore, any endomorphism $\alpha \in \mathrm{End}(\mathcal{F}_{n+m})$ is given by the images of the generators $x_1R, \ldots, x_{n+m}R$ under α, so in fact α is represented by a vector of $n+m$ elements from F_{n+m}.

In particular, say that two endomorphisms $\alpha, \beta \in \mathrm{End}(\mathcal{F}_{n+m})$ are given in that way by vectors $(\alpha_1, \ldots, \alpha_{n+m}), (\beta_1, \ldots, \beta_{n+m}) \in F_{n+m}^{n+m}$ such that, e.g., α_i is a free representant of $\alpha(x_iR)$. Then it is clear how to implement the composition $\alpha \circ \beta$: simply substitute every occurence of x_j (resp. x_j^{-1}) in every β_i with α_j (resp. α_j^{-1}).

Key generation: Choose $\varphi \in \mathrm{Aut}(\mathcal{F}_{n+m})$ together with its inverse φ^{-1} such that φ^{-1} cannot be efficiently deduced from φ. (How such a φ is chosen depends on the concrete choice of the underlying subgroup, cf. [**SZ06**].) The public key for encryption is $\hat{\varphi} := \pi_n \circ \varphi$, where $\pi_n \in \mathrm{End}(\mathcal{F}_{n+m})$ is the projection onto the first n generators (i.e., $\pi_n(x_iR) = x_iR$ for $1 \leq i \leq n$ and $\pi_n(x_iR) = 1$ for $n < i \leq n+m$). The secret key for decryption is φ^{-1}.

Encryption: A plaintext is an endomorphism $w \in \mathrm{End}(\mathcal{F}_{n+m})$ satisfying $w(x_iR) = 1$ for $n < i \leq n+m$ (such that $w \circ \pi_n = w$). Any such w is encrypted as $c := w \circ \hat{\varphi} \in \mathrm{End}(\mathcal{F}_{n+m})$.

Decryption: Decrypting of some $c \in \mathrm{End}(\mathcal{F}_{n+m})$ is done by $w' := c \circ \varphi^{-1}$ such that in case of a legitimately generated ciphertext $c = w \circ \hat{\varphi} = w \circ \pi_n \circ \varphi = w \circ \varphi$, it holds $w' = c \circ \varphi^{-1} = w \circ \varphi \circ \varphi^{-1} = w$ for the decrypted plaintext.

There is a fine point concerning decryption: as φ is not necessarily an automorphism of F_{n+m}, and the equation $\varphi \circ \varphi^{-1} = \mathrm{id}$ does not necessarily hold over F_{n+m}, the free representants of original plaintext and decrypted ciphertext may differ (although they represent the same elements in $\mathcal{F}_{n+m}$). Since as discussed we represent $\mathcal{F}_{n+m}$-endomorphisms by free group elements, the decrypted ciphertext

must eventually be put into an $\mathcal{F}_{n+m}$-normal form. So then, actually only the normal form of w (but not its free representation) can be transmitted.

As a remark on this, there is no need to be able to interpret any given plaintext message as a suitable normal form of an element from $\mathcal{F}_{n+m}$. It suffices to be able to choose normal forms in a random way and then to encrypt the actual plaintext with that randomness as a one-time-pad. More specifically, one could encrypt bitstrings m with $m \mapsto (c, H(w) \oplus m)$, where $c = w \circ \varphi$ for a random w, H is a hash function that maps vectors of normal forms to bitstrings, and $\oplus$ denotes the bitwise XOR. (Of course, this simple construction results in a highly malleable scheme [**DDN91**], but then again, efficient constructions are known for converting such a scheme into a non-malleable one, see, e.g., [**FO99**].)

2.2. Suggested Parameters. In [**Shp04, SZ04**], the following parameters were suggested for implementing the above system. First, take $n = 8$ and $m = 2$, such that $F_{n+m} = F_{10}$ is the free group of rank 10. Let $R = [[F_{10}, F_{10}], [F_{10}, F_{10}]]$, such that $\mathcal{F}_{n+m} = F_{n+m}/R$ is the free metabelian group of rank 10.

We omit a description of the way $\hat{\varphi}$ is chosen, as this is not important for our attack. However, it is worthwhile to describe the $\mathcal{F}_{n+m}$-normal form employed during decryption. Namely, it is suggested to use the following normal form $\mathrm{NF}(z)$ for a free representant $z \in F_{n+m}$ of an element from $\mathcal{F}_{n+m}$. First, consider z as an element of the group ring $\mathbb{Z}F_{n+m}$. Let $\mathrm{Fox}_i(z)$ denote the partial Fox derivative with respect to x_i.[1] Let $\mathbf{Fox}(z) = (\mathrm{Fox}_i(z))$ be the vector of the $n + m$ partial Fox derivatives of z. Then $\mathrm{NF}(z)$ is simply the component-wise abelianization of $\mathbf{Fox}(z)$, i.e., $\mathrm{NF}(z) = \left(\overline{\mathrm{Fox}_i(z)}\right)$ where $\overline{a}$ denotes the abelianization of a.

3. Cryptanalysis

Let $\mathcal{F}_{n+m}$ be as before, and let $\hat{\varphi} \in \mathrm{Aut}(\mathcal{F}_{n+m})$ be a public key. Our goal is to decipher a given ciphertext $c = w \circ \hat{\varphi} = w \circ \varphi$. We proceed in two steps: first, we derive the abelianized version $\overline{w}$ of w. Second, we give a heuristic algorithm that, using $\overline{w}$, outputs w with high probability.

At the end of the section, we also give experimental evidence that our approach works. We would like to emphasize that our attack works completely over the free group F_{n+m} and in particular does not solve a subgroup membership problem in a metabelian group. This shows that the system of [**Shp04, SZ04**] is not (at least not solely) based on such a subgroup membership problem, in contrast to what is implied by the title of [**SZ04**].

3.1. The Abelian Part. The crucial observation is that since all computations (apart from the postprocessing of the decrypted ciphertext) take place over the free group F_{n+m}, we know the abelianization of all transmitted group elements. (Note that the abelianization of an element of $\mathcal{F}_{n+m}$ is well-defined, since $R \leq [F_{n+m}, F_{n+m}]$.) So we can assume the abelianizations $\overline{c}$ and $\overline{\hat{\varphi}}$ of c and $\hat{\varphi}$ to be known, and are looking for the abelianization $\overline{w}$ of w. Now the abelianization of $\mathcal{F}_{n+m}$ is isomorphic to $\mathbb{Z}^{n+m}$. Hence, any endomorphism $\overline{\phi}$ of the abelianized $\mathcal{F}_{n+m}$ can be seen as an endomorphism of $\mathbb{Z}^{n+m}$, i.e., as an $(n + m) \times (n + m)$ matrix over $\mathbb{Z}$ acting by left-multiplication on $\mathbb{Z}^{n+m}$. Concretely, the columns of

[1] The partial Fox derivative with respect to x_i is the map on $\mathbb{Z}F_{n+m}$ defined by the recursion $\mathrm{Fox}_i(ab) = \mathrm{Fox}_i(a) + a\mathrm{Fox}_i(b)$, $\mathrm{Fox}_i(x_i) = 1$ and $\mathrm{Fox}_i(x_j) = 0$ for $i \neq j$.

this matrix are simply the images of the free abelian generators of under $\overline{\phi}$. In the following, we will thus consider the abelianized versions $\overline{c}, \overline{\hat{\varphi}}, \overline{w}$. of the endomorphisms $c, \hat{\varphi}, w$ as $(n+m) \times (n+m)$ matrices over $\mathbb{Z}$. The set of these matrices we write as $\mathbb{Z}^{(n+m)\times(n+m)}$. By $c = w \circ \hat{\varphi}$ it follows that

$$\overline{c} = \overline{w} \cdot \overline{\hat{\varphi}} \tag{3.1}$$

Further note that the abelianization $\overline{\pi}_n \in \mathbb{Z}^{(n+m)\times(n+m)}$ of π_n is the diagonal matrix with 1 on its first n diagonal elements and 0 on the remaining m diagonal elements. Since $\overline{w} = \overline{w} \cdot \overline{\pi}_n$, it follows that the last m columns of $\overline{m}$ are zero. And since $\overline{\hat{\varphi}} = \overline{\pi}_n \cdot \overline{\hat{\varphi}}$, the last m rows of $\overline{\hat{\varphi}}$ are zero, too. So we can w.l.o.g. consider $\overline{w}$ to be in $\mathbb{Z}^{(n+m)\times n}$ and $\overline{\hat{\varphi}}$ in $\mathbb{Z}^{n\times(n+m)}$. Then (3.1) is an overdetermined system of linear equations. Further, since φ is an automorphism, $\operatorname{rank}\overline{\varphi} = n+m$, and thus $\operatorname{rank}\overline{\hat{\varphi}} = \operatorname{rank}\overline{\pi}_n \cdot \overline{\varphi} = n$. So as an element of $\mathbb{Z}^{n\times(n+m)}$ has full rank and (3.1) has a unique solution $\overline{w}$ which is this original plaintext. This unique solution can then efficiently be found using Gaussian elimination (over $\mathbb{Q}$).

So in summary, we can easily obtain the abelianized version $\overline{w}$ of the plaintext from the public key $\hat{\varphi}$ and an encryption c of w alone.

3.2. The Non-Abelian Part. Let's have a closer look at the encryption operation. Encryption consists of computing $w \circ \hat{\varphi}$ (for public $\hat{\varphi}$) simply as a substitution. More specifically, say that w is given as $(w_1, \ldots, w_{n+m}) \in F_{n+m}^{n+m}$, and $\hat{\varphi}$ as $(\hat{\varphi}_1, \ldots, \hat{\varphi}_{n+m}) \in F_{n+m}^{n+m}$.

Then c is computed as $(c_1, \ldots, c_{n+m}) \in F_{n+m}^{n+m}$, where

$$c_i = \hat{\varphi}_i|_{x_j \to w_j} \tag{3.2}$$

(which means that c_i is equal to $\hat{\varphi}_i$, only that every occurence of any x_j is substituted with the corresponding w_j).

Now say that $\hat{\varphi}_1$ starts with x_j. Then (3.2) means that c_1 starts with w_j (modulo cancellations in the free group). A very simple approach might now be to try to "read off" w_j from the head of c_1. The problems are that (a) cancellations might have taken place and "corrupted" parts of w_j, and (b) there is no telling where w_j finishes and the next $w_{j'}$ starts.

We deal with those two problems by searching different $\hat{\varphi}_i$ that all start with the same generator, e.g., x_j. Although not necessarily the case, the greatest matching prefix of the corresponding c_i can be expected to be a prefix of w_j. Also, such potential prefixes can be found by looking at the tails of c_i for which $\hat{\varphi}_i$ ends on x_j^{-1}. Similarly, one can find potential suffixes of some w_j by looking at the tails of c_i where $\hat{\varphi}_i$ ends with x_j, or at the heads of c_i where $\hat{\varphi}_i$ starts with x_j^{-1}.

As soon as a potential prefix w_j^1 and a suffix w_j^2 of some w_j is found, it can be tried to put w_j together completely. Namely, one can try to chop off generators from the tail of w_j^1 and/or the head of w_j^2 until $w_j^1 w_j^2$ has the correct abelianization $\overline{w}$. (Recall that the previous section shows how to acquire $\overline{w}$.)

Then, as soon as a good candidate for w_j is found, (a) any x_j or x_j^{-1} at the head or the tail of an $\hat{\varphi}_i$ can be eliminated, and (b) the corresponding c_i has to be modified accordingly (i.e., has to be multiplied with the candidate w_j^{-1} or w_j). This yields a simplified system of equations of the type (3.2), in which all $x_j^{\pm 1}$ have been eliminated from the heads and tails of the $\hat{\varphi}_i$. The method described can then be iterated.

Of course, this method is heuristic and heavily relies on the assumption that not too many cancellations in the free group take place. The next subsection gives evidence that nonetheless, our method can be used to successfully attack the system.

3.3. Experimental Results. We have implemented the system in C++ on a standard PC, using the parameters from [**Shp04, SZ04**]. Also, we have implemented the attack described above. We tested several thousand instances, and our algorithm broke the system completely (i.e., correct guess for the complete plaintext w) in about 99% of the cases. The time the attack took ranged from under a second to several minutes, largely depending on the size of the generated public key. (In some rare cases, we even had to abort the key generation, since the memory usage was above one gigabyte.)

4. Foiling the Attack

The attack above needs in an essential way knowledge about free representatives of ciphertext and public key. And not only that, it assumes that encryption took place by performing a variable substitution according to (3.2) in the free group. In fact, the original system was specified exactly like this.

A very obvious way of how to break the assumptions needed for applying our attack would be to actually make use of the relations in $\mathcal{F}_{n+m}$. For example, the ciphertext could be "perturbed" by applying metabelian relations. This method can be combined by changing the presentation (i.e., the relations) of $\mathcal{F}_{n+m}$ as described in [**SZ05**, Section 7]. The problem with this is that there is no obvious way of how to do so *concretely* in a manner that is not invertible by an attacker.

Another way to use these relations would be to transmit the ciphertext as a vector of components in an $\mathcal{F}_{n+m}$-normal form. (In a certain sense, this means applying all relations simultaneously.) However, with the normal form described in [**Shp04, SZ04**] (for different purposes, see above), it is not clear how to decrypt the ciphertext in normal form. Namely, for decrypting c, it has to be composed with the secret key φ^{-1}. Basically, this means substituting all generators in φ^{-1} with the respective components (i.e., images on generators) of c. If these components are in a normal form, it must be possible to multiply two elements in such a normal form. It is not clear how to do so with the normal form from [**Shp04, SZ04**].

To this end, one can simply use a normal form that is multiplicative, in the sense that it is (efficiently) possible to multiply two elements in normal form to get the normal form of the product. The following $\mathcal{F}_{n+m}$-normal form is easily seen to be multiplicative:

For a free representant $z \in F_{n+m}$, let $\mathrm{NF}^*(z)$ be the vector of abelianized Fox derivatives of z, together with the abelianization of z itself. As $R \leq [F_{n+m}, F_{n+m}]$, this is still unique for two representants of the same word $\in \mathcal{F}_{n+m}$.

Furthermore, let two normal forms $\mathrm{NF}^*(z^1), \mathrm{NF}^*(z^2)$ of representatives $z^1, z^2 \in F_{n+m}$ be given. Then the normal form $\mathrm{NF}^*(z^1 z^2)$ of the product consists of the abelianized partial Fox derivatives $\overline{\mathrm{Fox}_i(z^1 z^2)}$ of $z^1 z^2$, and of the abelianized product $\overline{z^1 z^2}$. The latter can be trivially obtained from the abelianized $\overline{z^1}$ and $\overline{z^2}$ (which are part of $\mathrm{NF}^*(z^1)$, resp. $\mathrm{NF}^*(z^2)$), and $\overline{\mathrm{Fox}_i(z^1 z^2)}$ can be computed as

$$\overline{\mathrm{Fox}_i(z^1 z^2)} = \overline{\mathrm{Fox}_i(z^1)} + \overline{z^1} \cdot \overline{\mathrm{Fox}_i(z^2)}$$

by the rules of Fox derivatives.

Because of this efficient multiplication, this normal form can be applied to the ciphertext after encryption. In this way, an attacker does not learn a free representative of the encrypted value, and our attack from above will not work. However, for encryption, the public key still has to be in a free representation: the only obvious way to compute $c = w \circ \hat{\varphi}$ seems to be as a substitution as in (3.2). Here, generators x_j in a free representation of the public key $\hat{\varphi}$ are substituted with the corresponding components w_j of the plaintext w. For this, a free representation of $\hat{\varphi}$ must be available.

5. Further potential weaknesses

We have seen in the previous section how to address our attack by hiding the structure of the free representatives of the transmitted elements of $\mathcal{F}_{n+m}$. However, this may not be sufficient. First of all, note that the abelian part of the attack presented in Section 3.1 still works with the repaired scheme of Section 4. (This is so because for recovering the abelianized solution $\overline{w}$ of the plaintext, only the abelianized $\overline{c}$ and $\overline{\hat{\varphi}}$ are needed.) That is, the abelianized plaintext $\overline{w}$ can still be obtained in the repaired scheme; this might give at least partial information about the plaintext w. It does not seem easy to protect against this, since the abelianized $\overline{c}$ and $\overline{\hat{\varphi}}$ are already uniquely determined by $c, \hat{\varphi} \in \mathcal{F}_{n+m}$; the only way might be to "hide" these abelianizations in a normal form.

Moreover, we conducted experiments using a heuristic algorithm for finding a free representative of an element of $\mathcal{F}_{n+m}$ given in normal form. When a random element x of the free group F_{n+m} of length 500 was chosen, converted to normal form, and then converted back to a free element $\tilde{x}$ using the heuristic algorithm, the probability that $x = \tilde{x}$ was approximately 47 %.

To demonstrate the significance of this effect, assume as a thought experiment, that the probability for $x = \tilde{x}$ is near 1 even for long x. Then the improvements mentioned in Section 4 do not help against the attack of Section 3, since we can simply take the normal form and convert it back to the (probably) original element.

However, in reality the situation is not as simple. First, even for a length of 500, the probability of $x = \tilde{x}$ is only 47 %, so the probability that all transmitted elements are correctly reconstructed gets exponentially small in the number of the transmitted elements. Second, with increasing length, the probability of $x = \tilde{x}$ seems to fall rapidly (only 22 % for length 1000 and 6 % for length 2000). But the fact that words of length 500 can be perfectly reconstructed with probability 47 % indicates that the relations of $\mathcal{F}_{n+m}$ "strike" rarely, i.e., when considering a random element x of F_{n+m} the shortest representative $\tilde{x}$ of $x + R$ is with high probability similar to x (i.e., large subwords of x and $\tilde{x}$ are identical).

This hypothesis is further supported by the fact that the shortest word in R (except the empty word) has length 14 (in comparison to e.g., 4 for the relations of the free abelian group). Since further the approach of Section 3 could probably be made more fault tolerant by more sophisticated techniques,[2] it is possible that such a procedure might break the cryptosystem even if all transmitted elements

[2]Such techniques could include (1) eliminating heads or tails only if there are several indications (and not only one) to support this, (2) backtracking from errors, (3) looking at the interior of the free elements for additional hints, and (4) after each step converting the intermediate results back to the normal form and again to free elements to make use of the simplifications introduced by removing heads or tails (since by only dividing by the head or tail elements, we do not remove errors introduced by the relations of $\mathcal{F}_{n+m}$).

are sent using a normal form or disguised by random application of relations. This might also be considered an indication that the proposed platform group $\mathcal{F}_{n+m}$ is "too close" to a free group for cryptographic purposes.

6. Conclusions

We have shown a way to attack the metabelian group based public key cryptosystem due to Shpilrain and Zapata, and we have verified with experiments that our attack works. We have also shown how to prevent our attack, although even then adaptions of the attack often apply. In summary, we believe that further research is necessary regarding the suggested parameters (and possibly the proposed platform group) for the Shpilrain-Zapata cryptosystem.

References

[AAFG01] Iris Anshel, Michael Anshel, Benji Fisher, and Dorian Goldfeld, *New key agreement protocols in braid group cryptography*, Topics in Cryptology, Proceedings of CT-RSA 2001 (David Naccache, ed.), Lecture Notes in Computer Science, no. 2020, Springer-Verlag, 2001, pp. 13–27.

[AAG99] Iris Anshel, Michael Anshel, and Dorian Goldfeld, *An algebraic method for public-key cryptography*, Mathematical Research Letters **6** (1999), 287–291.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor, *Non-malleable cryptography*, Twenty-Third Annual ACM Symposium on Theory of Computing, Proceedings of STOC 1991, ACM Press, 1991, Extended abstract, full version online available at `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/nmc.ps`, pp. 542–552.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto, *How to enhance the security of public-key encryption at minimum cost*, Public Key Cryptography, Proceedings of PKC '99 (Hideki Imai and Yuliang Zheng, eds.), Lecture Notes in Computer Science, no. 1560, Springer-Verlag, 1999, pp. 53–68.

[GZ91] Max Garzon and Yechezkel Zalcstein, *The complexity of Grigorchuk groups with application to cryptography*, Theoretical Computer Science **88** (1991), no. 1, 83–98.

[Hof05] Dennis Hofheinz, *An attack on a group-based cryptographic scheme*, Invited talk at the 2nd Joint Meeting of AMS, DMV, and ÖMG, Mainz, June 2005.

[KLC+00] Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park, *New public-key cryptosystem using braid groups*, Advances in Cryptology, Proceedings of CRYPTO 2000 (Mihir Bellare, ed.), Lecture Notes in Computer Science, no. 1880, Springer-Verlag, 2000, pp. 166–183.

[Shp04] Vladimir Shpilrain, *Combinatorial group theory and public key cryptography*, Invited talk at the Canadian Mathematical Society Winter 2004 Meeting, Montreal, December 2004.

[SZ04] Vladimir Shpilrain and Gabriel Zapata, *Using the subgroup membership search problem in public key cryptography*, Unpublished, superseded by [**SZ06**], December 2004.

[SZ05] ———, *Combinatorial group theory and public key cryptography*, Applicable Algebra in Engineering, Communication and Computing (2005), To be published, online available at `http://eprint.iacr.org/2004/242.ps`.

[SZ06] ———, *Using the subgroup membership search problem in public key cryptography*, Algebraic Cryptography (Lothar Gerritzen, Dorian Goldfeld, Martin Kreuzer, Gerhard Rosenberger, and Vladimir Shpilrain, eds.), Contemporary Mathematics, American Mathematical Society, 2006, This volume, online available at `http://www.sci.ccny.cuny.edu/~shpil/crypmemb.pdf`, pp. 169–179.

[Wag84] Neal R. Wagner, *Searching for public-key cryptosystems*, IEEE Symposium on Security and Privacy, Proceedings of SSP '84, IEEE Computer Society, 1984, pp. 91–98.

Note also that the unmodified algorithm from Section 3 already has some fault tolerance, since it has to deal with elements corrupted by the cancellation of inverses.

[WM85] Neal R. Wagner and Marianne R. Magyarik, *A public key cryptosystem based on the word problem*, Advances in Cryptology, Proceedings of CRYPTO '84 (G. Robert Blakley and David Chaum, eds.), Lecture Notes in Computer Science, no. 196, Springer-Verlag, 1985, pp. 19–36.

Centrum voor Wiskunde en Informatica (CWI), Kruislaan 413, NL-1090 GB Amsterdam, The Netherlands

E-mail address: Dennis.Hofheinz@cwi.nl

Institut für Algorithmen und Kognitive Systeme (IAKS), Universität Karlsruhe, Am Fasanengarten 5, 76131 Karlsruhe, Germany

E-mail address: unruh@ira.uka.de