A note on the security of MST_3

María Isabel González Vasco, Angel L. Pérez del Pozo and Pedro Taborda Duarte

Dpto. de Matemática Aplicada, Universidad Rey Juan Carlos
C/ Tulipán s/n. 28933, Móstoles, Madrid, Spain
{mariaisabel.vasco, angel.perez, pedro.duarte}@urjc.es

January 30, 2009

Abstract

In this paper, we study the recently proposed encryption scheme MST_3, focusing on a concrete instantiation using Suzuki-2-groups. In a passive scenario, we argue that the one wayness of this scheme may not, as claimed, be proven without the assumption that factoring group elements with respect to *random covers* for a subset of the group is hard. As a result, we conclude that for the proposed Suzuki 2-groups instantiation, impractical key sizes should be used in order to prevent more or less straightforward factorization attacks.

Keywords: public key cryptography, cryptanalysis, group factorizations, covers, Suzuki 2-groups.

1 Introduction

There have been several attempts to exploit the computational properties of certain factorization sequences of finite groups for cryptography. More precisely, such an object is a finite sequence of blocks, which are also finite strings of group elements, such that it is possible to write each group element as a product selecting a factor from each block. This research line was initiated in [8], where S. Magliveras proposed a private key cryptosystem called PGM (*Permutation Group Mappings*) using *logarithmic signatures*, which are group bases yielding a unique factorization on each group element. Although many questions about PGM still remain unanswered, subsequent work revealed nice security properties of the scheme [11, 10, 9, 12].

Several years later, the ideas behind PGM were used to design two public key encryption schemes, MST_1 and MST_2 [13]. Again these constructions exploit special properties of factorization sequences in finite groups: while MST_1 makes use of a one-way permutation constructed from a logarithmic signature inducing a (hard to compute) factorization, MST_2 is inspired by ElGamal encryption [3]. For

this scheme, again, finding the factorization of a given group element with respect to certain public factorization sequences was assumed to be hard. Subsequent work proved this assumption to be rather unrealistic; in [5, 2], it was evidenced that all proposed key generation methods were susceptible to produce weak keys; i.e., factorization sequences with respect to which significant subsets of the underlying group could be factored efficiently.

Recently, Lempken et al. [7] put forward a new approach to designing public-key encryption schemes using the hardness of factoring with respect to group bases on finite groups. More precisely, their tools are *random covers* of finite non-abelian groups, i.e., factorization sequences in which blocks are constructed by sampling uniformly at random on the underlying group. They give a generic description of the proposed scheme – which they call MST_3 – and further ellaborate on an instantiation based on Suzuki 2-groups. This construction is indeed elegant and involves nice group theoretical tools, yet much work remains to be done in order to give precise guidelines towards a secure construction.

To that aim, in this paper, we focus on the proposed instantiation of MST_3 using Suzuki 2-groups. For the scheme to achieve one-wayness in a passive scenario, a certain factorization sequence which is part of the public key, should either cover a subset J of size significantly smaller than the center $\mathcal{Z}$ of the group or induce a hard factorization. In this note we explore to what extent these requirements may be met, if in the key generation process, covers are generated at random (following for instance [15]).

Our Contribution. We found, experimentally and for several parameters of the key generation algorithm, that the quotient $|\mathcal{Z}|/|J|$ (which in a sense measures the average number of representations through the cover for each element of J) is not necessarily large. As a result, in order to achieve one-wayness without further hypotheses one needs to assume the hardness of factoring with respect to a random cover.

We also evidence that the problem of factoring with respect to the public cover of the Suzuki 2-group can be reduced to factoring with respect to the naturally induced cover of the base field $\mathbb{F}_{2^m}$, which seems to be an easier task. Consequently, to avoid exhaustive-search enumeration attacks, a high value (of at least 80) for the security parameter m must be chosen. This leads to a public key of size much larger than the custom key size recommended for public key encryption nowadays.

Paper Roadmap. In Section 2 we briefly present the notations and main notions related to factorization sequences in finite groups, namely the notions of *cover* and of *logarithmic signature*. In Section 3 we describe the public key encryption scheme MST_3 and its instantiation using Suzuki 2-groups as presented in [7]. Subsequently, in Section 4 a security analysis of the proposal is presented. Further, we describe the experimental results which evidence that the assumption that random covers are hard to invert is essential for the security of this instantiation.

Finally, we argue the need for large parameters in order to thwart more or less straightforward factorization attacks.

2 Preliminaries

2.1 Covers and logarithmic signatures

Let us start by giving formal definitions of the notions we will be needing in the sequel:

Let $\mathcal{G}$ be a finite group and n be a positive integer. Suppose that for each $i = 1, \ldots, n$ we have a finite sequence $\alpha_i = [\alpha_{i1}, \ldots, \alpha_{ir_i}]$ with each $\alpha_{ij} \in \mathcal{G}$. Write $\alpha = [\alpha_1, \ldots, \alpha_n]$. Let S be a subset of $\mathcal{G}$.

Definition 2.1 *[Cover, Logarithmic Signature]*

i) α as above is said to be a cover *for S if any $g \in S$ can be written as a product*

$$g = \alpha_{1i_1} \ldots \alpha_{ni_n} \tag{1}$$

The vector $(r_1, \ldots, r_n)$ is called the type *of the cover α.*

ii) Let α be a cover for S. If the decomposition (1) is unique for every $g \in S$, then α is said to be a logarithmic signature *for S.*

Let α be a cover of type $(r_1, \ldots, r_n)$ for $S \subseteq \mathcal{G}$ and for each $m \in \mathbb{N}$, denote by $\mathbb{Z}_m$ the set $\{0, 1, \ldots, m-1\}$. Consider the mappings

$$\begin{aligned} \lambda : \ \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_n} &\longrightarrow \mathbb{Z}_{|\mathcal{G}|} \\ (k_1, \ldots, k_n) &\mapsto \sum_{i=1}^{n} (k_i \prod_{j=1}^{i-1} r_j) \end{aligned}$$

and

$$\begin{aligned} \theta_\alpha : \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_n} &\longrightarrow \mathcal{G} \\ (k_1, \ldots, k_n) &\mapsto \alpha_{1k_1} \ldots \alpha_{nk_n} \end{aligned}$$

The mapping λ is easily seen to be injective and moreover there is an efficient algorithm for computing λ^{-1}. Therefore we are able to efficiently compute

$$\begin{aligned} \check{\alpha} : \mathbb{Z}_{|\mathcal{G}|} &\longrightarrow \mathcal{G} \\ k &\mapsto \theta_\alpha(\lambda^{-1}(k)) \end{aligned}$$

Given a cover α for S and $g \in S$, computing an n-tuple $(k_1, \ldots, k_n) \in \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_n}$ such that $g = \alpha_{1k_1} \cdots \alpha_{nk_n}$ is equivalent to computing an element in $\check{\alpha}^{-1}(g)$. If this can be done in polynomial time for every $g \in \mathcal{G}$, α is said to be *tame*, otherwise, it is referred to as *wild*. For further details see [7].

3 The encryption scheme MST_3

We summarize in this section the presentation from [7].

3.1 Description

Given a security parameter $m \in \mathbb{N}$, let $\mathcal{G}$ be a finite non-abelian group with non-trivial center $\mathcal{Z}$ and such that $\mathcal{G}$ does not split over $\mathcal{Z}$. Moreover, assume the size of $\mathcal{Z}$ to be exponential in m. With this at hand, the MST_3 protocol is described in Figure 1.

Observe that the decryption process $\mathcal{D}ec$ is correct: on input a valid ciphertext (y_1, y_2) the corresponding plaintext is retrieved as follows:

$$\begin{aligned} y_2 &= \check{\gamma}(x) \\ &= \beta_{1i_1} t_0^{-1} \alpha_{1i_1} t1 \cdot \beta_{2i_2} t_1^{-1} \alpha_{2i_2} t_2 \cdot \ldots \cdot \beta_{si_s} t_{s-1}^{-1} \alpha_{si_s} t_s \\ &= \beta_{1i_1} \ldots \beta_{si_s} t_0^{-1} \alpha_{1i_1} \ldots \alpha_{si_s} t_s \\ &= \check{\beta}(x) t_0^{-1} \check{\alpha}(x) t_s \\ &= \check{\beta}(x) t_0^{-1} y_1 t_s \end{aligned}$$

and therefore $x = \check{\beta}^{-1}(y_2 t_s^{-1} y_1 t_0)$ which can be efficiently computed knowing the tame logarithmic signature β.

Remark 3.1 *The authors do not specify the response of $\mathcal{D}ec$ on input an invalid ciphertext; if, for instance $y_2 t_s^{-1} y_1^{-1} t_0$ does not lie in $\mathcal{Z}$. Clearly, if that is the case, $\mathcal{D}ec$ would not be able to compute an output value.*

Remark 3.2 *The condition imposed that "$\mathcal{G}$ does not split over $\mathcal{Z}$" is not enough to thwart attacks using permutation group algorithms. Note that for any subgroup H with $H \cap \mathcal{Z} = 1$ it is possible to efficiently write elements in $\mathcal{Z} \rtimes H$ as a product zh with $z \in \mathcal{Z}$ and $h \in H$. As a result, if a certain $\gamma_{ij} \in \mathcal{Z} \rtimes H$ (which can happen with high probability if $\mathcal{Z} \rtimes H$ is large) the corresponding β_{ij} could be achieved in polynomial time.*

3.2 A realization of MST3

In Section 4 of [7] the authors propose a concrete instantiation of the above scheme using Suzuki 2-groups as a base. We recall the basic terminology and some needed facts on the group, and refer the interested reader to Chapter VIII of [6] (see also Section 4.1 of [7]).

Let $m \geq 3$, be an odd natural number and θ a non trivial automorphism of odd order of the finite field $\mathbb{F}_q$ with $q = 2^m$. The *Suzuki* 2-group $\mathcal{G}$ of order q^2 can be realized as the subgroup of $GL(3, q)$ consisting of the matrices:

- $\mathcal{K}$: on input the security parameter m
  - Selects a group $\mathcal{G}$ as described above and generates a tame logarithmic signature
    $$\beta = [\beta_1, \ldots, \beta_s] := (\beta_{ij})$$
    of type $(r_1, \ldots, r_s)$ for $\mathcal{Z}$
  - Generates a random cover
    $$\alpha = [\alpha_1, \ldots, \alpha_s] := (\alpha_{ij})$$
    of the same type as β, for a certain "large" subset $J \subseteq \mathcal{G}$, and such that $\alpha_k \subseteq \mathcal{G} \backslash \mathcal{Z}$, $k = 1, \ldots, s$
  - Chooses $t_0, \ldots, t_s \in \mathcal{G} \backslash \mathcal{Z}$ and computes $\bar{\alpha} = [\bar{\alpha}_1, \ldots, \bar{\alpha}_s] := (\bar{\alpha}_{ij})$ where $\bar{\alpha}_k = t_{k-1}^{-1} \, \alpha_k \, t_k$
  - Computes $\gamma := (\gamma_{ij}) = (\beta_{ij} \; \bar{\alpha}_{ij})$
  - Outputs
    - a description of the group $\mathcal{G}$
    - the pair $pk = (\alpha, \gamma)$ as public key
    - the pair $sk = (\beta, (t_0, \ldots, t_s))$ as secret key

- $\mathcal{E}nc$: on input $x \in \mathbb{Z}_{|\mathcal{Z}|}$
  - Computes $y_1 = \check{\alpha}(x)$ and $y_2 = \check{\gamma}(x)$
  - Outputs $y = (y_1, y_2)$.

- $\mathcal{D}ec$: on input a ciphertext (y_1, y_2)
  - Computes $y_2 t_s^{-1} y_1^{-1} t_0$
  - Outputs $x = \check{\beta}^{-1}(y_2 t_s^{-1} y_1^{-1} t_0)$

Figure 1: MST3 Encryption Scheme

$$S(a,b) = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & a^\theta & 1 \end{pmatrix}$$

with $a,\, b \in \mathbb{F}_q$, i.e

$$\mathcal{G} = \{S(a,b) : a,\, b \in \mathbb{F}_q\}$$

It is easy to see that the center of $\mathcal{G}$ is $\mathcal{Z} = \{S(0,b) : b \in \mathbb{F}_q\}$ and thus has order q. Further, the group operation is given by the rule

$$S(a,b) \cdot S(x,y) = S(a+x, b+y+a^\theta \cdot x)$$

and as a result,

$$S(a,b)^{-1} = S(a, a^\theta a + b).$$

From the above it is easy to check that all elements in the center are involutions and all elements not in the center have order 4.

An instantiation of MST_3 is proposed on this platform groups, adding one only restriction to the key generation process: any two elements on the same block of the cover α must not be in the same coset of the center, i.e.,

$$\forall\, i \in \{1,\ldots,s\}, \forall\, j,k \in \{1,\ldots,r_i\} \text{ if } \alpha_{ij} \neq \alpha_{ik} \text{ then } \alpha_{ij}\, \alpha_{ik}^{-1} \notin \mathcal{Z}$$

4 Security Analysis

Security Model. We assume the reader to be somewhat familiar with standard security notions for encryption schemes; formal definitions can be seen, for instance in [1]. As proposed in [7], MST_3 is a deterministic scheme; as a result, the best we can hope for is security in the sense of one-wayness.

Remark 4.1 *Here it is worth noting that the randomized version of the scheme suggested by the authors is also insecure in the sense of indistinguishability, even when only considering passive adversaries. Indeed, the encryption procedure in this randomized version works as follows to encrypt a message $x \in \mathbb{Z}_{|\mathcal{Z}|}$,:*

- *It chooses a random number $R \in \mathbb{Z}_{|\mathcal{Z}|}$ and sets $y_0 = x + R$,*
- *computes $y_1 = \check{\alpha}(R)$ and $y_2 = \check{\gamma}(R)$*
- *outputs $y = (y_0, y_1, y_2)$.*

Now, if the adversary is to choose to messages m_0 and m_1, and then guess whether a ciphertext $c^* = (y_0^*, y_1^*, y_2^*)$ corresponds to any of them, all he has to do is check whether $\check{\alpha}(y_0^* - m_0) = y_1^*$, if affirmative, the hidden plaintext is m_0, otherwise c^* is an encryption of m_1.

Now, consider an active scenario of CCA type, where the adversary has access to a decryption oracle that will decrypt any ciphertext different from the challenge one $c^* = (y_1^*, y_2^*)$, for which he attempts to retrieve the corresponding plaintext. Clearly, he may simply choose an element $z \in \mathcal{Z}$ and feed the decryption oracle with the ciphertext $c = (y_1^* z, y_2^* z)$, obtaining the plaintext x corresponding to c^* as output.

Once it is clear that MST_3 is insecure against an active adversary one can wonder what happens if one considers a slightly weaker (yet active) scenario; that of VCA —validity checking— attacks:

Remark 4.2 *It is easy to argue that security against validity checking attacks is equivalent to CPA security, as a validity checking oracle can be simulated using only the public information.*
For this, let $y_1 = S(a, b),\ y_2 = S(\bar{a}, \bar{b}),\ t_0 = S(t_{0a}, t_{0b})$ *and* $t_s = S(t_{sa}, t_{sb})$. *Note that* $\mathcal{D}ec(y_1, y_2)$ *outputs a value if and only if* $y_2 t_s^{-1} y_1^{-1} t_0 \in \mathcal{Z}$, *and this happens if and only if* $a + \bar{a} = t_{0a} + t_{sa}$. *Therefore, all that is needed in order to simulate a validity checking oracle using only the public information, is to compute* $t_{0a} + t_{sa}$. *This can be done from any ciphertext* $(\check{\alpha}(x), \check{\gamma}(x))$ *as the "a-part" of* $\check{\gamma}(x)\check{\alpha}(x)$ *is* $t_{0a} + t_{sa}$.

Remark 4.3 *We note that it is possible to construct ciphertexts that would be successfully decrypted without following the encryption procedure. Just choose random* $a, b, \bar{b} \in \mathbb{F}_q$ *and send* $(y_1, y_2) = (S(a, b), S(a + (t_{0a} + t_{sa}), \bar{b}))$. *One may thus wonder whether a stronger VCA oracle should be considered. For example, an oracle* $\mathcal{O}$, *that on input* $S(a, b) \in \mathcal{G}$ *outputs 1 if there exists* x *such that* $S(a, b) = \check{\alpha}(x)$ *and 0 otherwise. Such an oracle would, in particular, help to discard many ciphertexts that did not follow the encryption procedure but that were accepted by the oracle constructed in remark 4.2. Nevertheless, assuming such an oracle is at hand is actually close to assuming access to a factorization oracle for the cover* α.

Seeing this, in the sequel we will focus on the security notion OW-CPA i.e. *one-wayness* in a passive scenario. In other words, we consider an adversary only observing the message flow and having access to the public keys, and explore whether he is able to retrieve the complete plaintext from an eavesdropped ciphertext.

4.1 Estimating the size of J

In Section 4.4 of [7] the authors discuss the security of MST_3 without the cryptographic hypothesis "*factorizing with respect to a randomly generated cover of a large subset of a finite group is hard*". They observe that the value $|\mathcal{Z}|/|J|$ can

be viewed as the average number of representations for each element of J with respect to the cover α. Then it is claimed that, if this value is large, the cryptosystem remains secure when the crytographic hypothesis for α is removed. This is due to the fact that finding a factorization of $y_1 = \breve{\alpha}(x)$ provides only a small probability of retrieving the correct x, as the number of different factorizations of y_1 is expected to be large.

However the authors do not explain how one can generate a cover α such that a large value $|\mathcal{Z}|/|J|$ is achieved, neither if this frequently happens when one just generate a random cover. We have found that, for uniformly randomly generated covers for several parameters of the proposed realization in the Suzuki 2-groups, the value $|\mathcal{Z}|/|J|$ is small, in fact smaller than 2, in all of our experiments. We devote the rest of this section to describe these experiments.

Remark 4.4 *A cover α for a set $J \subseteq \mathcal{G}$ induces a cover α^* of the same type for a subset $J_1 \subseteq \mathbb{F}_q$, by restricting to the "first" coordinate elements of α. The existence of a natural surjective map $p : J \longrightarrow J_1$, defined as $p(S(a,b)) = a$, implies that $|J| \geq |J_1|$. Therefore $|\mathcal{Z}|/|J| \leq |\mathcal{Z}|/|J_1|$ and it is enough for our purposes to upper bound $|\mathcal{Z}|/|J_1|$.*

Remark 4.5 *We assume that generating u.a.r. the cover α satisfying the imposed condition $\alpha_{ij}\ \alpha_{ik}^{-1} \notin \mathcal{Z}$ for the public key might be done in the following way:*

For each block α_i, choose u.a.r. values a_{ij} and b_{ij} in $\mathbb{F}_q$ such that:

1. *$a_{ij} \neq 0$ for every j and*
2. *$a_{ik} \neq a_{il}$ for $k \neq l$.*

But then, the induced cover α^* looks exactly like if we had constructed it by selecting u.a.r. values a_{ij} in $\mathbb{F}_q$ satisfying conditions (1) and (2) for each block. Therefore, for our experiments, we have generated covers just for subsets of $\mathbb{F}_q$, precisely in this way.

Description of the experiments: For each experiment we have arbitrarily chosen a value for the exponent m and a value for the type of the cover $(r_1, \ldots, r_s)$. Then, for these fixed values, we have repeated 50 times the following steps:

1. Generate a random cover as described in Remark 4.5, which will be a cover for certain unknown set $J_1 \subseteq \mathbb{F}_q$.
2. Determine $|J_1|$ just by brute force: make all the possible sums and count how many elements we obtain.
3. Store the value $|\mathcal{Z}|/|J_1| = q/|J_1|$.

Once we are done with the 50 iterations we store the minimum, the maximum and the arithmetic mean of the obtained values for $|\mathcal{Z}|/|J_1|$. All calculations have been made with GAP [4].

We repeated this experiment for different values of m and the type of the cover. The results are summarized in the following table:

m	type of the cover	min.	max.	mean
15	(16,16,16,8)	1.55	1.653	1.581
15	(64,32,16)	1.552	1.611	1.578
15	(8,8,8,8,8)	1.517	1.678	1.581
17	(32,16,16,16)	1.562	1.608	1.582
17	(128,64,16)	1.567	1.593	1.58
17	(8,8,8,8,8,4)	1.529	1.627	1.576
19	(32,32,32,16)	1.569	1.595	1.581
19	(256,64,32)	1.574	1.593	1.582
19	(8,8,8,8,8,4,4)	1.547	1.624	1.575
21	(64,32,32,32)	1.577	1.59	1.582
21	(512,128,32)	1.577	1.588	1.581

Remark 4.6 *Note that we cannot use large values for m, because step (2) in our previous experiment would have too much computational cost. In fact, if we assume that we have enough computational power for certain parameters to enumerate all the possible sums formed with one element from each block of α^*, then we can decrypt any ciphertext constructed in MST_3 with those parameters, just by enumeration. Therefore it is not feasible to make this experiment for any parameters which are believed to provide some kind of security to MST_3. Nevertheless our experiments evidence that assuming $|\mathcal{Z}|/|J_1|$ to be large is rather unrealistic (even if m is large).*

Remark 4.7 *The results of the experiments also weaken MST_3 in another important way: the fact that $|\mathcal{Z}|/|J_1|$ is smaller than 2 implies that, for each element in $|J_1|$, there will often be just one unique factorization with respect to α^*. Thus, if $y_1 = \check{\alpha}(x) = S(a, b)$, then factoring a with respect to α^* will allow us to retrieve x, i.e. the problem of factoring y_1 with respect to α is reduced to factoring a with respect to α^*.*

4.2 Remarks on factoring in $\mathbb{F}_q$ and the key length

It follows from Remark 4.7 that the OW-CPA security of the proposed realization of MST_3 is compromised if an adversary is able to factorize with respect to a random cover of $\mathbb{F}_q$. Although we do not know how to efficiently solve this problem in the

general case, some remarks about it can be made. In the sequel, α^* will denote a random cover of $\mathbb{F}_q$ of type $(r_1, \ldots, r_s)$.

Remark 4.8 *We will assume, without loss of generality, that the vector* $0 \in \mathbb{F}_q$ *belongs to every block of* α^*.

Proof. Such a signature is constructed from a random cover α as follows:

1. Choose a sandwich transform $\widetilde{\alpha}$ of α^* such that the first element of each of the blocks $\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_{s-1}$ equals 0. Both signatures are equivalent, that is, the factorization of an element $a \in \mathbb{F}_q$ is obtained in the same way for both (see [13]).

2. Construct another signature $\widehat{\alpha}$ with $\widehat{\alpha}_j = \widetilde{\alpha}_j$ for $j < s$ and $\widehat{\alpha}_s = \widetilde{\alpha}_s + \widetilde{\alpha}_{s,1}$. Now, the first element of each block of $\widehat{\alpha}$ equals 0. Finding the factorization of an element $a \in \mathbb{F}_q$ w.r.t. $\widetilde{\alpha}$ is equivalent to factorize $\alpha + \widetilde{\alpha}_{s,1}$ w.r.t. $\widehat{\alpha}$.

□

A tame subcover of α^*. We will look for a cover δ such that:

i) $0 \in \delta_j \subseteq \alpha^*_j$ for every $j \in \{1, \ldots, s\}$.

ii) $d_j = |\delta_j| \geq 2$ for every $j \in \{1, \ldots, s\}$.

iii) $\sum_{i=1}^{s} d_j = m + s$ (i.e. there are m non-zero vectors in δ).

The probability that m vectors chosen at u.a.r. from $\mathbb{F}_q$ are linearly independent is higher than $1/4$, thus it is possible to find, just by trial and error, a cover δ satisfying conditions i) to iii) and such that the m non-zero vectors are linearly independent. Such a cover will be, by construction, a sublogarithmic signature of a linear transformation of a canonical logarithmic signature of $\mathbb{F}_q$. Therefore δ is tame (see [14]).

This means that, by using the signature δ we are able to find the factorization w.r.t. α^* of $\prod_{j=1}^{s} d_j$ different elements among the 2^m elements of $\mathbb{F}_q$. Let us show how this works in a couple of examples, where we fix the type of the random cover α^*:

Example 4.9 *Let* $r_j = 2$ *for every* j. *Then* $s = m$ *and* $\delta = \alpha^*$ *(in case the* m *non-zero vectors are independent, which happens with probability higher than* $1/4$*). Therefore such a signature will always be tame.*

Example 4.10 *Let $r_j = 4$ for every $j \in \{1, \ldots, s-1\}$ and $r_s = 2$, therefore $s = (m+1)/2$. Suppose we find δ as described before with $d_j = 3$ for $j < s$ and $d_s = 2$. Then we can factor $2 \cdot 3^{s-1}$ elements of $\mathbb{F}_q$ through δ, that is we have a success probability*

$$\frac{2 \cdot 3^{s-1}}{2^m} = \frac{3^{(m-1)/2}}{2^{m-1}} = \left(\frac{3}{4}\right)^{(m-1)/2}$$

The success probability obviously decreases exponentially to zero as m grows. But it is non-negligible for certain values of m which could seem, at first glance, adequate for the proposed realization of MST_3. For example, for $m = 61$ the success probability is around 10^{-4}.

Key length. At this point, it is worth noting that the Suzuki 2-groups instantiation is far away from being suitable for practical applications, as it can only be derived using rather large keys. Indeed, for a fixed odd m, the number of elements involved in β—and so, in γ and α—is, at least $2m$ group elements.[1] As each group element is represented via two elements in F_{2^m}, we need $2m$ bits to represent it; thus, the public key has at least $8m^2$ bits. On the other hand, note that the results in this section imply that it is necessary to set $m \geq 80$, to avoid trivial exhaustive-search factorization attacks. This yields a public key of $27,200$ bits, which is over ten times larger than the custom key size recommended for public key encryption nowadays.

References

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO '98: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, pages 26–45, London, UK, 1998. Springer-Verlag.

[2] J.M. Bohli, M.I. González Vasco, C. Martínez, and R. Steinwandt. Weak Keys in MST_1. *Designs, Codes and Cryptography*, 37:509–524, 2005.

[3] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithm . *IEEE Trans. Info. Theory*, 31:469–472, 1985.

[4] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4.10*, 2007. `http://www.gap-system.org`.

[5] M.I. González Vasco and R. Steinwandt. Obstacles in Two Public-Key Cryptosystems Based on Group Factorizations. In *Cryptology*, Tatra Mountains Math. Publications, pages 23–37, 2002.

[1] This is argued from $\prod_{i=1}^{s} r_i = 2^m$, see [5].

[6] B. Huppert and N. Blackburn. *Finite Groups II.* Springer, 1982.

[7] W. Lempken, S.S. Magliveras, T. van Trung, and W. Wei. A Public Key Cryptosystem Based on Non-abelian Finite Groups. *Journal of Cryptology.*

[8] S.S. Magliveras. A cryptosystem from logarithmic signatures of finite groups. In *Proceedings of the 29'th Midwest Symposium on Circuits and Systems*, pages 972–975. Elsevier Publishing Company, 1986.

[9] S.S. Magliveras and N.D. Memon. Linear complexity profile analysis of the PGM cryptosystem. *Congresus Numerantium, Utilitas Mathematica*, 72:51–60, 1989.

[10] S.S. Magliveras and N.D. Memon. Properties of cryptosystem PGM. In *Advances in Cryptology. Proceedings of CRYPTO 1989*, Lecture Notes on Computer Science, pages 447–460, Berlin, 1989. Springer-Verlag.

[11] S.S. Magliveras and N.D. Memon. Complexity tests for cryptosystem PGM. *Congresus Numerantium, Utilitas Mathematica*, 79:61–68, 1990.

[12] S.S. Magliveras and N.D. Memon. Algebraic properties of cryptosystem PGM. *Journal of Cryptology*, 5:167–183, 1992.

[13] S.S. Magliveras, D.R. Stinson, and T. Trung. New approaches to designing public key cryptosystems using one-way functions and trap-doors in finite groups. *Journal of Cryptology*, 15:285–297, 2002.

[14] S.S. Magliveras, P. Svaba, T. van Trung, and P. Zajac. On the security of a realization of cryptosystem MST_3. 2008. preprint: `http://www.iem.uni-due.de/preprints/Security_of_MST3_Preprint Tran.pdf`.

[15] P. Svaba and T. Van Trung. On generation of random covers for finite groups. *Tatra Mountains Mathematical Publications*, 37:105–112, 2007.