Cryptanalysis of the MST_3 Public Key Cryptosystem

Simon R. Blackburn, Carlos Cid and Ciaran Mullan*
Information Security Group,
Royal Holloway, University of London
Egham, Surrey TW20 0EX, United Kingdom
{s.blackburn,carlos.cid,c.mullan}@rhul.ac.uk

June 16, 2009

Abstract

In this paper we describe a cryptanalysis of MST_3, a public key cryptosystem based on non-commutative groups recently proposed by Lempken, Magliveras, van Trung and Wei.

1 Introduction

Recently Lempken, Magliveras, van Trung and Wei [9] proposed a new public key cryptosystem called MST_3 based on non-commutative groups. In their paper, the authors present a practical instance of the cryptosystem using Suzuki 2-groups. This is the third version of a family of cryptosystems, all roughly based on the idea that factorisation in groups is a hard problem. There has been a significant amount of research on these problems in the cryptographic literature: see [2, 3, 6, 11, 12, 14, 16] for example. The MST_3 proposal is part of a trend in recent years to study *post-quantum* cryptography, namely primitives that might remain secure if practical quantum computers are constructed; examples include lattice-based, code-based and multivariate public key cryptosystems as well as cryptosystems based on non-commutative objects [1].

An initial discussion of the security of MST_3 was presented in [9]. Magliveras, Svaba, van Trung and Zajac [13] showed that MST_3 is insecure (even for passive adversaries) when a certain special but natural method (using 'canonical transversal logarithmic signatures') for generating secret keys is used. González Vasco, Perez del Pozo and Taborda Duarte [4] showed that a randomised version of MST_3 is insecure in the sense of indistingishability, even in a passive adversary model. They also provided strong evidence that the One Way Encryption

[*] This work was supported by E.P.S.R.C. PhD studentship EP/P504309/1.

(OWE) security of MST_3 against passive attacks relies on a certain security assumption (on the one-wayness of random covers of groups) that the authors of MST_3 claimed was not needed.

The papers above still leave open the question of whether MST_3 is secure in practice if canonical transversal logarithmic signatures are avoided in the generation of the private key. The aim of this paper is to provide a practical cryptanalysis of the MST_3 cryptosystem when private keys are generated in the most general way known to the authors.

This paper is organised as follows. In Section 2 we describe the MST_3 cryptosystem and make some initial observations on its security. We also discuss the cryptanalytic results of Magliveras et al [13] and of González Vasco et al [4]. In Section 3 we describe a simplification of the cryptosystem. We present our attacks against MST_3 in Section 4, and provide a conclusion in Section 5.

2 The MST_3 Cryptosystem

We first describe the basic concepts behind the MST_3 cryptosystem, as well as the notation used. For more details, see [9].

2.1 Covers and logarithmic signatures

Let G be a finite group, $S \subseteq G$ a subset of G and s a positive integer. For all $1 \leq i \leq s$, let $A_i = [\alpha_{i1}, \ldots, \alpha_{ir_i}]$ be a finite sequence of elements of G of length $r_i > 1$, and let $\alpha = [A_1, \ldots, A_s]$ be the ordered sequence of A_i. We say that α is a *cover* for S if any $g \in S$ can be written as a product

$$g = g_1 \cdots g_s,$$

where $g_i = \alpha_{ik_i} \in A_i$. If such a decomposition is unique for every $g \in S$, then α is said to be a *logarithmic signature* for S. The *type* of a cover α is the vector $(r_1, \ldots, r_s)$. Given an element $g \in G$ and a cover α of G, obtaining a factorisation $g = \alpha_{1k_1} \cdots \alpha_{sk_s}$ associated with α could well be a hard problem in general. But if this factorisation can be efficiently computed for every $g \in G$, we say that α is *tame* (otherwise, α is said to be *wild*).

Let α be a cover for a subset S of G of type $(r_1, \ldots, r_s)$, and $q = \prod_{i=1}^{s} r_i$. Consider the maps λ_α and θ_α defined by

$$\begin{array}{rccl} \lambda_\alpha : & \mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_s} & \longrightarrow & \mathbb{Z}_q \\ & (k_1, \ldots, k_s) & \longmapsto & \sum_{i=1}^{s} (k_i \prod_{j=1}^{i-1} r_j), \end{array}$$

and

$$\begin{array}{rccl} \theta_\alpha : & \mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_s} & \longrightarrow & G \\ & (k_1, \ldots, k_s) & \longmapsto & \alpha_{1k_1} \cdots \alpha_{sk_s}. \end{array}$$

We note that λ_α is a bijection, and both λ_α and λ_α^{-1} are efficiently computable. We can define the map

$$\begin{array}{rccl} \breve{\alpha} : & \mathbb{Z}_q & \longrightarrow & G \\ & k & \longmapsto & \theta_\alpha(\lambda_\alpha^{-1}(k)). \end{array}$$

Given a cover α for S and an element $g \in S$, computing an s-tuple $(k_1, \ldots, k_s)$ such that $g = \alpha_{1k_1} \cdots \alpha_{sk_s}$ is equivalent to computing the inverse $\theta_\alpha^{-1}(g)$. It follows that α is tame if and only if $\breve{\alpha}^{-1}(g)$ can be efficiently computed for every $g \in S$.

2.2 Definition of MST_3

Let G be a finite non-abelian group with non-trivial centre $\mathcal{Z}$, with the property that G does not split over $\mathcal{Z}$ (so G cannot be written as a direct product $G = \mathcal{Z} \times H$ for some subgroup H). The MST_3 cryptosystem can be described as follows:

Key Generation:

- Generate a tame logarithmic signature $\beta = [B_1, \ldots, B_s] := (\beta_{ij})$ of type $(r_1, \ldots, r_s)$ for $\mathcal{Z}$.
- Generate a random cover $\alpha = [A_1, \ldots, A_s] := (\alpha_{ij})$ of the same type as β for a certain (large) subset $J \subseteq G$.
- Select random elements $t_0, \ldots, t_s \in G \backslash \mathcal{Z}$ and compute $\bar{\alpha} = [\bar{A}_1, \ldots, \bar{A}_s] := (\bar{\alpha}_{ij})$, where $\bar{A}_k = t_{k-1}^{-1} A_k t_k$ for $k = 1, \ldots, s$.
- Compute $\gamma := (\gamma_{ij}) = (\beta_{ij}\bar{\alpha}_{ij})$.

The pair (α, γ) is the public key, while $(\beta, (t_0, \ldots, t_s))$ is the corresponding private key.

Encryption:
A message $p \in \mathbb{Z}_{|\mathcal{Z}|}$ is encrypted as the pair $(\breve{\alpha}(p), \breve{\gamma}(p)) := (y_1, y_2)$ (recall that given covers α, γ, one can efficiently compute the mappings $\breve{\alpha}$ and $\breve{\gamma}$).

Decryption:
The plaintext p can be obtained from the ciphertext (y_1, y_2) as follows:

- Since $y_2 = \breve{\gamma}(p) = \beta_{1j_1}\bar{\alpha}_{1j_1} \cdot \beta_{2j_2}\bar{\alpha}_{2j_2} \cdots \beta_{sj_s}\bar{\alpha}_{sj_s}$, and the elements β_{ij} are in the centre of G, we have

$$\begin{aligned} y_2 &= (\beta_{1j_1}\beta_{2j_2} \cdots \beta_{sj_s}) t_0^{-1} (\alpha_{1j_1}\alpha_{2j_2} \cdots \alpha_{sj_s}) t_s \\ &= \breve{\beta}(p) t_0^{-1} \breve{\alpha}(p) t_s \\ &= \breve{\beta}(p) t_0^{-1} y_1 t_s. \end{aligned}$$

  As a result one can compute $\breve{\beta}(p) = y_2 t_s^{-1} y_1^{-1} t_0$.

- Now one can recover $p = \breve{\beta}^{-1}(y_2 t_s^{-1} y_1^{-1} t_0)$, since β is tame.

We note that Lempken et al [9] require the random cover α to have the property that $A_k \subseteq G \backslash \mathcal{Z}$ for $k = 1, \ldots, s$. We have dropped this requirement, since: the property is not needed for the encryption and decryption algorithms to work correctly; the property holds with high probability if the elements α_{ij} are chosen uniformly and independently at random; we wish to allow all covers α as valid private keys for the purposes of our cryptanalysis.

Note that the cryptosystem is not yet completely specified: a suitable platform group G needs to be defined, and we need to specify how to choose the logarithmic signature β. (It seems reasonable to assume that the elements in α and the elements t_i are chosen uniformly and independently at random.) In the next subsection, we discuss the proposal in [9] for the platform group G. The issue of how to generate β is not discussed in depth in [9], but we discuss a general method for accomplishing this in Subsection 2.4.

2.3 A realisation of MST_3

In [9], the authors propose a practical realisation for MST_3 using Suzuki 2-groups. (See Higman [7] for a description of these groups.) Let $m \geq 3$ be an odd natural number and θ a non-trivial automorphism of odd order of the finite field $\mathbb{F}_q$, where $q = 2^m$. The Suzuki 2-group G of order q^2 can be realised as the subgroup of $GL_3(q)$ consisting of the matrices

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

Thus $G = \{S(a,b) : a, b \in \mathbb{F}_q\}$ with centre $\mathcal{Z} = \{S(0,b) : b \in \mathbb{F}_q\}$. Multiplication and inversion in G are given by

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

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

It follows that all elements in the centre have order 2, while elements not in the centre have order 4.

Lempken et al [9] impose an extra condition on α when Suzuki 2-groups are used as a platform for MST_3, namely that no two elements of a set A_i should lie in the same coset of $\mathcal{Z}$. Since this condition holds for an overwhelming proportion of keys for interesting parameters (and since the condition is not relevant to our attacks) we ignore it for the sake of simplicity.

2.4 Constructing tame logarithmic signatures for $\mathcal{Z}$

The cryptosystem calls for us to generate a tame logarithmic signature β for the centre $\mathcal{Z}$ of the Suzuki 2-group. How should this be done?

Magliveras, Svaba, van Trung and Zajac [13] suggest the following procedure for generating β. Elements of $\mathcal{Z}$ are represented as binary vectors of length m, where the group operation is XOR. Partition $\{1, 2, \ldots, m\}$ into disjoint sets $C_1, C_2, \ldots, C_s$. Define integers d_i by $d_i = |C_i|$, and set $r_i = 2^{d_i}$. Let

$V_1, V_2, \ldots, V_s$ be subgroups of $\mathcal{Z}$ where V_i consists of the 2^{d_i} vectors that are all zero except possibly the positions indexed by integers in C_i. Let M be a randomly chosen invertible $m \times m$ matrix, and set B_i to be the result of multiplying the elements in V_i by the matrix M. The resulting cover $\beta = [B_1, B_2, \ldots, B_s]$ is easily seen to be a logarithmic signature of type $(r_1, r_2, \ldots, r_s)$; Magliveras et al use the term *canonical logarithmic signatures* for covers constructed in this way. Note these logarithmic signatures form a very special class: in particular, all blocks B_i are in fact subgroups of $\mathcal{Z}$. Canonical logarithmic signatures have the advantage that they can be stored and manipulated very efficiently, provided the elements of the sets B_i are stored in a certain order: see [13] for details.

We are interested in a much more general way of constructing a logarithmic signature for $\mathcal{Z}$. Suppose we begin by choosing a chain

$$1 = \mathcal{Z}_0 < \mathcal{Z}_1 < \cdots < \mathcal{Z}_u = \mathcal{Z}$$

of subgroups in $\mathcal{Z}$. Let E_i be a complete set of coset representatives for $\mathcal{Z}_{i-1}$ in $\mathcal{Z}_i$. Then $\epsilon = [E_1, \ldots, E_u]$ is a tame logarithmic signature (a so-called *transversal logarithmic signature*) for $\mathcal{Z}$. We now perform some of the following operations on our logarithmic signature (which do not alter the fact that we have a tame logarithmic signature for $\mathcal{Z}$):

- permute elements within each E_i;
- permute the E_i;
- replace E_i by $E_i \cdot z$ for some $z \in \mathcal{Z}$;
- amalgamate two sets by replacing E_i and E_j by the single set $E_i \cdot E_j := \{gh \mid g \in E_i, h \in E_j\}$.

We will call a logarithmic signature constructed in this fashion an *Amalgamated Transversal Logarithmic Signature* (ATLS). This is the most general method known to the authors for generating a logarithmic signature of elementary abelian 2-group. One might try to generalise this construction by considering the operation which replaces each element by its image under a fixed automorphism of $\mathcal{Z}$. However, this operation does not lead to new logarithmic signatures: the signatures are obtained by starting with a different chain of subgroups $\mathcal{Z}_i$. Holmes [8] and Lempken and van Trung [10] constructed logarithmic signatures for some simple groups using a double coset decomposition as part of an attempt to settle a question of González Vasco, Rötteler and Steinwandt [5]. However, these methods do not lead to new logarithmic signatures because our group $\mathcal{Z}$ is abelian. Another idea might be to try and reverse the amalgamating operation, by writing a set E_i as the product of two smaller sets. But we do not know of examples where this produces a non-ATLS (it is likely that the result of a splitting operation is just an ATLS which uses fewer amalgamation operations). Note that even if examples of new logarithmic signatures can be obtained using this splitting operation, the problem of efficiently detecting when a set E_i can be split would remain.

We are interested in generating logarithmic signatures that can be stored and manipulated without too much computational overhead. Because of this, the number of amalgamation operations has to be kept small: an amalgamation increases the number of elements we have to store by $|E_i||E_j| - (|E_i| + |E_j|)$, and so an indiscriminate use of amalgamation could lead to an exponential storage requirement. From the perspective of efficiency, generating an ATLS of type $(2, 2, \ldots, 2)$ is very attractive (though this would mean that we are unable to use amalgamation to construct them).

The following key property that holds for any ATLS will prove useful for our cryptanalysis:

Lemma 1. *Let $\beta = [B_1, \ldots, B_s]$ be an ATLS. Suppose that $1 \in B_i$ for all i. Then there exists a subset B_i and a non-trivial element $b_{ij} \in B_i$ such that $B_i \cdot b_{ij} = B_i$.*

Proof. The subgroup $\mathcal{Z}_1$ has been amalgamated into one of the subsets B_i. It is not difficult to show that $\mathcal{Z}_1 \subseteq B_i$ and $B_i\mathcal{Z}_1 = B_i$, and so we may take b_{ij} to be any non-identity element of $\mathcal{Z}_1$. □

2.5 Previous work on the security of MST_3

In this subsection, we briefly review previous work addressing the security of MST_3, and make some elementary observations on the system's security.

In [9], the authors of MST_3 provide a brief discussion on the security of the scheme, and give an attack on the cryptosystem in the passive adversary model with complexity approximately q^2 when Suzuki 2-groups are used, where $q = |\mathcal{Z}| = |G/\mathcal{Z}|$.

Magliveras et al [13] provide a better attack with complexity approximately q. They only claim that their attack applies when the Suzuki 2-groups are used as the platform, but in fact their attack works for any platform group. We provide a similar generic attack in Section 3 below, as the first step in our cryptanalysis. Magliveras et al go on to show that MST_3 is insecure whenever β is a canonical logarithmic signature. (In fact their attack does not work in the interesting special case when $d_i = 1$ for all i, as they need that the sum of the vectors in a subspace is zero; our cryptanalysis will cover this special case.) Note that it is easy to avoid the attack in [13]: either choose $d_i = 1$ for all i, or generate an ATLS as described in Subsection 2.4 (which is very unlikely to be canonical).

The authors of MST_3 assume [9, Section 1] that a randomly chosen cover α in a finite group will (with overwhelming probability) induce a one-way function $\breve{\alpha}$. This is a reasonable assumption, but the authors claim (in Section 4.4 of their paper) that this assumption is not actually needed to establish the security of MST_3 (in a passive model). Gonzalez Vasco, Perez del Pozo and Taborda Duarte [4] provide strong evidence that this last claim is false, by showing that when α does not induce a one-way function, MST_3 is insecure unless the quotient $|\mathcal{Z}|/|J|$ is large. They then provide experimental evidence that $|\mathcal{Z}|/|J|$ is usually rather small. Gonzalez Vasco et al also show that a randomised version of MST_3 is insecure in the sense of indistinguishability, even for passive adversaries.

We finish this section with two elementary remarks on the security of the scheme:

1. Note that although the private key consists of the tame logarithmic signature β and the $s+1$ randomly generated elements $\{t_0, \ldots, t_s\}$, the $s-1$ elements $t_1, \ldots, t_{s-1}$ are not actually needed: only β and t_0, t_s are used in the decryption procedure.

2. Note that any triplet of the form $(\beta, g \cdot t_0, g \cdot t_s)$, where g is in the centralizer of J (in particular, if $g \in \mathcal{Z}$), can be used to decrypt the ciphertext. Thus there are many equivalent private keys.

3 A Simplification of the MST_3 Cryptosystem

The aim of this section is to simplify the problem of cryptanalysing MST_3: we will show that it is sufficient to consider a much smaller class of public and private keys than in the original definition. This simplification works for all suitable platform groups, not just the Suzuki 2-groups considered above.

Let (α, γ) be a public key for MST_3, with $(\beta, (t_0, t_1, \ldots, t_s))$ the corresponding private key. Note that the algorithm for deriving γ from the private key implies that

$$\gamma_{ij} = \beta_{ij} t_{i-1}^{-1} \alpha_{ij} t_i. \tag{1}$$

Define elements p_i, q_i and z_i by setting $p_0 = q_0 = z_0 = 1$ and for $i \in \{1, 2, \ldots, s\}$ defining

$$p_i = \prod_{k=1}^{i} \alpha_{k1}, \ q_i = \prod_{k=1}^{i} \gamma_{k1} \text{ and } z_i = \prod_{k=1}^{i} \beta_{k1}.$$

Note that (1) and the fact that the elements β_{ij} are central together imply that

$$q_i = \prod_{k=1}^{i} (\beta_{k1} t_{k-1}^{-1} \alpha_{k1} t_k) = z_i t_0^{-1} p_i t_i. \tag{2}$$

Define $\alpha' = [A'_1, A'_2, \ldots, A'_s]$, $\gamma' = [H'_1, H'_2, \ldots, H'_s]$ and $\beta' = [B'_1, B'_2, \ldots, B'_s]$ by

$$\begin{aligned} A'_i &= p_{i-1} A_i p_i^{-1}, \\ H'_i &= q_{i-1} H_i q_i^{-1}, \\ B'_i &= z_{i-1} B_i z_i^{-1}. \end{aligned}$$

The following lemma is easy to prove.

Lemma 2. *We use the notation defined above. For all $i \in \{1, 2, \ldots, s\}$, the first elements α'_{i1}, γ'_{i1}, β'_{i1} of the sets A'_i, H'_i, B'_i are all equal to the identity. Moreover,*

$$\breve{\alpha}'(x) = \breve{\alpha}(x) p_s^{-1}, \breve{\gamma}'(x) = \breve{\gamma}(x) q_s^{-1} \text{ and } \breve{\beta}'(x) = \breve{\beta}(x) z_s^{-1}.$$

In particular, β' is a logarithmic signature for $\mathcal{Z}$, and α' is a cover for some subset $\mathcal{J}'$ of G.

Lemma 3. *Let (α, γ) be a public key for MST_3, with $(\beta, (t_0, t_1, \ldots, t_s))$ the corresponding private key. Define α', γ' and β' as above, and let $t'_0 = t'_1 = \cdots = t'_s = t_0$. Then (α', γ') is a public key for MST_3, with corresponding private key $(\beta', (t'_0, t'_1, \ldots, t'_s))$.*

Proof. Suppose we use α', β' and $t'_0, t'_1, \ldots, t'_s$ to generate a public key (α', δ), where $\delta = [D_1, D_2, \ldots, D_s]$, so $\delta_{ij} = \beta'_{ij}(t'_{i-1})^{-1}\alpha'_{ij}t'_i$. It suffices to show that $\delta = \gamma'$. But

$$\begin{aligned}
\delta_{ij} &= \beta'_{ij}t_0^{-1}\alpha'_{ij}t_0 \\
&= z_{i-1}\beta_{ij}z_i^{-1}t_0^{-1}\alpha'_{ij}t_0 \\
&= z_{i-1}\beta_{ij}z_i^{-1}t_0^{-1}p_{i-1}\alpha_{ij}p_i^{-1}t_0 \\
&= \beta_{ij}z_{i-1}z_i^{-1}t_0^{-1}p_{i-1}\alpha_{ij}p_i^{-1}t_0 \\
&= \beta_{ij}\beta_{i1}^{-1}t_0^{-1}p_{i-1}\alpha_{ij}p_i^{-1}t_0.
\end{aligned}$$

Equation (2) implies that $t_0^{-1}p_{i-1} = z_{i-1}^{-1}q_{i-1}t_{i-1}^{-1}$ and $p_i^{-1}t_0 = t_iq_i^{-1}z_i$. So

$$\begin{aligned}
\delta_{ij} &= \beta_{ij}\beta_{i1}^{-1}z_{i-1}^{-1}q_{i-1}t_{i-1}^{-1}\alpha_{ij}t_iq_i^{-1}z_i \\
&= \beta_{ij}q_{i-1}t_{i-1}^{-1}\alpha_{ij}t_iq_i^{-1}
\end{aligned}$$

by the definition of z_i, and since z_i is central. But

$$\gamma'_{ij} = q_{i-1}\gamma_{ij}q_i = q_{i-1}\beta_{ij}t_{i-1}^{-1}\alpha_{ij}t_iq_i^{-1}$$

by (1). Since β_{ij} is central, we have that $\gamma'_{ij} = \delta_{ij}$, as required. □

We define the **Restricted OWE problem** for MST_3 as follows. The input is a public key (α, β) for MST_3 and a challenge ciphertext (y_1, y_2). The public key must have the extra property that $\alpha_{i1} = \gamma_{i1} = 1$ for $1 \leq i \leq s$; the corresponding private key must have the property that $t_0 = t_1 = \cdots = t_s$ and also that $\beta_{i1} = 1$ for $1 \leq i \leq s$. The output is the plaintext p corresponding to the ciphertext (y_1, y_2).

Theorem 4. *There is a polynomial time reduction from the OWE problem for MST_3 (for general keys) to the Restricted OWE problem for MST_3.*

Proof. Let $\mathcal{O}(\alpha, \gamma, y_1, y_2)$ be an oracle for the restricted OWE problem for MST_3. We show that this oracle can be used to solve the OWE problem for MST_3 for general keys.

Suppose (α, γ) is an (unrestricted) public key, with corresponding private key $(\beta, (t_0, t_1, \ldots, t_s))$. Let (y_1, y_2) be a challenge ciphertext with corresponding message p.

Suppose we are given (α, γ) and (y_1, y_2). Define (α', γ') as above. Note that α' and γ' can be efficiently constructed from α and γ using public information only. By Lemmas 2 and 3, (α', γ') is a public key with corresponding private key $(\beta', (t_0, t_0, \ldots, t_0))$, and these keys satisfy our restrictions. Define $y_1' = y_1 p_s^{-1}$ and $y_2' = y_2 q_s^{-1}$. Again, we note that p_s and q_s are defined using public information, so y_1' and y_2' can be efficiently computed from the information we are given.

We call the oracle $\mathcal{O}$ on $(\alpha', \gamma', y_1', y_2')$, and receive a message p such that $(\alpha'(p), \gamma'(p)) = (y_1', y_2')$. Then p is the message we require, since

$$\breve{\alpha}(p) = \breve{\alpha}'(p)p_s = y_1' p_s = y_1 p_s^{-1} p_s = y_1 \text{ and}$$
$$\breve{\gamma}(p) = \breve{\gamma}'(p)q_s = y_2' q_s = y_2 q_s^{-1} q_s = y_2.$$

□

4 Cryptanalysis of MST_3

This section is concerned with the cryptanalysis of MST_3. In Subsection 4.1 we provide an attack that is independent of the underlying platform group G. In Subsection 4.2 we outline an approach that works for platform groups G such that $G/\mathcal{Z}$ is abelian. Finally, in Subsection 4.3 we report on our experiments with implementing these attacks in the case when G is a Suzuki 2-group.

4.1 A generic attack

From now on, we assume our public key (α, γ) and corresponding private key $(\beta, (t_0, t_1, \ldots, t_s))$ are such that

$$\alpha_{i1} = \beta_{i1} = \gamma_{i1} = 1$$

for $1 \leq i \leq s$ and there exists $t \in G$ such that

$$t_0 = t_1 = \cdots = t_s = t.$$

Theorem 4 shows that we may do this without loss of generality.

The secret logarithmic signature β can be obtained from the public key once t is known, since

$$\beta_{ij} = \gamma_{ij} t^{-1} \alpha_{ij}^{-1} t. \tag{3}$$

So we may think of the private key of the cipher as being the single group element t.

Define $\overline{t} \in G/\mathcal{Z}$ by $\overline{t} = t\mathcal{Z}$. Let $z \in \mathcal{Z}$. Replacing t by tz does not change the value of the right hand side of (3), and does not change the output of the decryption algorithm. So once $\overline{t}$ is known, the cryptosystem is broken as an equivalent private key can be derived efficiently. A search over all $|G/\mathcal{Z}|$ possibilities for $\overline{t}$ will therefore break the cipher. This cryptanalysis can be regarded as a generalisation of the 'attack on t_0' presented by Magliveras et al [13, Subsection 4.1] in the case of Suzuki 2-groups.

4.2 A more efficient approach

We would like to break the cipher much more efficiently than the attack in the previous subsection. We are most interested in the case when G is a Suzuki 2-group. However, in this subsection we consider a more general situation that includes these groups: the case when $G/\mathcal{Z}$ is abelian.

Let t' be a guess for the value of t. (Of course, it is only the coset $t'\mathcal{Z}$ that matters.) Define

$$\mathfrak{b}_{ij} = \gamma_{ij}(t')^{-1}\alpha_{ij}^{-1}t'$$

for all i and j. (Note that $\mathfrak{b}_{ij}$ can be computed without knowledge of the private key.) Define a cover $\mathfrak{b} = [\mathfrak{B}_1, \mathfrak{B}_2, \ldots, \mathfrak{B}_s]$ for some subset $\mathfrak{J}$ of G by

$$\mathfrak{B}_i = [\mathfrak{b}_{i1}, \mathfrak{b}_{i2}, \ldots, \mathfrak{b}_{ir_i}].$$

Let $q = |\mathcal{Z}|$, and define the map $\omega : \mathbb{Z}_q \to G$ by

$$\omega(x) = \breve{\gamma}(x)t'^{-1}\breve{\alpha}(x)^{-1}t'$$

for all $x \in \mathbb{Z}_q$. (Note that ω can also be computed without knowledge of the private key.)

When $t \equiv t' \bmod \mathcal{Z}$ (so our guess for t' is correct) we have that $\mathfrak{b} = \beta$ and $\omega = \breve{\beta} = \breve{\mathfrak{b}}$. In particular, when we have guessed correctly:

1. $\mathfrak{b}$ is a tame logarithmic signature for $\mathcal{Z}$, and
2. $\omega = \breve{\mathfrak{b}}$.

Lemma 5. *If the above two conditions are satisfied for a particular guess t', then $(\mathfrak{b}, (t', t', \ldots, t'))$ is an equivalent private key for the cipher.*

Proof. Since $\mathfrak{b}$ is a tame logarithmic signature for $\mathcal{Z}$, the pair $(\mathfrak{b}, (t', t', \ldots, t'))$ is a valid private key. Let (y_1, y_2) be the ciphertext obtained as encryption of the plaintext p under the public key corresponding to the private key $(\beta, (t, t, \ldots, t))$. So $y_1 = \breve{\alpha}(p)$ and $y_2 = \breve{\gamma}(p)$. Decryption using the key $(\mathfrak{b}, (t', t', \ldots, t'))$ gives us

$$\breve{\mathfrak{b}}^{-1}(y_2t'^{-1}y_1^{-1}t') = \omega^{-1}(y_2t'^{-1}y_1^{-1}t') = \omega^{-1}(\breve{\gamma}(p)t'^{-1}\breve{\alpha}(p)^{-1}t') = \omega^{-1}(\omega(p)) = p,$$

as required. □

Lemma 6. *Suppose that $G/\mathcal{Z}$ is abelian. For any choice of t' we have that $\mathfrak{b}$ is a cover of a subset of $\mathcal{Z}$.*

Proof. For any i and j we have that

$$\begin{aligned}\mathfrak{b}_{ij}\mathcal{Z} &= \gamma_{ij}(t')^{-1}\alpha_{ij}^{-1}t'\mathcal{Z} = \gamma_{ij}\alpha_{ij}^{-1}t'^{-1}t'\mathcal{Z} \\ &= \gamma_{ij}\alpha_{ij}^{-1}t^{-1}t\mathcal{Z} = \gamma_{ij}t^{-1}\alpha_{ij}^{-1}t\mathcal{Z} = \beta_{ij}\mathcal{Z} = \mathcal{Z}.\end{aligned}$$

So the elements of the cover $\mathfrak{b}$ all lie in $\mathcal{Z}$, as required. □

Lemma 7. *Suppose that $G/\mathcal{Z}$ is abelian. For any choice of t' we have that $\omega = \breve{\mathfrak{b}}$.*

Proof. By an abuse of notation we will identify the sets $\mathbb{Z}_q$ and $\mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_s}$ via the map λ_α defined in Subsection 2.1. So we think of the domain of the functions $\breve{\mathfrak{b}}$ and ω as being $\mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_s}$ rather than $\mathbb{Z}_q$.

We first note that

$$\mathfrak{b}_{i1} = \gamma_{i1}(t')^{-1}\alpha_{i1}^{-1}t' = (t')^{-1}t' = 1$$

for all i. In particular,

$$\breve{\mathfrak{b}}(x_1, x_2, \ldots, x_k, 1, 1, \ldots, 1) = \prod_{i=1}^{s} \mathfrak{b}_{ix_i} = \prod_{i=1}^{k} \mathfrak{b}_{ix_i}.$$

Moreover, writing $x = (x_1, x_2, \ldots, x_k, 1, 1, \ldots, 1)$, we find that

$$\breve{\alpha}(x) = \prod_{i=1}^{k} \alpha_{ix_i} \text{ and } \breve{\gamma}(x) = \prod_{i=1}^{k} \gamma_{ix_i},$$

since $\alpha_{i1} = \gamma_{i1} = 1$.

We will prove the lemma by induction. Let $P(k)$ be the following statement:

$$\omega(x_1, x_2, \ldots, x_s) = \breve{\mathfrak{b}}(x_1, x_2, \ldots, x_s) \text{ whenever } x_{k+1} = x_{k+2} = \cdots = x_s = 1.$$

The first paragraph of the proof shows that $\breve{\mathfrak{b}}(1, 1, \ldots, 1) = 1$. Moreover, since $\breve{\gamma}(1, 1, \ldots, 1) = \breve{\alpha}(1, 1, \ldots, 1)$ we find that $\omega(1, 1, \ldots, 1) = 1$. Hence $P(0)$ holds.

Assume, as an inductive hypothesis that $P(k-1)$ holds. Our assumption that $G/\mathcal{Z}$ is abelian implies that $\mathfrak{b}_{ij} \in \mathcal{Z}$ for all i and j. Let $x_1, x_2, \ldots, x_k$ be fixed. Define x and x' by

$$x = (x_1, x_2, \ldots, x_k, 1, \ldots, 1) \text{ and } x' = (x_1, x_2, \ldots, x_{k-1}, 1, \ldots, 1).$$

Then

$$\begin{aligned}
\breve{\mathfrak{b}}(x) &= \prod_{i=1}^{k} \mathfrak{b}_{ix_i} = \breve{\mathfrak{b}}(x')\mathfrak{b}_{kx_k} = \omega(x')\mathfrak{b}_{kx_k} \text{ (by our inductive hypothesis)} \\
&= \breve{\gamma}(x')t'^{-1}\breve{\alpha}(x')^{-1}t'\mathfrak{b}_{kx_k} \\
&= \breve{\gamma}(x')\mathfrak{b}_{kx_k}t'^{-1}\breve{\alpha}(x')^{-1}t' \text{ (since } \mathfrak{b}_{kx_k} \in \mathcal{Z}) \\
&= \left(\prod_{i=1}^{k-1} \gamma_{ix_i}\right) \gamma_{kx_k} t'^{-1}\alpha_{kx_k}^{-1}t't'^{-1} \left(\prod_{i=1}^{k-1} \alpha_{ix_i}\right)^{-1} t' \\
&= \breve{\gamma}(x)t'^{-1}\breve{\alpha}(x)t' \\
&= \omega(x).
\end{aligned}$$

So $P(k)$ is true whenever $P(k-1)$ is true. By induction, $P(s)$ holds and so the lemma follows. □

The following theorem is a consequence of Lemmas 5, 6 and 7 above.

Theorem 8. *Let G be such that $G/\mathcal{Z}$ is abelian. Then $(\mathfrak{b}, (t', t', \ldots, t'))$ is an equivalent private key for MST_3 if and only if $\breve{\mathfrak{b}} : \mathbb{Z}_{|\mathcal{Z}|} \to \mathcal{Z}$ is a bijection whose inverse is efficiently computable.*

So a general approach to finding a private key for MST_3 may be described as follows. We use the fact that $\breve{\mathfrak{b}}$ must be a bijection to derive some conditions on t'. If applying these conditions leads to a small number of possibilities for t', we perform an exhaustive search to find a private key that works. If there are still many possibilities for t', we choose one at random and hope that $\breve{\mathfrak{b}}^{-1}$ is efficiently computable: the probability that this will be successful will depend on the way that the logarithmic signature β has been generated. In the next subsection we will analyse the performance of this attack for Suzuki 2-groups.

4.3 Recovering the key in practice

Subsection 4.2 outlined a method for deriving an equivalent private key for the cipher. We now describe our computer experiments to verify that this method works in practice. All computer experiments were performed using the mathematics software SAGE [15].

Let $m = 81$. Our platform group is the Suzuki 2-group over the field $\mathbb{F}_q$, where $q = 2^m$. The generic attack described in Subsection 4.1 requires a search of size q to succeed: we fix $m = 81$ so that this generic attack is not feasible. Note that the public key is already rather long when $m = 81$: in the most efficient case we consider (Case 1 below), we need over 19 000 bits to store the non-identity elements in the logarithmic signatures α and γ. Our techniques do not seem to depend significantly on the automorphism θ in the definition of the Suzuki 2-group, so we fix θ to be the squaring automorphism in all our experiments.

We construct our logarithmic signature β using the ALTS method discussed in Subsection 2.4; this is the most general method we know of for generating a logarithmic signature on an elementary abelian 2-group. We wish to generate logarithmic signatures of type $(r_1, r_2, \ldots, r_s)$, where $\prod_{i=1}^{s} r_i = 2^m$. Note that the integers r_i must be fairly small, as otherwise the logarithmic signatures we produce cannot be stored efficiently. The precise method we use to generate β depends on its type: we give explicit details below. By Theorem 4, it is enough to consider logarithmic signatures that have an extra property: the elements β_{i1} are all equal to the identity. Our methods for generating β always produce logarithmic signatures with this property (and no generality is lost by generating logarithmic signatures in this way).

We follow the approach in Subsection 4.2 in our cryptanalysis. In the notation of that subsection, we begin by deriving conditions that t must satisfy as a consequence of the fact that β is bijective. We then choose t' at random subject to these conditions; our attack is successful if we obtain a valid private key after trying a small number of guesses t'. In our experiments, our attack was always successful.

Recall the notation $S(a,b)$ for an element in the Suzuki 2-group defined in Subsection 2.3. Our remark at the end of Subsection 2.5 shows that we may assume that $t = S(x,0)$ where $x \in \mathbb{F}_q$ is unknown, and so we restrict our guess t' to be of the form $S(y,0)$ for some $y \in \mathbb{F}_q$. The conditions on t that we derive are $\mathbb{F}_2$-linear conditions, so it is easy to choose t' satisfying these conditions at random. The precise conditions on t we derive will depend on the number of components r_i of the type of β that are equal to 2: when there are many of such components, the conditions we derive are weaker. For this reason, we provide three cases to illustrate our methods. In Case 1, $r_i = 2$ for all i. In this case we find no conditions on t, but simply randomly choosing a small number of values for t' leads to a successful attack. In Case 2, $r_i \neq 2$ for all i. In this case, we find that every condition we derive restricts t' to such a small number of possibilities that a negligible exhaustive search can be carried out. Case 3, with approximately half of the components of the type of β being equal to 2, illustrates an intermediate case. Here, each condition limits the number of possibilities for t' significantly (to approximately 2^{40} possibilities). Very few guesses t' can satisfy two of these conditions simultaneously, so combining two conditions allows us to derive an equivalent private key by a negligible exhaustive search.

Case 1: β has type $(2, 2, \ldots, 2)$

In this case, we assume β consists of 81 blocks of size 2. Such logarithmic signatures are very attractive from the perspective of efficiency: we only need to store the 81 non-trivial elements in the sets B_i; moreover these elements form a basis of $\mathcal{Z}$ when $\mathcal{Z}$ is considered as a 81-dimensional vector space over $\mathbb{F}_2$, and computations with β can be carried out using straightforward linear algebra. (We note that β is an example of *canonical logarithmic signature* as defined in [13]; however the attack described in that paper does not work in this particular case.)

We derive public and private keys for the MST_3 cryptosystem as follows. We randomly choose a generating set $\{z_1, \ldots, z_{81}\}$ for $\mathcal{Z}$. Define elements $d_{i2} \in \mathbb{F}_q$ by $z_i = S(0, d_{i2})$, so the elements d_{i2} form an $\mathbb{F}_2$-basis for $\mathbb{F}_q$. Set

$$\beta = [B_1, \ldots, B_{81}], \text{ where } B_i = \{1, S(0, d_{i2})\}.$$

We then generate elements $e_{i2}, f_{i2} \in \mathbb{F}_q$ at random, and define

$$\alpha = [A_1, \ldots, A_{81}], \text{ where } A_i = \{1, S(e_{i2}, f_{i2})\}.$$

Let $t = S(x,0)$ where $x \in \mathbb{F}_q$ is chosen at random. We construct γ as specified in the definition of MST_3. So we define

$$\begin{aligned}\gamma_{i2} &= \beta_{i2} t^{-1} \alpha_{i2} t \\ &= S(0, d_{i2}) S(x, x^\theta x) S(e_{i2}, f_{i2}) S(x, 0) \\ &= S(e_{i2}, d_{i2} + f_{i2} + e_{i2} x^\theta + e_{i2}^\theta x) =: S(e_{i2}, g_{i2}),\end{aligned}$$

no. guesses t'	1	2	3	4	5	6	7	8	9
frequency	2829	2111	1429	1048	799	490	374	279	181

no. guesses t'	10	11	12	13	14	15	16	17	18
frequency	133	98	66	47	31	26	19	11	5

no. guesses t'	19	20	21	22	23	24	25	26	27
frequency	3	7	7	4	2	1	0	0	0

Table 1: Experimental Results for Case 1

and set $\gamma = [C_1, \dots, C_{81}]$, where $C_i = \{1, \gamma_{i2}\}$.

Our attack works as follows. Let $t' = S(y, 0)$ be a random guess for t. We form $\mathfrak{b} = [\mathfrak{B}_1, \dots, \mathfrak{B}_{81}]$, where $\mathfrak{B}_i = \{1, \mathfrak{b}_{i2}\}$ and $\mathfrak{b}_{i2}$ is given by

$$\begin{aligned}\mathfrak{b}_{i2} &= \gamma_{i2} {t'}^{-1} \alpha_{i2}^{-1} t' \\ &= S(e_{i2}, g_{i2}) S(y, y^\theta y) S(e_{i2}, {e_{i2}}^\theta e_{i2} + f_{i2}) S(y, 0) \\ &= S(0, g_{i2} + f_{i2} + e_{i2} y^\theta + {e_{i2}}^\theta y).\end{aligned}$$

If the set $\{\mathfrak{b}_{i2}\}_{i=1}^{81}$ is linearly independent, then $\breve{\mathfrak{b}}$ is a bijection and it follows from Theorem 8 that we have an equivalent private key. If the set is linearly dependent, we repeat this process with another guess t'.

We have implemented this attack for 10 000 random instances of MST_3. The results of this experiment, which took a few minutes to carry out on a standard PC, are given in Table 1. The average number of guesses for t' before finding an equivalent private key, was approximately 3.47. Thus the scheme is insecure in this case.

Case 2: β has type $(8, 64, 64, \dots, 64)$

We now consider the case when our logarithmic signatures consist of one block of size 8 and thirteen blocks of size 64.

We construct β as follows. We generate a random basis $\{z_1, \dots, z_{81}\}$ for $\mathcal{Z}$. We consider the subgroup chain

$$1 = \mathcal{Z}_0 < \mathcal{Z}_1 < \cdots < \mathcal{Z}_{27} = \mathcal{Z},$$

where $Z_i = \langle z_1, \dots, z_{3i} \rangle$ for $1 \leq i \leq 27$. We form a transversal logarithmic signature of type $(8, 8, \dots, 8)$ (with 27 blocks in total), whose ith block is a transversal for Z_{i-1} in Z_i containing the identity as its first element. We then randomly amalgamate 26 blocks of size 8 in pairs to form 13 blocks of size 64. Reordering the blocks we have constructed an ATLS $\beta = [B_1, B_2, \dots, B_{14}]$ of type $(8, 64, 64, \dots, 64)$ for $\mathcal{Z}$. Define elements $d_{ij} \in \mathbb{F}_q$ by $\beta_{ij} = S(0, d_{ij})$.

We generate the element $t = S(x, 0)$, the elements $\alpha_{ij} = S(e_{ij}, f_{ij})$, the elements $\gamma_{ij} = S(e_{ij}, g_{ij})$, and the covers α and γ as in Case 1. In particular, the equation

$$g_{ij} = d_{ij} + f_{ij} + e_{ij}^\theta x + e_{ij} x^\theta$$

holds.

Our attack recovers a private key directly by a small exhaustive search, rather than guessing an equivalent private key. Lemma 1 implies that there exists i and j such that $j \geq 2$ and $B_i \cdot b_{ij} = B_i$. There is only a small number of possibilities for i and j so (using a negligible exhaustive search) we may assume that a valid choice for i and j are known. We know that

$$d_{ij} = g_{ij} + f_{ij} + e_{ij}^{\theta} x + e_{ij} x^{\theta}. \tag{4}$$

Moreover, when $B_i \cdot b_{ij} = B_i$, the equation

$$d_{ij} + d_{ik} = d_{il} \tag{5}$$

holds for at least $|B_i| - 2$ pairs of indices k, l where $2 \leq k, l \leq |B_i|$ and where j, k and l are distinct. Writing u_{ijkl} for $u_{ij} + u_{ik} + u_{il}$, equations (4) and (5) combine to give

$$g_{ijkl} + f_{ijkl} = {e_{ijkl}}^{\theta} x + e_{ijkl} x^{\theta}. \tag{6}$$

Note that the elements e_{ijkl}, f_{ijkl} and g_{ijkl} are all known (forming part of the public key (α, γ)), but x is unknown. For a fixed $e \in \mathbb{F}_q$, the map

$$\phi_e : \mathbb{F}_q \to \mathbb{F}_q \text{ given by } x \mapsto e^{\theta} x + e x^{\theta}$$

is an $\mathbb{F}_2$-linear map. Moreover, when $e \neq 0$, we have that ϕ_e has a kernel of size 2. Assuming (as is very likely) that e_{ijkl} is non-zero, we find that each equation of the form (6) is satisfied by at most two possibilities for x (and these choices are easily computed using elementary linear algebra). There are fewer than 2^{18} choices for i, j, k and l. Once these choices are fixed, there are at most 2 values for x that satisfy equation (6). So we can recover x by an exhaustive search though 2^{20} possibilities. (For each possibility for x, we can construct $\mathfrak{b}$ and check to see whether $\breve{\mathfrak{b}}$ is a bijection: this check can be carried out efficiently for an ATLS.)

Note that this attack makes use of the fact that $|B_i| > 2$ in an essential way: if $|B_i| = 2$ then there are no valid choices for j and k. Note also that when we have a correct value for i and j the same element x will occur at least $|B_i| - 2$ times as a solution to (6) as j and k vary over all possible values: this observation can be used to recover x more efficiently. Finally, we note that when i is guessed correctly the set B_i has the property that the product of its elements must be the identity (as the same is true for any coset of a subgroup of $\mathcal{Z}$ of order 4 or more); this property can be used to find x without the need to guess j, k or l.

We implemented the attack using SAGE on a standard PC, and in each run the randomly chosen secret value x was returned correctly within 30 minutes. Thus the MST_3 cryptosystem is also insecure in this case.

Case 3: β has type $(2, 2, \ldots 2, 16, 16, \ldots, 16)$

Finally, we consider the case when β consists of 41 sets of size 2 and 10 sets of size 16. In this situation, the analogue of equation (6) does not restrict the number of possibilities for x sufficiently, and so we combine two equations to recover x.

We construct β by starting with the subgroup chain

$$1 = \mathcal{Z}_0 < \mathcal{Z}_1 < \cdots < \mathcal{Z}_{61} = \mathcal{Z},$$

where each $\mathcal{Z}_i$ has index 2 in $\mathcal{Z}_{i+1}$ for $0 \leq i \leq 40$ and index 4 for $41 \leq i \leq 60$. We form a random transversal logarithmic signature for this chain (including the identity as the first element in each transversal): this logarithmic signature will consist of 41 sets of size 2 and 20 sets of size 4. We then amalgamate the 20 sets of size 4 in pairs to form 10 sets of size 16, where the pairing of these sets is chosen at random. The result is an ATLS $\beta = [B_1, B_2, \ldots, B_{41}, B_{42}, \ldots, B_{51}]$ of the type we are seeking. We then choose t and α, and construct γ, just as before.

Our attack in this case is as follows. Define the subgroup $H = \langle B_1, \ldots, B_{41} \rangle$. Write $\beta_{ij} = S(0, d_{ij})$, and define $V = \langle d_{i2} : 1 \leq i \leq 41 \rangle$. Note that V has dimension 41 and $H = \{S(0, v) : v \in V\}$. Clearly the image of $[B_{42}, \ldots, B_{51}]$ in $\mathcal{Z}/H$ is an ATLS for $\mathcal{Z}/H$ with no blocks of size 2. So we may proceed in the same way as in Case 2, this time working in the quotient $\mathcal{Z}/H$ to derive equations that x must satisfy modulo V. Using the notation from Case 2, we obtain equations of the form

$$g_{ijkl} + f_{ijkl} + V = \phi_{e_{ijkl}}(x), \tag{7}$$

where $\phi_{e_{ijkl}}$ is an $\mathbb{F}_2$-linear map. On the assumption that e_{ijkl} is non-zero, an equation of this form restricts x to lie in an affine subspace of dimension at most 42, and so we have reduced the size of an exhaustive search for x to 2^{42} possibilities. But a correct guess for i means that x satisfies at least $|B_i| - 2 \geq 2$ such equations as j, k and l vary. If we correctly guess two such combinations of j, k and l, we know that x lies in the intersection of two affine subspaces of dimension at most 42 (namely the solution sets corresponding to the two equations), and this reduces the number of possibilities for x to a negligible number. The validity of each possibility for x can be determined by checking the bijectivity of $\breve{\mathfrak{b}}$ as in Case 2.

Implementing these ideas, we generated 1000 random ATLSs for $\mathcal{Z}/H$. For each ATLS we picked a random pair of equations (7) where the indices i, j, k and l have been guessed correctly, and computed the size of the intersection of the two solution sets. We did the same when the indices have been guessed incorrectly, to check that the number of possibilities for x is not too large in this case. We record the results in Table 2. As Table 2 indicates, in either case the number of possibilities for x is small. There are less than 2^{20} pairs of equations to check, and so we typically expect (guided by Table 2) an exhaustive search for x to be of size 2^{24} at most. (Furthermore, within this search we expect x to occur with a relatively high frequency, since it appears for *every* correct pair of equations (7).) Thus we conclude that the MST_3 cryptosystem is also insecure in this case.

No. possibilities for x	0	1	2	4	8	16
Freq. (correct indices)	0	579	386	33	2	0
Freq. (incorrect indices)	276	543	170	10	1	0

Table 2: Experimental Results for Case 3

5 Conclusion

We have described the analysis of MST_3, a public key cryptosystem based on non-commutative groups recently proposed by Lempken, Magliveras, van Trung and Wei [9]. We have given a generic attack on the scheme (in the sense of being independent of the platform group) which substantially reduces the estimate originally given in [9] for the complexity for attacking the scheme. We have presented attacks on MST_3 that operate for a wide class of platform groups and work under the assumption that private keys are generated in the most general way known to the authors. We successfully implemented these attacks when the proposed platform group suggested in [9] is used, and under a wide variety of methods for generating the private key. Thus we conclude that, until a method for generating secure tame logarithmic signatures in this context is invented, the MST_3 cryptosystem is insecure.

References

[1] D. Bernstein, J. Buchmann, E. Dahmen (Eds.) Post-Quantum Cryptography Springer-Verlag, Berlin Heidelberg, 2009.

[2] S. Blackburn, S. Murphy and J. Stern. The Cryptanalysis of a Public Key Implementation of Finite Group Mappings. *J. Cryptology* **8** (3) 157-166 (1995).

[3] J.M. Bohli, R. Steinwandt, M.I. González Vasco and C. Martínez. Weak keys in MST 1. *Designs, Codes and Cryptography*, **37** (3) 509-524 (2005).

[4] M.I. González Vasco, A. L. Perez del Pozo, and P. Taborda Duarte. A note on the security of MST_3. Cryptology ePrint Archive, `http://eprint.iacr.org`, Report 2009/096 (2009).

[5] M.I. González Vasco, M. Rötteler and R. Steinwandt. On minimal length factorizations of finite groups. *Experimental Math.*, **12** (1), 1-12 (2003).

[6] M.I. González Vasco and R. Steinwandt. Obstacles in two public key cryptosystems based on group factorizations. *Tatra Mt. Math. Publ*, **25** (23), 23-37 (2002).

[7] G. Higman. Suzuki 2-groups. *Ill. J. Math*, **7**, 79-96 (1963).

[8] P.E. Holmes. On minimal factorisations of sporadic groups. *Experimental Math.*, **13** (4), 435-440 (2004).

[9] W. Lempken, S.S. Magliveras, T. van Trung, and W. Wei. A public key cryptosystem based on non-abelian finite groups. *J. Cryptology*, **22**(1), 62-74 (2009).

[10] W. Lempken and T. van Trung. On minimal logarithmic signatures of finite groups. *Experimental Math.*, **14** (3), 257-269 (2005).

[11] S.S. Magliveras and N.D. Memon. Algebraic Properties of cryptosystem PGM. *J. Cryptology*, **5** (3) 167-183 (1992).

[12] S.S. Magliveras, D.R. Stinson and T. van Trung. New approaches to designing public key cryptosystems using one-way functions and trap-doors infinite groups. *Journal of Cryptology*, **15**, 285-297 (2002).

[13] S.S. Magliveras, P. Svaba, T. van Trung and P. Zajac. On the security of a realization of cryptosystem MST_3 *Tatra Mt. Math. Publ.*, **41** 65-78 (2008).

[14] M. Qu and S.A. Vastone. Factorizations in the Elementary Abelian p-Group and Their Cryptographic Significance. *Journal of Cryptology*, **7**(4), 201-212 (1994).

[15] SAGE Mathematical Software, Version 3.4.1, `http://www.sagemath.org`

[16] P. Svaba, T. van Trung. On generation of random covers for finite groups. Institut für Experimentelle Mathematik, Universität Duisburg-Essen, `http://www.exp-math.uni-essen.de/preprints/RanCoversTran.pdf` Preprint (2006).