New approaches to designing public key cryptosystems using one-way functions and trap-doors in finite groups

S. S. Magliveras
Department of Computer Science and Engineering
University of Nebraska, Lincoln NE 68588 *
spyros@helios.unl.edu

D.R. Stinson
Department Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario †
Canada N2L 3G1
dstinson@uwaterloo.ca

and

Tran van Trung
Institute for Experimental Mathematics, University of Essen
Ellernstrasse 29, 45326 Essen, Germany
trung@exp-math.uni-essen.de

Abstract

A symmetric key cryptosystem based on *logarithmic signatures* for finite permutation groups was described by the first author in [6], and its algebraic properties were studied in [7]. In this paper we describe two possible approaches to the construction of new public key cryptosystems with message space a large finite group G, using logarithmic signatures and their generalizations. The first approach relies on the fact that permutations of the message space G induced by transversal logarithmic signatures almost always generate the full symmetric group $\mathcal{S}_G$ on the message space. The second approach could potentially lead to new ElGamal - like systems based on trap-door, one-way functions induced

*Research supported in part by National Science Foundation grant CCR-9610138
†Research supported in part by NSERC grants IRC #216431-96 and RGPIN # 203114-98.

by logarithmic signature-like objects we call *meshes*, which are *uniform covers* for G.

Key words. Trap-door one-way functions, group factorizations, public key cryptosystems, finite groups.

1 Introduction

At the writing of this paper, only a few known public key cryptosystems remain unbroken. All of these are based on the perceived difficulty of certain problems in particular large finite abelian groups. Whether the underlying group is the multiplicative group of units in the ring $\mathbb{Z}_{pq}$, where p and q are primes, the multiplicative group of a finite field, or a cyclic component of an elliptic curve, the theoretical foundations for the known system lie in the intractability of problems closer to number theory than group theory. These known public key systems usually operate in the context of a cumbersome arithmetic resulting in very low data rates.

In this paper we describe how, in principle, we can use finite groups to build two new public key cryptosystems. When the underlying groups are permutation groups, composition of two permutations can be done in one machine cycle in specially designed hardware, and therefore the new systems are very fast. In addition to the data rate difference, there is a fundamental difference between the known systems and the ones we propose. While the former are based on large cyclic groups, our systems are based on large non-solvable groups of much higher structural complexity than that of a cyclic group.

In Section 2 we present some preliminaries. In Section 3 we discuss transformations and a brief classification scheme for logarithmic signatures. We also present a basic result from [7] which states that, almost always, permutations induced by *transversal* logarithmic signatures generate the full symmetric group $\mathcal{S}_{|G|}$. In Section 4 we discuss our first proposed public key system MST_1 which is based on the fact that one-way permutations induced by non-transversal logarithmic signatures can be factorized as the product of permutations induced by transversal logarithmic signatures. Such a factorization constitutes a trap-door. We propose algorithms for finding such trap-doors in the same section. In Section 5, we define what we call an $[s, r] - mesh$ in an arbitrary group G, propose our second public key system MST_2, and discuss its security. We stress that these systems are currently hypothetical, in the sense that we do not have efficient algorithms for parameter generation for secure versions of the proposed systems.

2 Preliminaries

We define the *degree* of an abstract finite group G to be the integer n such that $|G| = \lfloor n \log n \rfloor$. For permutation groups, however, the *degree* n has the usual meaning, independently of $|G|$. We denote by $G^{[\mathbb{Z}]}$ the collection of all finite sequences in G, and view the elements of $G^{[\mathbb{Z}]}$ as single-row matrices with entries in G. Under ordinary tensor product of matrices, $G^{[\mathbb{Z}]}$ is a monoid. The following example illustrates the operation,

$$[x_1, x_2, x_3] \otimes [y_1, y_2] = [x_1y_1, x_1y_2, x_2y_1, x_2y_2, x_3y_1, x_3y_2].$$

We simplify notation however, and for $X, Y \in G^{[\mathbb{Z}]}$ we write $X \cdot Y$ or XY for $X \otimes Y$. If $X = [x_1, x_2, \ldots, x_r] \in G^{[\mathbb{Z}]}$, the *length* r of X is denoted by $|X|$, and $\overline{X}$ denotes the element $\sum_{i=1}^{r} x_i$ in the group ring $\mathbb{Z}G$. It is immediate that $\overline{XY} = \overline{X}\,\overline{Y}$ and $|XY| = |X|\,|Y|$, for any $X, Y \in G^{[\mathbb{Z}]}$.

Let G be a finite group. Suppose that $\alpha = [A_1, A_2, \ldots, A_s]$ is a sequence of $A_i \in G^{[\mathbb{Z}]}$, such that $\sum_{i=1}^{s} |A_i|$ is bounded by a polynomial in the degree of G. Moreover, let

$$\overline{A_1} \cdot \overline{A_2} \cdots \overline{A_s} = \sum_{g \in G} a_g g\,, \quad a_g \in \mathbb{Z} \tag{2.1}$$

Then, we say that α is

i) a *pseudo logarithmic signature* for G if $\prod_{i=1}^{s} |A_i| = |G|$.

ii) a *cover* for G if for all $g \in G$, $a_g > 0$.

iii) a λ-*quasi-logarithmic signature* for G, if $a_g \in \{\lambda, \lambda + 1\}$ for each $g \in G$.

iv) a λ-*logarithmic signature* for G, if $a_g = \lambda$ for each $g \in G$.

v) a *quasi-logarithmic signature* for G, if α is a 1-*quasi-logarithmic signature* for G, and

vi) a *logarithmic signature* for G, if α is a 1-*logarithmic signature* for G.

For the most part we are concerned with objects of type vi). Note that if $\alpha = [A_1, \ldots, A_s]$ is a logarithmic signature for G, then, each element $y \in G$ can be expressed uniquely as a product of the form

$$y = q_1 \cdot q_2 \ldots q_{s-1} \cdot q_s \tag{2.2}$$

for $q_i \in A_i$.

Of course, for general covers the factorization in (2.2) is not unique, and the problem of finding a factorization for a given $y \in G$ is in general intractable.

Let $\alpha = [A_1, \ldots, A_s]$ be a logarithmic signature for G with $r_i = |A_i|$, then, the A_i are called the *blocks* of α and the vector of block lengths $(r_1, \ldots, r_s)$ the *type* of α. We define the *length* of α to be the integer $\ell = \sum_{i=1}^{s} r_i$. We say that α is *nontrivial* if $s \geq 2$ and $r_i \geq 2$ for $1 \leq i \leq s$; otherwise α is said to be *trivial.* A logarithmic signature is called *tame* if the factorization in equation (2.2) can be achieved in time polynomial in the degree n of G; it is called *supertame* if the factorization can be achieved in time $O(n^2)$. The existence of supertame logarithmic signatures is discussed in [7]. A logarithmic signature is called *wild* if it is not tame. We denote by $\mathcal{C}(G)$, $\Omega_\lambda(G)$, and $\Lambda_\lambda(G)$ the respective collections of *covers*, *λ-quasi-logarithmic signatures*, and *λ-logarithmic signatures* of G. We write $\Omega = \Omega_1 = \Omega_1(G)$, and $\Lambda = \Lambda_1 = \Lambda_1(G)$.

In some occasions, we represent a logarithmic signature $\alpha = [A_1, \ldots, A_s]$ of type $(r_1, \ldots, r_s)$ by an $s \times r$ matrix

$$\alpha = (a_{i,j}) \tag{2.3}$$

where $r = \max\{r_i\}$, $a_{i,j} = A_i(j)$ for $1 \leq j \leq r_i$, and $a_{i,j} = 0$ for $j > r_i$.

Proposition 2.1 *In the class $\mathcal{G}$ of finite groups there are instances (G, α), $\alpha \in \mathcal{C}(G)$, where the factorization in* (2.2) *is equivalent to solving the discrete logarithm problem in G.*

Proof: Let G be a cyclic group, written multiplicatively, and let f be a generator of G. Further, let s the least positive integer such that $2^{s-1} \leq |G| < 2^s$. If $\alpha = [A_1, A_2, \ldots, A_s]$, where $A_i = [1, f^{2^{i-1}}]$, then $\alpha \in \Omega_1(G)$, and factorization with respect to α is equivalent to solving the discrete logarithm problem (DLP) in G with respect to the generator f. □

If $\alpha = [A_1, \ldots, A_s]$ is a logarithmic signature for a group G, then the sequence $A_1 \otimes \ldots \otimes A_s$ contains each element of G exactly once. Thus, α induces a bijection $\breve{\alpha} : \mathbb{Z}_{|G|} \rightarrow G$. The following proposition is rather obvious.

Proposition 2.2 *Suppose that α is a logarithmic signature of a finite group G, then, $\breve{\alpha}$ is always efficiently computable. However, $\breve{\alpha}^{-1}$ is efficiently computable if and only if α is tame.*

Proof: Suppose that $\alpha = [A_1, A_2, \ldots, A_s]$ is a logarithmic signature of type $(r_1, r_2, \ldots, r_s)$, with $A_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,r_i}]$. The canonical isomorphism τ from $\mathbb{Z}_{r_1} \oplus \mathbb{Z}_{r_2} \oplus \cdots \oplus \mathbb{Z}_{r_s}$ onto $\mathbb{Z}_{|G|}$ is compatible with $\otimes$, and both τ and τ^{-1} are efficiently computable (also see [7]). Thus, given $x \in \mathbb{Z}_{|G|}$, to compute $\breve{\alpha}(x)$ first compute

$\tau^{-1}(x) = (j_1, j_2, \ldots, j_s)$, and then compute $\breve{\alpha}(x) = a_{1,j_1} \cdot a_{2,j_2} \cdots a_{s,j_s}$. So, $\breve{\alpha}$ is efficiently computable. Conversely, given α and an element $y \in G$, to determine $\breve{\alpha}^{-1}(y)$ it is necessary to obtain the factorization (2.2) for y and determine the indices $j_1, j_2, \ldots, j_s$ such that $y = a_{1,j_1} \cdot a_{2,j_2} \cdots a_{s,j_s}$. This is possible if and only if α is tame. Once the vector $(j_1, j_2, \ldots, j_s)$ has been determined, $\breve{\alpha}^{-1}(y) = \tau(j_1, j_2, \ldots, j_s)$ can be computed efficiently. $\square$

The following statement follows naturally:

Proposition 2.3 *Let G be a finite group, α a* wild *logarithmic signature and β a* tame *logarithmic signature for G, then the mapping $\breve{\alpha}\breve{\beta}^{-1} : \mathbb{Z}_{|G|} \to \mathbb{Z}_{|G|}$ is a one-way permutation.*

Abusing language somewhat we use the phrase "α can be inverted" to mean that $\breve{\alpha}$ can be inverted efficiently, i.e. that the factorization in (2.2) is achievable in polynomial time.

Definition 2.1 *Two logarithmic signatures α, β of G are said to be equivalent if $\breve{\alpha} = \breve{\beta}$.*

3 Classes and Transformations

In this section we briefly discuss *classes* of logarithmic signatures and basic *transformations* on logarithmic signatures for a group G. When we discuss matters involving computational complexity, we further assume that G is a permutation group of degree n.

Let $\gamma : 1 = G_0 < G_1 < \cdots < G_{s-1} < G_s = G$ be a chain of subgroups of G, and A_i an ordered, complete set of right coset representatives of G_{i-1} in G_i. It is clear that $[A_1, A_2, \ldots, A_s]$ forms a logarithmic signature for G. Such a logarithmic signature is called *exact-transversal* with respect to γ. The following proposition is an easy corollary of a theorem by Furst, Hopcroft, and Luks, [2]. For the proof see [7].

Proposition 3.1 *Let G be a finite permutation group of degree n, and α an exact-transversal logarithmic signature of length polynomial in n. Then, α is tame.*

We denote the set of all exact-transversal logarithmic signatures of G by $\mathcal{E}$. Recall that the order of a permutation group of degree n, generated by a polynomial in n number of generators can be computed in time polynomial in n [2].

Proposition 3.2 *Let G be a finite permutation group of degree n, and suppose that $\alpha = [A_1, A_2, \ldots, A_s]$ is a pseudo logarithmic signature for G. Then, there is a polynomial time algorithm which decides whether $\alpha \in \mathcal{E}$.*

Proof: For $k = 1, \ldots, s$ let $V_k = \langle A_1 \cup \cdots \cup A_k \rangle$. We first check that $V_1 = A_1$, i.e. that A_1 is a subgroup. This takes polynomial time by using [2], or simply testing closure. Now, suppose we have verified in polynomial time that $|V_k| = |V_{k-1}| \cdot |A_k|$. Using the Schreier-Sims [9] or similar algorithm we build "strong generators" for V_{k+1} using $A_1 \cup \cdots \cup A_k \cup A_{k+1}$ as the initial generating set and verify that $|V_{k+1}| = |V_k| \cdot |A_{k+1}|$. Both s and the time taken in the k^{th} step are bounded by a polynomial in n . □

Different types of transformations on logarithmic signatures have been considered in [7] and [8]. Here, we consider just two. The first type, called the *shuffle* or *monomial shuffle* , derives new logarithmic signatures from a given exact-transversal by changing the coset representatives, and permuting the elements within each block. The second type of transformation can be described as follows: Suppose that $\alpha = [A_1, A_2, \ldots, A_s] \in \Lambda$. Let $g_0, g_1, ..., g_s \in G$, and consider the sequence $\beta = [B_1, B_2, \ldots, B_s]$, where $B_i = g_{i-1}^{-1} A_i g_i$. It is easy to see that β is a logarithmic signature for G. When $g_0 = g_s = 1$ we say that β is a *sandwich* of α. When $g_0 = 1$ β is said to be a *right translation* of α by g_s. If $g_s = 1$, then β is called a left tralslation of α by g_0. The following proposition is proved in [7].

Proposition 3.3 *Let α and β be two logarithmic signatures of G, both of type $(r_1, r_2, \ldots, r_s)$. Then, α and β are equivalent if and only if one is a sandwich of the other.*

A logarithmic signature α for G is called *transversal* if and only if it is the sandwich of an exact-transversal logarithmic signature for G. We denote the set of all transversal logarithmic signatures of G by $\mathcal{T}$.

The following proposition says that we can decide in polynomial time whether a given logarithmic signature is transversal or not. In the proof we present an algorithm for accomplishing the task. Recall that the order of a permutation group of degree n, generated by a polynomial in n number of generators can be computed in time polynomial in n [2].

Proposition 3.4 *Let G be a finite permutation group of degree n, and suppose that $\alpha = [A_1, A_2, \ldots, A_s]$ is a logarithmic signature for G. Then, there is a polytime algorithm to determine whether α is transversal and, if so, determine an exact transversal β equivalent to α.*

Proof: We describe the algorithm for $s \geq 3$ and omit a proof of correctness. In the first step, replace $[A_1, A_2, A_3, \ldots, A_s]$ by $[B_1, C_2, A_3, \ldots, A_s]$ where $B_1 = A_1 x_1^{-1}$, $C_2 = x_1 A_2$ and $x_1 = A_1(1)$. The k^{th} step, $k = 2, \ldots, s-1$ replaces $[B_1, B_2, \ldots, B_{k-1}, C_k, A_{k+1}, A_{k+2}, \ldots, A_s]$ by $[B_1, B_2, \ldots, B_{k-1}, B_k, C_{k+1}, A_{k+2}, \ldots, A_s]$, where $B_k = C_k x_k^{-1}$, $C_{k+1} = x_k A_{k+1}$, and $x_k = C_k(1)$... We finally test whether the resulting logarithmic signature $\beta = [B_1, B_2, \ldots, B_{s-1}, C_s]$ is an exact transversal. If $\beta \in \mathcal{E}$ then $\alpha \in \mathcal{T}$ and the x_i provide the sandwiching transformation. If $\beta \notin \mathcal{E}$ then α is not transversal. If $s = 2$, we have a much easier task: replace $\alpha = [A_1, A_2]$ by $\beta = [B_1, C_2]$ where $B_1 = A_1 x_1^{-1}$, $C_2 = x_1 A_2$, $x_1 = A_1(1)$ and perform the final test as for the $s \geq 3$ case. $\square$

The process of obtaining $[B_1, B_2, \ldots, B_{s-1}, C_s]$ from $[A_1, A_2, \ldots, A_s]$ by appropriate sandwiching is called *normalization.* Normalization of a transversal logarithmic signature produces an equivalent exact transversal.

The following is a consequence of propositions (3.1) and (3.4).

Corollary 3.1 *Any transversal logarithmic signature of a finite permutation group G is tame.*

It follows therefore that the class $\mathcal{W}$ of wild logarithmic signatures is a subclass of the class $\mathcal{NT}$ of *non-transversal* logarithmic signatures of G. We are also interested in the following definition:

Definition 3.1 *A subset S of a finite group G is called* periodic *if S is the union of a number of right cosets (left cosets) of a non-trivial subgroup H of G, otherwise S is called* aperiodic.

There are two other classes of signatures which are farther from being transversal than logarithmic signatures in $\mathcal{NT}$. One is the class of *totally non-transversal* ($\mathcal{TNT}$) logarithmic signatures. A logarithmic signature α is said to be *totally-non-transversal* if each block of α is not a coset of a non-trivial subgroup of G. Moreover, a logarithmic signature is called *totally-aperiodic* ($\mathcal{TA}$) if each block is an aperiodic set in G. Clearly, $\mathcal{TA} \subset \mathcal{TNT} \subset \mathcal{NT} = \Lambda \setminus \mathcal{T}$.

We adopt the cryptographic assumption that logarithmic signatures which are far from being transversal are difficult to invert, i.e. that logarithmic signatures in $\mathcal{TNT}$ or $\mathcal{TA}$ are "wild-like". Experimentation with relatively small groups shows that there are many more $\mathcal{TNT}$ logarithmic signatures, than transversal ones. Moreover, we know of no theoretical reasons why in general there should exist polynomial time algorithms to invert $\mathcal{TNT}$ logarithmic signatures. On the contrary, proposition (2.1) supports our assumption.

We now identify the elements of G with the elements in $\mathbb{Z}_{|G|}$ by selecting a fixed supertame logarithmic signature η. Then $g \in G$ corresponds to $\eta^{-1}(g) \in \mathbb{Z}_{|G|}$. Once η has been selected, each logarithmic signature α gives rise to a computable permutation $\hat{\alpha}$ of $\mathbb{Z}_{|G|}$ defined by $\hat{\alpha} = \breve{\alpha}\breve{\eta}^{-1}$. If $\mathcal{F} \subset \Lambda$, we write $\hat{\mathcal{F}} = \{\hat{\alpha} : \alpha \in \mathcal{F}\}$. The following theorem is proved in [7]:

Theorem 3.1 *If G is a finite non-hamiltonian group with $|G|$ different from q, $(1+q^2)$, $(1+q^3)$, $\frac{(q^n-1)}{(q-1)}$, $2^{n-1}(2^n \pm 1)$, 11, 12, 15, 22, 23, 24, 176, 276 where q is a prime power and n a positive integer, then the group $\langle\hat{\mathcal{T}}\rangle$ generated by $\hat{\mathcal{T}}$ is the full symmetric group $\mathcal{S}_{|G|}$.*

Theorem (3.1) is rather technical, but has important consequences. Essentially, the theorem says that *almost always* the giant group $\mathcal{S}_{|G|}$ is generated by $\hat{\mathcal{T}} = \hat{\mathcal{E}}$, i.e. by the collection of permutations induced by exact-transversal logarithmic signatures. Thus, any permutation $\sigma \in \mathcal{S}_{|G|}$ can be written as the composition of permutations $\hat{\theta}_i$ induced by transversal logarithmic signatures. Note, moreover, that the conclusion of the theorem is independent of the choice of $\eta \in \mathcal{T}$. This follows from the simple observation that for $\theta \in \mathcal{T}$, $\alpha \in \Lambda$, $\breve{\alpha}\breve{\theta}^{-1} = \breve{\alpha}\breve{\eta}^{-1}\breve{\eta}\breve{\theta}^{-1} = \hat{\alpha}\hat{\theta}^{-1}$.

Incidentally, experimentation in groups of small order shows that the conclusion of Theorem (3.1) is true even when its hypotheses fail to hold.

4 First potential public key system $\mathbf{MST_1}$

Suppose now that α is a wild, and β a tame logarithmic signature for a finite permutation group G. By proposition (2.3), $\sigma = \hat{\alpha}\hat{\beta}^{-1}$ is a one-way permutation in $\mathcal{S}_{|G|}$. Then, by Theorem (3.1), σ can be written as the product of a finite (hopefully small) number of elements in $\hat{\mathcal{E}} = \hat{\mathcal{T}}$ and their inverses. Because the mappings induced from transversal logarithmic signatures can be inverted efficiently, such factorizations, if they could be efficently computed, could be used as trap-doors for a public key cryptosystem.

Suppose that the factorization of σ as the composition of elements of $\hat{\mathcal{T}}$ is known only by Alice. Then, Alice can efficiently invert σ but no-one else can. Now, given the one-way permutation σ above, the problem of factoring σ as the composition of transversal $\hat{\theta}_i$ is in general hard, otherwise if we had a general, efficient algorithm for factoring σ, we would in principle be able to solve DLP. This is the basis for cryptosystem $\mathbf{MST_1}$ we shall now discuss.

Our system is described as follows: User "Alice" is given a pair of logarithmic signatures (α, β) where α is in $\mathcal{TNT}$, and β is transversal. Alice is also given the factorization of $\sigma = \hat{\alpha}\hat{\beta}^{-1}$ as the composition $\sigma = \hat{\theta}_1 \cdots \hat{\theta}_k$ where the θ_i are exact-transversal, and where k is a small integer ≥ 2. Alice publishes (α, β) and G as her public key, but keeps $\theta_1, \ldots, \theta_k$ as her secret key. Since the θ_i are transversal, Alice can efficiently compute $\hat{\theta_i}^{-1}$, and therefore can compute efficiently σ^{-1}. The message and cipher space are $Z_{|G|}$. To send a message $m \in Z_{|G|}$ to Alice, Bob encrypts m as $c = \sigma(m) = [\hat{\alpha}\hat{\beta}^{-1}](m)$ and transmits c to Alice. Upon receiving c, Alice decrypts c by computing $\hat{\theta}_k^{-1}(\hat{\theta}_{k-1}^{-1}(\ldots(\hat{\theta}_1^{-1}(c)\ldots) = m$.

4.1 The smallest case

The smallest group which has non-transversal logarithmic signatures is the cyclic group $\mathbb{Z}_8$. Although the order of a group to be used in actual practice would be much larger (say 48!), $\mathbb{Z}_8$ illustrates the existence of non-transversal logarithmic signatures which are products of a small number of transversal logarithmic signatures. In particular, in $\mathbb{Z}_8$ there are 512 permutations of $\mathcal{S}_8$ induced by transversal logarithmic signatures and 640 by non-transversal, of which 384 are in $\mathcal{T}\hat{\mathcal{N}}\mathcal{T}$. Any one of the 384 can be written as the product of two transversals, and any one of the 640 can be written as the product of at most three transversals.

4.2 Algorithms for building a trapdoor

An analysis of the type undertaken in the previous section is fortunately not feasible in the general case. Even when G is the tiny group A_4 of order 12 the number of permutations of $\mathcal{S}_{12}$ induced by logarithmic signatures of A_4 is $|\hat{\Lambda}| = 304,128$. In the general case, to compute and store all induced permutations in $\hat{\Lambda}$, take their compositions and determine which products are also induced by non-transversal logarithmic signatures are hard problems. Recall that each permutation is of degree $|G|$, and in general, $|G|$ will be of exponential order in n.

In this section we discuss algorithms for constructing trap doors of the type discussed in section 4. In what follows we assume that the elements of G are identified with the elements of $Z_{|G|}$ by means of a bijection $\breve{\eta} : Z_{|G|} \to G$, where η is a supertame logarithmic signature of G.

In what follows for a logarithmic signature α of a finite group G, Σ_α denotes an oracle which computes $\hat{\alpha}$, i.e. when a querry $i \in \mathbb{Z}_{|G|}$ is presented to Σ_α, it responds with $\hat{\alpha}(i)$. We proceed to show how one can reconstruct α, or an equivalent, by a polynomial number of queries to Σ_α.

Proposition 4.1 *Let G be a permutation group of degree n, and suppose that $\sigma = \hat{\alpha}$ is the permutation induced by a secret logarithmic signature α. Suppose that the type $(r_1, \ldots, r_s)$ of α is known, and that oracle $\Sigma = \Sigma_\alpha$ is public. Then,* *a logarithmic signature β equivalent to α can be reconstructed using a polynomial number of queries.*

Proof: Let τ be the canonical isomorphism from $\mathbb{Z}_{r_1} \oplus Z_{r_2} \oplus \cdots \oplus Z_{r_s}$ onto $\mathbb{Z}_{|G|}$, and η the logarithmic signature selected so that $\hat{\alpha} = \breve{\alpha}\breve{\eta}^{-1}$. The existence of a logarithmic signature α inducing σ is known, and so is the type of α. In what follows we will determine the entries of α or an equivalent. The logarithmic signature we seek in the form of matrix (2.3) is:

$$\alpha = \begin{pmatrix} a_{1,1} & \dots & a_{1,r_1} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ a_{2,1} & \dots & * & * & \dots & a_{2,r_2} & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{s-1,1} & \dots & * & \dots & a_{s-1,r_{s-1}} & 0 & 0 & 0 & \dots & 0 \\ a_{s,1} & \dots & * & * & \dots & * & a_{s,r_s} & 0 & \dots & 0 \end{pmatrix},$$

where the $a_{i,j}$ are indeterminates to be replaced systematically by elements of the group. We begin by setting all $a_{i,j}$ to 0. A zero entry means that the corresponding group element has not yet been determined. All $a_{i,j}$ for $j > r_i$ will remain zero. By sandwitching, we can now assume that $a_{1,1} = a_{2,1} = \ldots = a_{s-1,1} = 1 \in G$, the group identity. Thus our matrix becomes:

$$\alpha = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix},$$

Iterative step: Now select an element $x \in Z_{|G|}$ such that for $\tau^{-1}(x) = (j_1, j_2, \ldots, j_s)$ all but one of the a_{i,j_i} are non-zero, and for exactly one index, say j_q, $a_{q,j_q} = 0$. We then have:

$$a_{1,j_1} a_{2,j_2} \cdots a_{s,j_s} = \breve{\alpha}(x) = \breve{\eta}(\hat{\alpha}(x)) = u \tag{4.4}$$

For simplicity, letting $z_k = a_{k,j_k}$ yields:

$$z_1 z_2 \cdots z_{q-1} z_q z_{q+1} \cdots z_{s-1} z_s = u \tag{4.5}$$

In equation (4.5) $u = \breve{\alpha}(x)$ is known and all z_i except for z_q are known. We now solve for a_{q,j_q} by:

$$a_{q,i_q} = z_q = z_{q-1}^{-1} \cdots z_2^{-1} z_1^{-1} u \, z_s^{-1} z_{s-1}^{-1} \cdots z_{q+1}^{-1}. \tag{4.6}$$

Having computed a_{q,j_q} uniquely, we replace the zero entry in the matrix by the group element a_{q,j_q}, and go to the Iterative step for as long as there is a new element $x' \in Z_{|G|}$ for which $\tau^{-1}(x') = (j'_1, j'_2, \ldots, j'_s)$ is such that all a_{i,j'_i} are non-zero in the matrix except for a single 0. This becomes impossible only when all $a_{i,j}$, $1 \leq j \leq r_i$ have been turned to non-zeroes, i.e. to elements of G.

The above process will require exactly $1-s+\sum_1^s r_i$ passes through the iterative step, and this is a necessary and sufficient amount of information needed to determine α. $\square$

The above process determines α uniquely (up to equivalence) and all degrees of freedom are removed in the initial sandwitching operation which sets $a_{1,1}$, $a_{2,1}$, ..., $a_{s-1,1}$ to $1 \in G$.

Suppose now that a small set of transversal logarithmic signatures $\theta_1, \ldots, \theta_k$ is selected in $\mathcal{E}$. Does there exist a non-transversal logarithmic signature γ which is the product $\theta_1 \cdots \theta_k$? Let $\pi = \hat{\theta}_1 \cdots \hat{\theta}_k$. Having selected the set of θ_i we can efficiently compute $\pi(x)$ for any $x \in Z_{|G|}$. For each possible type $(r_1, r_2, \ldots, r_s)$ of G, compute a *pseudo logarithmic signature* $\gamma = [A_1, \ldots, A_s]$ for G by using the algorithm described in the proof of proposition (4.1). If there is a logarithmic signature which induces π for this type, then it has to be γ (up to equivalence.). Otherwise γ will generate a "random" function $\hat{\gamma} : Z_{|G|} \rightarrow Z_{|G|}$. Using the "birthday paradox" principle we can test with high probability whether or not $\hat{\gamma}$ is a permutation.

If the above test concludes that $\hat{\gamma}$ is a permutation, then γ is indeed a logarithmic signature for G. It is either transversal or non-transversal. But we have a polynomial time algorithm which decides whether a given logarithmic signature is transversal or not. If γ is non-transversal we further test whether $\gamma \in \mathcal{TNT}$. This can be done in polynomial time by testing whether any block B of γ is a coset of some subgroup. Note that B is a coset if and only if BB^{-1} and $B^{-1}B$ are subgroups. If $\gamma \in \mathcal{TNT}$ then we have found a trap door as desired. Otherwise, we repeat the above process, for the same θ_i but different type for γ. If we have exhausted all possible types for γ unsuccessfully, we select at random a new set of θ_i and repeat the process.

For an integer $k \geq 2$ let $\hat{\mathcal{E}}^k$ denote the set of all permutations of $\mathcal{S}_{|G|}$ wich are compositions of k elements in $\hat{\mathcal{E}}$. Extensive computation in groups of small order provides estimates of the probability that an element of $\hat{\mathcal{E}}^k$ also belongs to $\mathcal{T}\hat{\mathcal{N}}\mathcal{T}$. For example the results in [3] show that these probabilities are often significantly greater than 0. However, we have not yet made general estimates for classes of large groups which are likely candidates in a possible implementation of MST_1. Consequently, at the present time, we do not know whether the trap doors indicated in the proposed scheme can be constructed efficiently.

5 Second potential public key system $\mathbf{MST_2}$

We now turn our attention to certain covers for arbitrary groups which are not perfectly uniform as λ-logarithmic signatures are, but which have a high degree of uniformity, and are obtainable by probabilistic methods.

Definition 5.1 *Let r and s be positive integers, and G an arbitrary finite group. A cover $[A_1, \ldots, A_s]$ of G is called an $[s, r]$ – mesh if:*

i) $A_i \in G^{[\mathbb{Z}]}$ *and* $|A_i| = r$, *for each* $i \in \{1, \ldots, s\}$

ii) In

$$\overline{A_1} \cdot \overline{A_2} \cdots \overline{A_s} = \sum_{g \in G} a_g \, g$$

the distribution of the $\{a_g : g \in G\}$ *is approximately uniform.*

In the spirit of (2.3) we represent an $[s, r]$-mesh by an $s \times r$ matrix $\alpha = (a_{i,j})$, where each $a_{i,j} \in G$.

We measure the *degree of uniformity* of a mesh by applying the standard statistical uniformity measures to the distribution $\{a_g : g \in G\}$, or to the probability distribution $\{P_g : g \in G\}$, where $P_g = a_g / r^s$ (for example an appropriate Kolmogorov-Smirnov statistic.)

Experimentation with matrices $(a_{i,j})$ constructed by sampling random elements $a_{i,j}$ in arbitrary groups shows that $[s, r]$-meshes proliferate. In practice, we may select $a_{i,j}$ from a particular conjugacy class of G.

Suppose now that $\alpha = (a_{i,j})$ is a random $[s, r]$-mesh covering a finite group G. Our cryptographic hypothes is that, if $g \in G$, then finding a factorization

$$g = a_{1,j_1} \cdot a_{2,j_2} \cdots a_{s,j_s} \tag{5.7}$$

is, in general, an intractable problem. Let H be a second group and $f : G \to H$ an epimorphism. Then $\beta = (b_{i,j})$, where $b_{i,j} = f(a_{i,j})$, is an $[s, r]$-mesh for H. In general, the surjections $\breve{\alpha} : \mathbb{Z}_{r^s} \to G$ and $\breve{\beta} : \mathbb{Z}_{r^s} \to H$ are not bijections, but are efficiently computable.

We are now ready to describe our second public key cryptosystem in the context of $[s, r]$-meshes for groups.

5.1 The Second system $\mathbf{MST_2}$

Alice chooses large groups G and H, an epimorphism $f : G \to H$, and generates a random $[s, r]$-mesh $\alpha = (a_{i,j})$ for G. Alice computes $\beta = f(\alpha) = (b_{i,j}) = (f(a_{i,j}))$.

She makes the pair (α, β) public, but keeps f secret. If Bob wants to send a message $x \in H$ to Alice, he:

i) Chooses a random integer $R \in \mathbb{Z}_{r^s}$

ii) Computes $y_1 = \breve{\alpha}(R), \quad y_2 = \breve{\beta}(R), \quad y_3 = xy_2$, and

iii) Sends $y = (y_1, y_3)$ to Alice.

Upon receiving (y_1, y_3), Alice computes $y_2 = \breve{\beta}(R) = f(\breve{\alpha}(R)) = f(y_1)$, and from $xy_2 = y_3$ obtains the message $x = y_3 y_2^{-1}$.

5.2 Security of MST$_2$

There are two types of attacks we can envision against MST$_2$. The first would be to determine a random number R from an intercepted $y = (y_1, y_3)$, so that $y_1 = \breve{\alpha}(R)$. Note that in general R is not unique, but finding any R' such that $y_1 = \breve{\alpha}(R) = \breve{\alpha}(R')$ constitutes breaking the system. Finding an R such that $y_1 = \breve{\alpha}(R)$, amounts to being able to factorize y_1 with respect to α, and determine pointers $(j_1, j_2, \ldots, j_s)$ for which

$$y_1 = a_{1,j_1} a_{2,j_2} \cdots a_{s,j_s}$$

As discussed earlier this attack is conjectured to be infeasible if G is large.

A second possible attack is to infer any homomorphism $f' \in \mathrm{HOM}(G, H)$ such that $\beta = f'(\alpha)$. Finding such an f' would constitute breaking Alice's key.

The security of MST$_2$ is based on the fact that if G is an arbitrary finite group and if $\{g_i\}$ is a collection of elements of G, then in general, computing the intersection of centralizers in G of the g_i is hard.

For example, we note that the ElGamal system [1] is a special case of MST$_2$, where the ambient space is a large cyclic group (it is in fact the ElGamal system that inspired our generalization). Whether we can find a practical and efficient implementation of MST$_2$ for large non-abelian groups is still an open question. We make a few observations on this possibility.

Consider the instance of the system where $G = H$ is a large, non-abelian group, and where $f : G \to G$ is conjugation by an element $g \in G$. Thus, Alice has chosen an $[s, r]$-mesh $(a_{i,j})$ for G, and a secret element $g \in G$, and computes the second mesh $\beta = (b_{i,j})$ by $b_{i,j} = a_{i,j}^g$. She publishes the two meshes $\alpha = (a_{i,j})$ and $\beta = (b_{i,j})$, but keeps g secret. Here, finding any element $g' \in G$ such that $b_{i,j} = a_{i,j}^{g'}$ would constitute breaking Alice's key.

Assume that finding respective elements $u_{i,j} \in G$ such that $a_{i,j}^{u_{i,j}} = b_{i,j}$ is easy. Now, from elementary group theory we know that if $x, y, z \in G$ such that $z = x^y$, then $\{w \in G | x^w = z\} = C_G(x)y$, where $C_G(x)$ is the centralizer of x in G. So, the set of all elements which conjugate x onto z is a right coset of the centralizer of x in G. Thus to make the second attack work, the cryptanalyst needs to compute an element in

$$\Theta = \bigcap_{i,j} C_G(a_{i,j}) u_{i,j}. \tag{5.8}$$

This problem is polynomially equivalent to finding the intersection

$$\Theta = \bigcap_{i,j} C_G(a_{i,j}) \tag{5.9}$$

which in general is known to be hard [4], [5]. However, for certain obvious choices of the group G, the problem can be solved in polynomial time. For example, in the special case where the group G is the symmetric group S_n in its natural representation on n points, the resulting system MST_2 is not secure. A similar situation arises if the any of the centralizer subgroups is solvable. So it remains an open problem to construct secure variants of MST_2 in nonabelian groups.

Acknowledgements

The first author wishes to express his thanks to the Institute for Experimental Mathematics, University of Essen, Germany, and to the Centre for Applied Cryptographic Research, University of Waterloo, Canada, whose hospitality he enjoyed while carrying out parts of this work. Thanks also to L. Babai and E. Luks for helpful comments.

References

[1] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory,* **31** (1985), pp. 469 – 472.

[2] M. Furst, J. E. Hopcroft, and E. Luks, Polynomial-time algorithms for permutation groups, In *Proceedings of the 21'st IEEE Symposium on Foundations of Computation of Computer Science*, (1980), pp. 36–41.

[3] Hanchang Fan, Group Factorizations and Cryptography, M.Sc. Thesis, *Department of Computer Science and Engineering, University of Nebraska - Lincoln*, (1999), pp. 1–58.

[4] E. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, J. Comput. System Sci. **25** (1982), 42–65.

[5] E. Luks, Computing the composition factors of a permutation group in polynomial time. Combinatorica **7** (1987), 87–99.

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

[7] S. S. Magliveras and N. D. Memon, The Algebraic Properties of Cryptosystem PGM, *J. of Cryptology*, **5** (1992), pp. 167-183.

[8] M. Qu and S. A. Vanstone, Factorizations of elementary abelian p-groups and their cryptographic significance, *J. of Cryptology*, **7** (1994), pp. 201–212.

[9] C. C. Sims, Some group-theoretic algorithms, *Topics in Algebra*, M.F. Newman, Editor, Springer-Verlag, Lecture Notes in Math., **697** (1978), pp. 108–124.