Towards a Uniform Description of Several Group Based Cryptographic Primitives

María Isabel González Vasco[1], Consuelo Martínez[1], Rainer Steinwandt[2]

[1]Departamento de Matemáticas, Universidad de Oviedo,
c/Calvo Sotelo, s/n, 33007 Oviedo, Spain
mvasco@orion.ciencias.uniovi.es, chelo@pinon.ccu.uniovi.es

[2]Institut für Algorithmen und Kognitive Systeme,
Arbeitsgruppe Systemsicherheit, Prof. Dr. Th. Beth,
Universität Karlsruhe, 76128 Karlsruhe, Germany
steinwan@ira.uka.de

Abstract

The public key cryptosystems MST_1 and MST_2 make use of certain kinds of factorizations of finite groups. We show that generalizing such factorizations to infinite groups allows a uniform description of several proposed cryptographic primitives. In particular, a generalization of MST_2 can be regarded as a unifying framework for several suggested cryptosystems including the ElGamal public key system, a public key system based on braid groups and the MOR cryptosystem.

1 Introduction

The security of many cryptographic tools relies on assumptions about the hardness of certain algorithmic problems. Being able to group several such problems and formalize them as particular instances of a general problem helps us in identifying essential algorithmic requirements for cryptographic applications. A good example is the identification of the *integer factorization* and the *discrete logarithm problem* as instances of the *abelian stabilizer problem* which is solvable through a polynomial quantum algorithm (cf. [3]). The *abelian stabilizer problem* can in turn be taken for an instance of an *abelian hidden subgroup problem* (e. g. [6]). Another example for studying common properties of several cryptographic primitives is provided by [13],

[1]Work partially supported by the project BFM2001-3239-C03-01

where a construction for a private information retrieval (PIR) system based on any *subgroup membership problem* is described.

It is worth remarking that in all these examples the identified general problem originates in group theory. Two public key cryptosystems based on the difficulty of computing certain factorizations in finite groups, have been introduced in [5]: MST_1 and MST_2. Unfortunately it is still unclear how to create practical instances of these conceptually rather appealing systems. Subsequently we demonstrate that, after a suitable generalization, the factorization concepts used in MST_1 and MST_2 allow a uniform description of several cryptographic primitives. Also, it turns out that a generalization of MST_2 can serve as a unifying framework for several proposed public key cryptosystems, e. g., the ElGamal and the braid group based system from [4].

The organization of our contribution is as follows: after defining suitable generalizations of some of the factorization concepts considered in [5], we give a uniform description of several known cryptographic primitives in this framework. Thereafter we introduce a generalization of MST_2 and show how various known cryptosystems can be regarded as instances hereof. Some comments on possible further research directions conclude the paper.

2 Factoring a set with respect to an action

Several kinds of finite group factorizations are introduced and discussed in [5]. We want to generalize some of these concepts: instead of considering only the usual product action in finite groups, we consider the action of a (possibly infinite) semigroup with unit on a set. This more general setting proves to be convenient for describing several well-known cryptographic primitives in a uniform manner.

Definition 2.1 *Let $M = (M, \cdot, 1)$ be a semigroup with unit acting on a set T by an action*

$$\circ : \begin{array}{ccc} M \times T & \longrightarrow & T \\ (m,t) & \longmapsto & m \circ t \end{array},$$

and Λ^* a countable set that is totally ordered. Set $\Lambda := \Lambda^* \cup \{\top\}$ with $\top \notin \Lambda^*$ and consider the extended order in Λ such that $\top$ is greater than each element in Λ^*.
Let $A = [A_\lambda]_{\lambda \in \Lambda}$ be a sequence such that:

- *for each* $\lambda \in \Lambda^*$, $A_\lambda := [\alpha_{\lambda i}]_{i \in I_\lambda}$ *is a countable sequence over* M *such that the unit appears in all but finitely many of the* blocks A_λ;
- *the block* $A_\top := [\alpha_{\top i}]_{i \in I_\top}$ *is a countable sequence over* T.

We call A *an* $(M, \circ, \Lambda)$-cover *for* T *(or simply, a* cover *for* T*) if each* $t \in T$ *can be expressed as* $t = m \circ \alpha_\top$, *for some* $\alpha_\top \in A_\top$ *and some element* $m \in M$ *which can be factored as a product* $\prod_{\lambda \in \Lambda^*} \alpha_\lambda$, *where* $\alpha_\lambda \in A_\lambda$ *and only finitely many factors are different from the unit.*

We will refer to the former factorization by writing

$$t = \bigcirc_{\lambda \in \Lambda} \alpha_\lambda,$$

where $\bigcirc_{\lambda \in \Lambda} \alpha_\lambda$ denotes iterative application of $\circ$.
In the sequel, when $\Lambda^* = \{1, \dots, n\} \subseteq \mathbb{N}$ with the natural order, we will simply choose $\top := n+1$, so that a cover A has the form $A = [A_1, \dots, A_{n+1}]$.

Some kinds of covers are of special interest for us, and we name them in such a way that for the special case of M being a finite group acting on itself through left-multiplication the definitions specialize essentially to those given in [5] (cf. Remark 2.3).

Definition 2.2 *Let* $A := [[\alpha_{\lambda i}]_{i \in I_\lambda}]_{\lambda \in \Lambda}$ *be an* $(M, \circ, \Lambda)$*-cover for a set* T, *and for* $t \in T$ *consider the set*

$$F(t) := \left\{ [i_\lambda]_{\lambda \in \Lambda} \in \prod_{\lambda \in \Lambda} I_\lambda : t = \bigcirc_{\lambda \in \Lambda} \alpha_{\lambda i_\lambda} \right\},$$

which characterizes the possible factorizations of t *with respect to* A. *Then we call* A *a*

- logarithmic signature *if* $\mathrm{card}(F(t)) = 1$ *for all* $t \in T$.
- mesh *if either*
  - *all* $F(t)$ *are infinite, or*
  - *all* $F(t)$ *are finite, and if* T *is infinite, of equal cardinality, whereas if* T *is finite, the probability distribution*

    $$[P_t = |F(t)|/|T| : t \in T]$$

    is approximately uniform (in the sense of [5]).

- $[\operatorname{card}(\Lambda), r]$-mesh *if A is a mesh and all index sets I_λ $(\lambda \in \Lambda)$ are of equal cardinality r.*

In particular, the elements within each block of a logarithmic signature A must be pairwise different. Note also that the notions of cover, logarithmic signature, and $[\operatorname{card}(\Lambda), r]$-mesh used in the construction of MST_1 and MST_2 can indeed be taken for a special case of the above concepts:

Remark 2.3 *Let $M = T := G$ be a finite group, and denote by $\circ$ the action of G on itself through left-multiplication. Choosing $\Lambda := \{1, \ldots, s\}$ with $s \in \mathbb{N}$ and imposing I_λ to be finite $(\lambda \in \Lambda)$, we obtain the original notions of cover, logarithmic signature, and $[\operatorname{card}(\Lambda), r]$-mesh from [5].*

In addition, other cryptographic primitives can be related to the problem of factoring elements with respect to some cover:

Example 2.4 [Prime factorization of natural numbers]
Let $M = T := \mathbb{N}$ be the multiplicative semigroup of natural numbers acting on itself through left-multiplication.
Further on, set $\Lambda^* := \mathbb{N}^2$ with the following (lexicographical) order $\preceq$: for $(\mu_1, \nu_1), (\mu_2, \nu_2) \in \mathbb{N}^2$ we have $(\mu_1, \nu_1) \preceq (\mu_2, \nu_2)$ iff $\mu_1 < \mu_2$ or $\mu_1 = \mu_2$ and $\nu_1 \leq \nu_2$.
Finally, denoting by $p_\mu \in \mathbb{N}$ the μ^{th} prime number $(\mu \in \mathbb{N})$, for $(\mu, \nu) \in \Lambda^*$ we set $A_{(\mu,\nu)} := [1, p_\mu^{2^{\nu-1}}]$ and define $A_\top := [1]$.

Factoring elements $t \in T$ with respect to the cover $A := [A_\lambda]_{\lambda \in \Lambda}$ translates into decomposing a natural number into its prime factors; in particular the cover just defined is a logarithmic signature.

Example 2.5 [Discrete logarithm in $\mathbb{F}_{p^n}^*$]
Let p be a prime number, $M = T := \mathbb{F}_{p^n}^*$ the cyclic group of order $p^n - 1$, and α some generator of M. Then M acts on T through left-multiplication, and setting $A_\lambda := [1, \alpha^{p^{\lambda-1}}, \alpha^{2p^{\lambda-1}}, \ldots, \alpha^{(p-1)p^{\lambda-1}}]$ $(1 \leq \lambda \leq n)$ yields an $[n, p]$-mesh $A := [A_1, \ldots, A_n]$ for T.

Obviously, factoring with respect to A translates into computing discrete logarithms with respect to α.

Example 2.6 [Yamamura's cryptosystem from [12]]
As explained in [12], $SL_2(\mathbb{Z})$ acts on the upper half plane

$$\mathcal{H} := \{z \in \mathbb{C} : \Im(z) > 0\}$$

through application of the fractional (Möbius) transformation determined by the corresponding matrix. Namely, for $S := (s_{ij})_{1\leq i,j\leq 2} \in SL_2(\mathbb{Z})$ and $z \in \mathcal{H}$ we have

$$S \circ z := \frac{s_{11}z + s_{12}}{s_{21}z + s_{22}} \, .$$

Analogously, for $B \in GL_2(\mathbb{R})$ arbitrary, the group $G := B^{-1}SL_2(\mathbb{Z})B$ acts on $B^{-1}\mathcal{H} := \{B^{-1} \circ z : z \in \mathcal{H}\}$. We denote the latter action also by $\circ$.

In the cryptosystem from [12], a suitably chosen point $p \in B^{-1}\mathcal{H}$ and two matrices $W_0, W_1 \in B^{-1}SL_2(\mathbb{Z})B$ satisfying certain restrictions (see [12] for details) are made public. The ciphertext $c \in B^{-1}\mathcal{H}$ corresponding to a (plaintext) bitstring $b = b_1 \dots b_l \in \{0,1\}^l$ is

$$c := \Bigl(\prod_{i=1}^{l} W_{b_i}\Bigr) \circ p \quad \bigl(= W_{b_1} \circ (W_{b_2} \circ (\dots (W_{b_l} \circ p) \dots))\bigr) \, .$$

In our terminology the problem of recovering a plaintext from a ciphertext can be expressed as follows: let M be the subsemigroup (with unit) of $SL_2(\mathbb{R})$ generated by the matrices W_0, W_1, and denote by T the M-orbit of the point p under the action $\circ$. Then decrypting a ciphertext $c \in T$ is equivalent to finding a factorization of c with respect to the mesh[1] $A = [A_\lambda]_{\lambda\in\mathbb{N}\cup\{\top\}}$ where $A_\lambda := [I_2, W_0, W_1]$ for $\lambda \in \mathbb{N}$ and $A_\top := [p]$.

When using a cover A in a cryptographic context, the following question arises: how difficult is it to actually compute factorizations with respect to A? E. g., factoring with respect to the cover defined in the last example is equivalent to decrypting ciphertexts in Yamamura's cryptosystem [12], and one could be tempted to believe that this is a difficult task. However, the successful attacks in [1, 9] against Yamamura's system show that this task is not difficult enough to offer acceptable cryptographic security.

To describe more precisely whether factoring with respect to some cover is difficult, one might think of adopting some terminology used in [5] for logarithmic signatures, namely to introduce a notion of *wild* and *tame* covers:

Definition 2.7 *We call an $(M, \circ, \Lambda)$-cover $A = [A_\lambda]_{\lambda\in\Lambda}$ for a set T* tame *if there exists a polynomial time algorithm[2] which on input $t \in T$ computes*

[1] Note that each $t \in T$ admits $\aleph_0$ 'different' factorizations with respect to A.

[2] Polynomial in the size of the input; this parameter should be specified depending on the used representation of the elements in M and T.

elements $\alpha_\lambda \in A_\lambda$ $(\lambda \in \Lambda)$ *so that* $t = \bigcirc_{\lambda\in\Lambda} \alpha_\lambda$. *A cover which is not tame is called* wild.

However, being wild in such a sense does not necessarily imply a oneway-property which is often desired in cryptographic contexts: even if no efficient algorithm for factoring arbitrary elements from T with respect to A is known, it might still be feasible to factor efficiently a non-negligible fraction of the elements in T with respect to A. For 'wild-like' logarithmic signatures for finite groups the problem of such a *partial inversion* is addressed in [11]. Both for the original MST_2 and for the generalization of MST_2 introduced below, it is crucial that such partial inversion attacks are infeasible.

3 Generalizing MST_2

To describe the generalization of MST_2 we do not need Definition 2.1 in full generality and thus we focus on a more specialized situation: throughout this section M is required to be a finitely presented group, instead of arbitrary covers we restrict our attention to meshes, and we require the index set Λ to be finite. It turns out that the definition of the cryptosystem MST_2 from [5] can then easily be adapted to our notion of meshes of groups and that several known group based cryptosystems can be regarded as instances of such a generalized MST_2 scheme: the ElGamal public key system [2], the braid group based public key system from [4], and the MOR cryptosystem from [7, 8].

Let $M = G = \langle X; R\rangle$ be a finitely presented group acting on two sets T, T'. We denote the action of G on the G-sets T and T' by $\circ$ and $\bullet$, respectively. Further on, let $A = [A_\lambda]_{1\leq\lambda\leq s}$ be a mesh for T with $s \in \mathbb{N}$ and $A_\lambda = [\alpha_{\lambda i}]_{i\in I_\lambda}$ for all $\lambda \in \Lambda$. Then we can define a map

$$\begin{array}{rccc} \breve{\alpha}: & I_1 \times \cdots \times I_s & \longrightarrow & T \\ & (r_1, \ldots, r_s) & \longmapsto & \bigcirc_{\lambda=1}^{s} \alpha_{\lambda r_\lambda} \end{array}.$$

Now let $f : T \longrightarrow T'$ be a G-map (i. e. $f(g \circ t) = g \bullet f(t)$ for all $t \in T$, $g \in G$) such that $B := [A_1, \ldots, A_{s-1}, f(A_s)]$ is a mesh of T'. Analogously as above, we define a corresponding mapping $\breve{\beta} : I_1 \times \cdots \times I_s \longrightarrow T'$.

We also need some set $\mathcal{T}$ (that will serve as plaintext space) along with a (public) map $\tau : T' \longrightarrow S_{\mathcal{T}}$ that associates to each $t' \in T'$ a permutation on $\mathcal{T}$. With this notation we can formulate the following framework for a public key cryptosystem:

- **Public key:** $\breve{\alpha}$ and $\breve{\beta}$
- **Secret key:** the G-map f
- **Encryption:** to send a message $m \in \mathcal{T}$ to Alice, Bob
  1. chooses $(r_1, \ldots, r_s) \in I_1 \times \cdots \times I_s$ at random,
  2. computes $y_1 := \breve{\alpha}(r_1, \ldots, r_s) \in T$, $y_2 := \breve{\beta}(r_1, \ldots, r_s) \in T'$ and
  3. sends the pair $(y_1, \tau(y_2)(m)) \in T \times \mathcal{T}$ to Alice.
- **Decryption:** From $y_2 = f(y_1)$, Alice derives $m = \tau(y_2)^{-1}(\tau(y_2)(m))$.

The security of the system relies on the hardness of computing y_2 from the public information. As $\breve{\beta}$ is public, the sequence $r_1, \ldots, r_s$ should be hard to retrieve from y_1; in particular α should be 'wild'. Of course, $\tau(y_2)(m)$ must not leak significant information about m, either.

The only concrete example of the original MST_2, i. e., without our generalizations, we are aware of is the ElGamal cryptosystem. Before demonstrating that our generalization allows to give further examples, let us verify that the above scheme is really a generalization of the original MST_2 described in [5]:

Remark 3.1 *Let $f : G \longrightarrow H$ be a group epimorphism. Then G acts on $T := G$ through left-multiplication ($\circ$), and on $T' := H$ via $g \bullet h := f(g) \cdot h$ (the ordinary product of $f(g)$ with h in H). In particular, f is a G-map.*

Now each $[s, r]$-mesh for T in the sense of [5] is also an $[s, r]$-mesh $A = [A_1, \ldots, A_s]$ for T in the sense of Definition 2.2. Finally, we set $\mathcal{T} := H$ and define the map $\tau : H \longrightarrow S_H$ via $\tau(h)(x) := x \cdot h$.

To publish the maps $\breve{\alpha}$, $\breve{\beta}$, Alice makes both sequences $A = [A_1, \ldots, A_s]$ and $f(A) = [f(A_1), \ldots, f(A_s)]$ public. The scheme obtained in this way is nothing but a more complicated description of the original MST_2 defined in [5].

4 Known systems as instances of MST_2

In this section we show that several group based public key cryptosystems that have been suggested can be seen as instances of the above framework.

As we have just seen, the original MST_2 is a special case of our setting, and hence it comes to no surprise that we can express the ElGamal cryptosystem [2] in our framework, too (cf. [5]):

Example 4.1 [ElGamal cryptosystem]
Let p be a prime number, α some generator of the cyclic group $G := \mathbb{F}_{p^n}^*$ of $p^n - 1$ elements, and $T = T' = \mathcal{T} := G$. Next, by choosing an integer $1 < d < p^n - 1$ with $\gcd(d, p^n - 1) = 1$ we define an isomorphism $f : G \longrightarrow G$ via $f(g) := g^d$.

Consider the actions of G on T through left-multiplication and on T' through $g \bullet t' := f(g) \circ t' = f(g) \cdot t'$. Further on, fix A as in Example 2.5. Then f is a G-map, and it is easy to see that A and $B := [A_1, \ldots, A_{n-1}, f(A_n)]$ are $[n, p]$-meshes of T, T', respectively. Finally, we set

$$\tau(g)(x) := x \cdot g$$

and publish the mappings $\breve{\alpha}$, $\breve{\beta}$ by making A and $f(A) = [f(A_1), \ldots, f(A_n)]$ public. So we obtain the following scheme:

- **Public key:** *A (that is essentially given by α) and $f(A)$ (which is essentially specified by α^d)*
- **Secret key:** *f (that is determined by d)*
- **Encryption:** *to send a message $m \in G$ to Alice, Bob*
  1. *chooses $(r_1, \ldots, r_n) \in \{0, \ldots, p-1\}^n$ at random,*
  2. *computes $y_1 := \breve{\alpha}(r_1, \ldots, r_n)$, i. e., y_1 is a random power of α, $y_2 := \breve{\beta}(r_1, \ldots, r_n) = y_1^d$, and*
  3. *sends $(y_1, m \cdot y_1^d)$ to Alice.*
- **Decryption:** *Alice computes $y_2 = y_1^d$ and obtains $m = (m \cdot y_1^d) \cdot y_2^{-1}$.*

Thus we get a (slightly complicated) description of the ElGamal cryptosystem.

While the ElGamal cryptosystem also fits into the original MST_2 framework, the next example exploits our more general setting:

Example 4.2 [Braid group based cryptosystem from [4]]
For braid groups we adopt the terminology from [4]. Let B_n be the braid group on $n = l + r$ strands for a suitably chosen $n \in \mathbb{N}$, and denote by LB_l (resp. RB_r) the subgroup of B_n where only the left l (resp. right r) strands are braided.

Fixing suitable (see [4]) $x \in B_n$ and $a \in LB_l$, the group $G := RB_r$ acts on the sets $T := GxG^{-1}$ and $T' := aTa^{-1}$ by conjugation.
Moreover, the map $f : T \longrightarrow T'$, $t \mapsto ata^{-1}$ is a G-map, and one easily checks that $A := [A_1, A_2]$, where $A_1 := [b]_{b\in RB_r}$ and $A_2 := [x]$, is a mesh for T: given any $t = b_0xb_0^{-1} \in T$, for $b \in G$ we have $b \circ x = t$ if and only if $b^{-1}b_0 \in C_G(x)$ (i. e. $b^{-1}b_0x = xb^{-1}b_0$), and therefore each element of T admits exactly $\mathrm{card}(C_G(x))$ factorizations with respect to A. Analogously, we recognize $B := [[b]_{b\in RB_r}, [f(x)]]$ as mesh for T'.
Finally, the plaintext space $\mathcal{T} := \{0, 1\}^k$ consists of the bitstrings of some fixed length k, and by means of an ideal hash funtion $h : B_n \longrightarrow \mathcal{T}$ we define

$$\begin{array}{rccc} \tau : & T' & \longrightarrow & S_{\mathcal{T}} \\ & b & \longmapsto & (m \mapsto h(b) \oplus m) \end{array},$$

where $\oplus$ denotes the usual 'XOR' operation of bitstrings.
The resulting system reads as follows:

- **Public key:** *$\breve{\alpha}$ (given by x) and $\breve{\beta}$ (given by $f(x) = axa^{-1}$)*
- **Secret key:** *f (that is determined by a)*
- **Encryption:** *to send a message $m \in \{0, 1\}^k$ to Alice, Bob*
  1. *chooses some $b \in RB_r$ at random,*
  2. *computes $y_1 := \breve{\alpha}(b, 1) = bxb^{-1}$ (note that $\mathrm{card}(A_2) = 1$), $y_2 := \breve{\beta}(b, 1) = baxa^{-1}b^{-1}$, and*
  3. *sends $(y_1, h(y_2) \oplus m)$ to Alice.*
- **Decryption:** *Alice computes $y_2 = ay_1a^{-1}$ and $m = h(y_2)\oplus(h(y_2)\oplus m)$.*

In other words, we get a description of the braid group based system from [4].

Finally, we show that the MOR schemes from [7, 8] also fit into our setting:

Example 4.3 [MOR cryptosystem]
Let K, K' be finite groups and $\varphi : K \longrightarrow \mathrm{Aut}(K')$ a suitable homomorphism (see [8]). For some element $g \in K$, define G as the subgroup of $\mathrm{Aut}(K')$ generated by $\Phi := \varphi(g)$. Moreover, let $\Gamma = [\gamma_i]_{i\in I}$ be a sequence of generators of K'. Define $T := G(\Gamma)$, namely,

$$T := \{\Psi(\Gamma) = [\Psi(\gamma_i)]_{i\in I} : \Psi \in G\}.$$

Choosing a (secret) integer a, the set T' is defined as $G^a(\Gamma)$, that is,

$$T' := \{\Psi^a(\Gamma) : \Psi \in G\}.$$

The group G acts on the set T through $\Phi^k \circ \Upsilon := \Phi^k(\Upsilon)$ and on T' through $\Phi^k \bullet (\Phi^a)^i(\Gamma) := (\Phi^a)^{i+k}(\Gamma)$. Moreover, $f : T \longrightarrow T', \Phi^i(\Gamma) \longmapsto (\Phi^i)^a(\Gamma)$ is a G-map.

The plaintext space $\mathcal{T}$ is the group K', and the map $\tau : T' \longrightarrow S_{\mathcal{T}}$ is defined through

$$\tau(\Phi^{ak}(\Gamma))(x) := \Phi^{ak}(x).$$

Suppose the order of Φ is $m = p_1 \cdots p_{s-1}$, and for $1 \le \lambda \le s-1$ define

$$m_\lambda := \prod_{i=1}^{\lambda-1} p_i \quad \text{('mixed radix representation').}$$

Now consider the sequence $A := [A_1, \ldots, A_s]$ where $A_\lambda := [\Phi^{im_\lambda}]_{0\le i\le p_\lambda-1}$ for $1 \le \lambda \le s-1$ and $A_s := [\Gamma]$. Owing to the uniqueness of the mixed radix representation, A is easily verified to be a mesh for T. Similarly, we recognize $B := [A_1, \ldots, A_{s-1}, f(A_s)]$ as mesh for T'. The corresponding map $\breve{\beta}$ can be published by making $f(A) = [f(A_1), \ldots, f(A_s)]$ public. In summary, the resulting cryptosystem may be described as follows:

- **Public key:** *A and $f(A)$ (given by Φ and Φ^a)*
- **Secret key:** *f (that is determined by a)*
- **Encryption:** *We assume that an efficient algorithm for expressing a given $x \in K'$ as a word in the generators Γ is available (see [7, 8]). Then computing images under $\tau(\Phi^{ak}(\Gamma))$ is straightforward. To send a message m to Alice, Bob*
  1. *chooses at random integers r_λ with $0 \le r_\lambda \le p_\lambda - 1$ ($1 \le \lambda \le s-1$)*

2. computes

$$y_1 := \breve{\alpha}(r_1, \ldots, r_{s-1}, 1) = \Phi^b(\Gamma),$$
$$y_2 := \breve{\beta}(r_1, \ldots, r_{s-1}, 1) = \Phi^{ab}(\Gamma)$$

where $b = \sum_{i=1}^{s-1} r_i m_i$*, and*

3. sends $(y_1, \Phi^{ab}(m))$ *to Alice.*

- **Decryption:** *From* $f(y_1) = \Phi^{ab}(\Gamma)$ *Alice derives* $m = \Phi^{-ab}(\Phi^{ab}(m))$.

In other words, we obtain a description of the generalized MOR scheme from [8] (and thus also for the basic MOR system introduced in [7]).

5 Concluding remarks

The generalizations of the concepts of cover, mesh, and logarithmic signature introduced in this paper are, in our opinion, of interest in their own. Not only may it be interesting to search for analogues of the existing results and algorithms for the 'old' group factorizations valid for the more general definition, but also to find new ones. Indeed, a crucial question remains unanswered (for both the 'classical' and our generalized definition): how to identify 'truly wild' covers, i.e. covers for which computing factorizations is *almost always* very hard.

As we have seen in the above examples, several proposed cryptographic primitives and schemes rely on the difficulty of computing factorizations with respect to a certain cover. It might be worthwhile to explore the question of which cryptographic algorithms can be constructed having at hand a 'truly wild' cover, or to identify some guidelines on how such algorithms could look like (as it is done in [13], where a general construction for a PIR system based on any subgroup membership problem is given).

Various ideas motivate the introduction of the generalized MST_2 scheme: in the spirit just mentioned, it provides an abstract construction of a public key cryptosystem feasible whenever we have certain mathematical tools available. But it also yields a uniform description of several well-known cryptosystems, which may be useful for studying their common properties. In addition, this framework may also encourage the proposal of new schemes inspired by the existing examples.

References

[1] S. R. Blackburn. Cryptanalysis of two cryptosystems based on group actions. In K. Y. Lam, editor, *Advances in cryptology — ASIACRYPT '99*, volume 1716 of *Lecture Notes in Computer Science*, pages 52–61. Springer, 1999.

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

[3] A. Kitaev. Quantum measurements and the Abelian Stabilizer Problem. Electronic Colloquium on Computational Complexity Report TR96-003, 1996. At the time of writing available electronically at `ftp://ftp.eccc.uni-trier.de/pub/eccc/reports/1996/TR96-003/Paper.ps`.

[4] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. s. Kang, and C. Park. New Public-Key Cryptosystem Using Braid Groups. In M. Bellare, editor, *Advances in cryptology — CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 166–183. Springer, 2000.

[5] S.S. Magliveras, D.R. Stinson, and T. van Trung. New Approaches to Designing Public Key Cryptosystems Using One-Way Functions and Trapdoors in Finite Groups. To appear in *Journal of Cryptology*. See also the Technical Report CORR 2000-49, Centre for Applied Cryptographic Research, University of Waterloo (at the time of writing available electronically at the URL `http://www.cacr.math.uwaterloo.ca/techreports/2000/corr2000-49.ps`).

[6] M. Mosca and E. Ekert. The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum Computer. In C.P. Williams, editor, *Quantum Computing and Quantum Communications, First NASA International Conference, QCQC'98*, volume 1509 of *Lecture Notes in Computer Science*, pages 174–188. Springer, 1999.

[7] S.-H. Paeng, K.-C. Ha, J. H. Kim, S. Chee, and C. Park. New Public Key Cryptosystem Using Finite Non Abelian Groups. In J. Kilian, editor, *Advances in cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 470–485. Springer, 2001.

[8] S.-H. Paeng, D. Kwon, K.-C. Ha, and J. H. Kim. Improved public key cryptosystem using finite non abelian groups. Cryptology ePrint Archive: Report 2001/066, 2001. At the time of writing available electronically at `http://eprint.iacr.org/2001/066/`.

[9] R. Steinwandt. Loopholes in Two Public Key Cryptosystems Using the Modular Group. In K. Kim, editor, *4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 180–189. Springer, 2001.

[10] M. I. González Vasco, C. Martínez, and R. Steinwandt. Un Marco Común para Varios Esquemas de Clave Pública Basados en Grupos. To appear in VII Reunión Española sobre Criptología y Seguridad de la Información, VII RECSI Proceedings.

[11] M. I. González Vasco and R. Steinwandt. Obstacles in Two Public Key Cryptosystems Based on Group Factorizations. To appear in *Cryptology*, volume 25 of *Tatra Mountains Mathematical Publications*, 2002.

[12] A. Yamamura. A functional cryptosystem using a group action. In J. Pieprzyk, editor, *Information security and privacy. 4th Australasian conference, ACISP '99*, volume 1587 of *Lecture Notes in Computer Science*, pages 314–325. Springer, 1999.

[13] A. Yamamura and T. Saito. Private Information Retrieval Based on the Subgroup Membership Problem. In V. Varadharajan and Y. Mu, editors, *Information Security and Privacy, 6th Australasian Conference, ACISP 2001*, volume 2119 of *Lecture Notes in Computer Science*, pages 206–220. Springer, 2001.