Non-associative key establishment for leftdistributive systems

Arkadius Kalka, Mina Teicher

Abstract

We construct non-associative key establishment protocols for all LD-, multi-LD-, and other left-distributive systems.

2010 *Mathematics Subject Classification.* 20N02, 20F36.
Key words and phrases. Non-commutative cryptography, key establishment protocol, magma (grupoid), left-distributive system, braid group.

1 Introduction

In an effort to construct new key establishment protocols (KEPs), which are hopefully harder to break, the first author introduced in his PhD thesis [Ka07] (see also [Ka12]) first non-associative generalizations of the Anshel-Anshel-Goldfeld KEP [AAG99], which revolutionized the field of *non-commutative public key cryptography* (PKC) more than ten years ago. For an introduction into non-commutative public key cryptography we refer to the book by Myasnikov et al. [MSU11]. For further motivation and on *non-associative PKC* we refer to [Ka12]. It turns out (see [Ka12]) that in the context of AAG-like KEPs for magmas, left-selfdistributive systems (LD-systems) and their generalizations (like multi-LD-systems) naturally occur. Though we constructed several examples for non-associative KEPs for LD- and multi-LD-systems [Ka12], we did not provide a general method to construct a KEP that works for all LD- and multi-LD-systems. We are going to fill this gap in the present paper.

Outline In section 2 we review LD-, multi-LD-, and other left-distributive systems with many examples (see also section 4.1 in [Ka12]). Section 3 describes a KEP for all LD-systems with a discussion of related base problems. Finally, Protocol 2, described and analyzed in section 4, applies not only for all multi-LD-systems, but also for a big class of partial multi-LD-systems. We close with a discussion in section 5.

2 LD-systems and other distributive systems

2.1 Definition

Definition 2.1. *An LD-system $(S, *)$ is a set S equipped with a binary operation $*$ on S which satisfies the left-selfdistributivity law*

$$x * (y * z) = (x * y) * (x * z) \quad \forall x, y, z \in S.$$

Definition 2.2. *(Section X.3. in [De00]) Let I be an index set. A multi-LD-system $(S, (*_i)_{i\in I})$ is a set S equipped with a family of binary operations $(*_i)_{i\in I}$ on S such that*

$$x *_i (y *_j z) = (x *_i y) *_j (x *_i z) \quad \forall x, y, z \in S$$

is satisfied for every i, j in I. Especially, it holds for $i = j$, i.e., $(S, *_i)$ is an LD-system. If $|I| = 2$ then we call S a bi-LD-system.

We begin with some examples of LD-systems taken from [De06].

1. We begin with a trivial example. $(S, *)$ with $x * y = f(y)$ is an LD-system for any function $f : S \to S$.

2. A set S with a binary operation $*$, that satisfies no other relations than those resulting from the left self-distributivity law, is a free LD-system. Free LD-systems are studied extensively in [De00].

3. A classical example of an LD-system is $(G, *)$ where G is a group equipped with the conjugacy operation $x * y = x^{-1}yx$ (or $x *^{\text{rev}} y = xyx^{-1}$). Note that such an LD-system cannot be free, because conjugacy satisfies additionaly the idempotency law $x * x = x$.

4. Finite groups equipped with the conjugacy operation are not the only finite LD-systems. Indeed, the socalled *Laver tables* provide the classical example for finite LD-systems. There exists for each $n \in \mathbb{N}$ an unique LD-system $L_n = (\mathbb{Z}/2^n\mathbb{Z}, *)$ with $k * 1 = k + 1$. The values for $k * l$ with $l \neq 1$ can be computed by induction using the left self-distributive law. The Laver tables for $n = 1, 2, 3$ are

L_1	1	0
1	0	0
0	1	0

L_2	1	2	3	0
1	2	0	2	0
2	3	0	3	0
3	0	0	0	0
0	1	2	3	0

L_3	1	2	3	4	5	6	7	0
1	2	4	6	0	2	4	6	0
2	3	4	7	0	3	4	7	0
3	4	0	4	0	4	0	4	0
4	5	6	7	0	5	6	7	0
5	6	0	6	0	6	0	6	0
6	7	0	7	0	7	0	7	0
7	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	0

Laver tables are also described in [De00].
Many examples for LD-, bi-LD- and multi-LD-systems are given in Dehornoy's monography [De00].

2.2 f-conjugacy

One may consider several generalizations of the conjugacy operation as candidates for natural LD-operations in groups. Consider an Ansatz like $x * y = f(x^{-1})g(y)h(x)$ for some group endomorphisms f, g, h.

Proposition 2.3. *Let G be a group, and $f, g, h \in End(G)$. Then the binary operation $x * y = f(x^{-1}) \cdot g(y) \cdot h(x)$ yields an LD-structure on G if and only if*

$$fh = f, \quad gh = hg = hf, \quad fg = gf = f^2, \quad h^2 = h. \tag{1}$$

PROOF. A straightforward computation yields

$$\begin{aligned} \alpha * (\beta * \gamma) &= f(\alpha^{-1})gf(\beta^{-1})g^2(\gamma)gh(\beta)h(\alpha), \quad \text{and} \\ (\alpha * \beta) * (\alpha * \gamma) &= fh(\alpha^{-1})fg(\beta^{-1})f^2(\alpha)gf(\alpha^{-1})g^2(\gamma)gh(\alpha)hf(\alpha^{-1})hg(\beta)h^2(\alpha). \end{aligned}$$

A comparison of both terms yields the assertion. □

The simplest solution of the system of equations (1) is $f = g$ and $h = \mathrm{id}$. This leads to the following definition.

Definition 2.4. *(*LD- OR f-CONJUGACY*) Let G be a group, and $f \in End(G)$. An ordered pair $(u, v) \in G \times G$ is called f-LD-conjugated or LD-conjugated, or simply f-conjugated, denoted by $u \longrightarrow_{*_f} v$, if $\exists c \in G$ such that $v = c *_f u = f(c^{-1}u)c$.*

Remark. For any non-trivial endomorphism f, the relation $\longrightarrow_{*_f}$ defines not an equivalence relation on G. Even the relation $\longrightarrow_*$ defined by $u \longrightarrow_* v$ iff $\exists f \in Aut(G)$ s.t. $u \longrightarrow_{*_f} v$ is not an equivalence relation. Indeed, transitivity requires the automorphisms (relation must be symmetric!) to be an idempotent endomorphism ($f^2 = f$) which implies $f = \mathrm{id}$.
Compare the notion of f-LD-conjugacy with the well known notion *f-twisted conjugacy* defined by $u \sim_f v$ (for $f \in Aut(G)$) iff $\exists c \in G$ s.t. $v = f(c^{-1})uc =: c *_f^{tw} u$, which yields indeed an equivalence relation. On the other hand, the operation $*^{tw} = *_f^{tw}$ is not LD - rather it satisfies the following "near" LD-law:

$$\alpha *^{tw} (\beta *^{tw} \gamma) = (\alpha *^{tw} \beta) *^{tw} (\alpha^f *^{tw} \gamma)$$

where α^f is short for $f(\alpha)$.
Anyway, it follows directly from the definitions that $u \longrightarrow_* v$ if and only if $f(u) \sim_f v$, i.e., any f-LD conjugacy problem reduces to a twisted conjugacy problem and vice versa. Here we have to extend the notion of twisted conjugacy from $f \in Aut(G)$ to all $f \in End(G)$.

Example. Let G be the n-strand pure braid group P_n. For some small integer $d \geq 1$, consider the epimorphism $\eta_d : P_n \longrightarrow P_{n-d}$ given by 'pulling out' (or erasing) the last d strands, i.e. the strands $n - d + 1, \ldots, n$. Recall the shift map ∂, and note that $\partial^d(P_{n-d}) \leq P_n$. Now, we define the endomorphism $f : P_n \longrightarrow P_n$ by the composition $f = \partial^d \circ \eta_d$.

2.3 Shifted conjugacy

Patrick Dehornoy introduced the following generalization of f-conjugacy, and he points out, that once the definition of shifted conjugacy is used, braids inevitably appear [De00, De06].

Proposition 2.5. (Exercise I.3.20. in [De00]) *Consider a group G, a homomorphism $f : G \to G$, and a fixed element $a \in G$. Then the binary operation*

$$x * y = x *_{f,a} y = f(x)^{-1} \cdot a \cdot f(y) \cdot x$$

yields an LD-structure on G if and only if $[a, f^2(x)] = 1$ for all $x \in G$, and a satisfies the relation $af(a)a = f(a)af(a)$. Hence the subgroup $H = \langle \{ f^n(a) \mid$

$n \in \mathbb{N}\}\rangle$ *of* G *is a homomorphic image of the braid group*

$$B_\infty = \langle \{\sigma_i\}_{i\geq 1} \mid \sigma_i\sigma_j = \sigma_j\sigma_i \text{ for } |i-j| \geq 2,\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ for } |i-j| = 1\rangle$$

with infinitely many strands, i.e., up to an isomorphism, it is a quotient of B_∞.

There exists a straightforward generalization of Proposition 2.5 for multi-LD-systems:

Proposition 2.6. *Let* I *be an index set. Consider a group* G*, a family of endomorphisms* $(f_i)_{i\in I}$ *of* G*, and a set of fixed elements* $\{a_i \in G \mid i \in I\}$*. Then* $(G, (*_i)_{i\in I})$ *with*

$$x *_i y = f_i(x^{-1}) \cdot a_i \cdot f_i(y) \cdot x$$

is a multi-LD-system if and only if $f_i = f_j =: f$ *for all* $i \neq j$*,* $[a_i, f^2(x)] = 1$ *for all* $x \in G$*,* $i \in I$*, and* $a_i f(a_i) a_j = f(a_j) a_i f(a_i)$ *for all* $i, j \in I$.

Proof. A straightforward computation gives

$$\begin{aligned} x *_i (y *_j z) &= f_i(x^{-1})a_i[f_i(f_j(y^{-1}))f_i(a_j)f_i(f_j(z))f_i(y)]x, \\ (x *_i y) *_j (x *_i z) &= [f_j(x^{-1})f_j(f_i(y^{-1}))f_j(a_i^{-1})f_j(f_i(x))]a_j[f_j(f_i(x^{-1}))\cdot \\ &\quad f_j(a_i)f_j(f_i(z))f_j(x)][f_i(x^{-1})a_i f_i(y)x]. \end{aligned}$$

A comparison of both terms yields the assertion. □

Note that this proof also contains proofs of Proposition 2.5 (setting $|I| = 1$) and of the following Corollary 2.7 (setting $G = B_\infty$, $I = \{1, 2\}$, $s = \partial$, $*_1 = *$, $*_2 = \bar{*}$, $a_1 = \sigma_1$ and $a_2 = \sigma_1^{-1}$).

Consider the injective *shift endomorphism* $\partial : B_\infty \longrightarrow B_\infty$ defined by $\sigma_i \mapsto \sigma_{i+1}$ forall $i \geq 1$.

Corollary 2.7. (Shifted conjugacy, Example X.3.5. in [De00]) B_∞ *equipped with the* shifted conjugacy *operations* $*$, $\bar{*}$ *defined by*

$$x * y = \partial x^{-1} \cdot \sigma_1 \cdot \partial y \cdot x, \qquad x \,\bar{*}\, y = \partial x^{-1} \cdot \sigma_1^{-1} \cdot \partial y \cdot x$$

is a bi-LD-system. In particular, $(B_\infty, *)$ *is an LD-system.*

2.4 Generalized shifted conjugacy in braid groups

In the following we consider generalizations of the shifted conjugacy operations $*$ in B_∞. Therefore we set $s = \partial^p$ for some $p \in \mathbb{N}$, and we choose $a_i \in B_{2p}$ for all $i \in I$ such that

$$a_i\partial^p(a_i)a_j = \partial^p(a_j)a_i\partial^p(a_i) \quad \forall i, j \in I. \tag{2}$$

Since $a_i \in B_{2p}$, we have $[a_i, \partial^{2p}(x)] = 1$ for all $x \in B_\infty$. Thus the conditions of Proposition 2.6 are fulfilled, and $x *_i y = x\partial^p(y)a_i\partial^p(x^{-1})$ defines an multi-LD-structure on B_∞. For $|I| = 1$, $p = 1$ and $a = \sigma_1$, which implies $H = B_\infty$, we get Dehornoy's original definition of shifted conjugacy $*$.
It remains to give some natural solutions $\{a_i \in B_{2p} \mid i \in I\}$ of the equation set (1). Note that in case $|I| = 1$ (notation: $a_1 = a$), of course, every endomorphism f of B_∞ with $f(\sigma_1) \in B_{2p}$ provides such solution $a = f(\sigma_1)$.

Definition 2.8 *(Definition I.4.6. in [De00]) Let, for $n \geq 2$, be $\delta_n = \sigma_{n-1} \cdots \sigma_2 \sigma_1$. For $p, q \geq 1$, we set*

$$\tau_{p,q} = \delta_{p+1} \partial(\delta_{p+1}) \cdots \partial^{q-1}(\delta_{p+1}).$$

Since $a = \tau_{p,p}^{\pm 1} \in B_{2p}$ fulfills $a\partial^p(a)a = \partial^p(a)a\partial^p(a)$, it provides a lot of (multi)-LD-structures on B_∞.

Proposition 2.9 *(a) The binary operation $x *_a y = \partial^p(x^{-1})a\partial^p(y)x$ with $a = a'\tau_{p,p}a''$ for some $a', a'' \in B_p$ yields an LD-structure on $B_{\mathbb{N}}$ if and only if $[a', a''] = 1$.*
(b) Let I be an index set. The binary operations $x *_i y = \partial^p(x^{-1})a_i\partial^p(y)x$ with $a_i = a_i'\tau_{p,p}a_i''$ for some $a_i', a_i'' \in B_p$ $(i \in I)$ yields a multi-LD-structure on $B_{\mathbb{N}}$ if and only if $[a_i', a_j'] = [a_i', a_j''] = 1$ for all $i, j \in I$. (Note that a_i'' and a_j'' needn't commute for $i \neq j$.)
(c) The binary operations $x *_i y = \partial^p(x^{-1})a_i\partial^p(y)x$ $(i = 1, 2)$ with $a_1 = a_1'\tau_{p,p}a_1''$, $a_2 = a_2'\tau_{p,p}^{-1}a_2''$ for some $a_1', a_1'', a_2', a_2'' \in B_p$ yields a bi-LD-structure on $B_{\mathbb{N}}$ if and only if $[a_1', a_1''] = [a_2', a_2''] = [a_1', a_2''] = [a_2', a_1''] = [a_1', a_2'] = 1$. (Note that a_1'' and a_2'' needn't commute.)

We see that there exist infinitely many (multi)-LD-structures on $B_{\mathbb{N}}$. Further examples are provided by Proposition 2.10, which, of course, admits a lot of variations and generalizations.

Proposition 2.10 *Let be $p, p_1, p_2 \in \mathbb{N}$ with $p_1 + p_2 = p$. The binary operation $x *_a y = \partial^p(x^{-1})a\partial^p(y)x$ with*

$$a = a_1'\partial^{p_1}(a_2')\partial^{p_1}(\tau_{p_2,p})\tau_{p,p_1}^{-1}a_1''\partial^{p_1}(a_2'')$$

for some $a_1', a_1'' \in B_{p_1}$, $a_2', a_2'' \in B_{p_2}$ yields an LD-structure on $B_{\mathbb{N}}$ if and only if $[a_1', a_1''] = [a_2', a_2''] = 1$.

The proofs of Proposition 2.9 and 2.10 are straightforward computations. The reader is recommended to draw some pictures.

2.5 Yet another group-based LD-system

Though we are sure that it must have been well known to experts, we haven't been able to find the following natural LD-operation for groups in the literature.
For a group G, $(G, \circ)$ is an LD-system with $x \circ y = xy^{-1}x$.
Note that, contrary to the conjugacy operation $*$, for this *"symmetric decomposition" or conjugacy operation* $\circ$, the corresponding relation $\longrightarrow_\circ$ defined by $x \longrightarrow_\circ y$ iff $\exists c \in G$ such that $y = c \circ x$) is not an equivalence relation. In particular, $\longrightarrow_\circ$ is reflexive and symmetric, but not transitive.
One may consider several generalizations of this symmetric conjugacy operation $\circ$, as candidates for natural LD-operations in groups. Consider an Ansatz like $x \circ y = f(x)g(y^{-1})h(x)$ for some group endomorphisms f, g, h.

Proposition 2.11. *Let G be a group, and $f, g, h \in End(G)$. Then the binary operation $x \circ y = f(x) \cdot g(y^{-1}) \cdot h(x)$ yields an LD-structure on G if and only if*

$$f^2 = f, \quad fh = gh = fg, \quad hg = gf = hf, \quad h^2 = h. \tag{3}$$

PROOF. A straightforward computation yields

$$\begin{aligned} \alpha \circ (\beta \circ \gamma) &= f(\alpha)gh(\beta^{-1})g^2(\gamma)gf(\beta^{-1})h(\alpha), \quad \text{and} \\ (\alpha \circ \beta) \circ (\alpha \circ \gamma) &= f^2(\alpha)fg(\beta^{-1})fh(\alpha)gh(\alpha^{-1})g^2(\gamma)gf(\alpha^{-1})hf(\alpha)hg(\beta^{-1})h^2(\alpha). \end{aligned}$$

A comparison of both terms yields the assertion. □

Except for $f^2 = f = g = h = h^2$, the simplest solutions of the system of equations (3) are $f^2 = f = g$ and $h = \mathrm{id}$, or $f = \mathrm{id}$ and $g = h = h^2$.

Corollary 2.12. *(*LD- OR f-SYMMETRIC CONJUGACY*) Let G be a group, and $f \in End(G)$ an endomorphism that is also a projector ($f^2 = f$). Then $(G, \circ_f)$ and $(G, \circ_f^{\mathrm{rev}})$, defined by $x \circ_f y = f(xy^{-1})x$ and $x \circ_f^{\mathrm{rev}} y = xf(y^{-1}x)$, are LD-systems.*

Proposition 2.13. *Let G be a group, and $f, g \in End(G)$.*
(i) Then the binary operations $\circ_f$ and $*_f$ (and $*_f^{\mathrm{rev}}$), defined by $x \circ_f y = f(x) \cdot g(y^{-1}) \cdot h(x)$ and $x *_f y = f(x^{-1} \cdot y) \cdot h(x)$ ($x *_f^{\mathrm{rev}} y = x \cdot f(y \cdot x^{-1})$), are distributive over $\circ$. In particular $*$ ($*^{\mathrm{rev}}$) is distributive over $\circ$. In short, the following equations hold.

$$x *_f (y \circ z) = (x *_f y) \circ (x *_f z), \quad x \circ_f (y \circ z) = (x \circ_f y) \circ (\circ_f z) \forall x, y, z \in G.$$

(ii) The operations $\circ_f$ and $*_f$ ($*_f^{\mathrm{rev}}$) are distributive over $\circ_g$ if and only if $f = gf = fg$.

3 Key establishment for all LD-systems

3.1 The protocol

Recall that a *magma* is a set M equipped with a binary operation, say $\bullet$, which is possibly non-associative. For our purposes all interesting LD-systems are non-associative. Consider an element y of a magma $(M, \bullet)$ which is an iterated product of other elements in M. Such an element can be described by a planar rooted binary tree T whose k leaves are labelled by these other elements $y_1, \ldots, y_k \in M$. We use the notation $y = T_\bullet(y_1, \ldots, y_k)$. Here the subscript $\bullet$ tells us that the grafting of subtrees of T corresponds to the operation $\bullet$.
Now, it is easy to prove by induction (over the depth of the involved trees) that any magma homomorphism $\beta : (M, \bullet) \to (N, \circ)$ satisfies

$$\beta(T_\bullet(y_1, \ldots, y_k)) = T_\circ(\beta(y_1), \ldots, \beta(y_k))$$

for all $y_1, \ldots, y_k \in M$.

Proposition 3.1. *Let $(L, *)$ be an LD-system. Then, for any element $x \in L$, the left multiplication map $\phi_x : y \mapsto x * y$ defines a magma endomorphism of L.*

PROOF $\phi_x(y_1 * y_2) = x * (y_1 * y_2) \overset{LD}{=} (x * y_1) * (x * y_2) = \phi_x(y_1) * \phi_x(y_2)$. □

We are going to describe a KEP that applies to any LD-system $(L, *)$. There are two public submagmas $S_A = \langle s_1, \cdots, s_m \rangle_*$, $S_B = \langle t_1, \cdots, t_n \rangle_*$ of $(L, *)$, assigned to Alice and Bob. Alice and Bob perform the following protocol steps.

Protocol 1: KEY ESTABLISHMENT FOR ANY LD-SYSTEM $(L, *)$.

1. Alice generates her secret key $(a_0, a) \in S_A \times L$, and Bob chooses his secret key $b \in S_B$

2. Alice computes the elements $a*t_1, \ldots, a*t_n, p_0 = a*a_0 \in L$, and sends them to Bob. Bob computes $b*s_1, \ldots, b*s_m \in L$, and sends them to Alice.

3. Alice, knowing $a_0 = T_*(r_1, \ldots, r_k)$ with $r_i \in \{s_1, \ldots, s_m\}$, computes from Bob's public key

$$T_*(b*r_1, \ldots, b*r_k) = b*T_*(r_1, \ldots, r_k) = b*a_0.$$

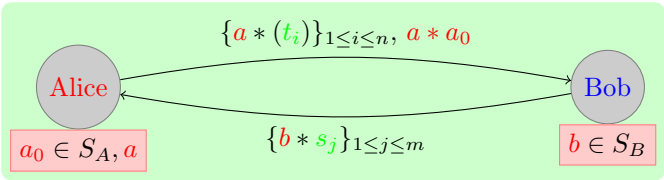
   And Bob, knowing $b = T'_*(u_1, \ldots, u_{k'}$ with $u_j \in \{t_1, \ldots, t_n\}$, computes from Alice's public key

$$T'_*(a*u_1, \ldots, a*u_{k'}) = a*T'_*(u_1, \ldots, u_{k'}) = a*b.$$

4. Alice computes $K_A = a*(b*a_0)$. Bob gets the shared key by $K_B := (a*b)*p_0 = (a*b)*(a*a_0) \overset{(LD)}{=} K_A$.

This protocol is an asymmetric modification of the Anshel-Anshel-Goldfeld protocols for magmas introduced in [Ka07, Ka12].

Figure 1: PROTOCOL 1: KEY ESTABLISHMENT FOR ANY LD-SYSTEM



3.2 Base problems

In order to break Protocol 1 an attacker has to find the shared key $K = K_A = K_B$. A successful attack on Bob's secret key b requires the solution of

m-simLDP (m-simultaneous LD-Problem):
INPUT: Element pairs $(s_1, s'_1), \ldots, (s_m, s'_m) \in L^2$ with $s'_i = b*s_i$ $\forall 1 \le i \le m$ for some (unknown) $b \in L$.
OBJECTIVE: Find $b' \in L$ with $b'*s_i = s'_i$ for all $i = 1, \ldots, m$.

Even if an attacker finds Bob's original key b or a pseudo-key b' (solution to the m-simLDP above), then she still faces the following problem.

$*$-MSP ($*$-submagma Membership Search Problem):
INPUT: $t_1, \ldots, t_n \in (L, *)$, $b \in \langle t_1, \ldots, t_n \rangle_*$.
OBJECTIVE: Find an expression of b as a tree-word in the submagma $\langle t_1, \ldots, t_n \rangle_*$ (notation $b = T_*(u_1, \ldots, u_k)$ for $u_i \in \{t_j\}_{j\le n}$).

Proposition 3.2. *Let $(L, *)$ be an LD-system. We define the generalized m-simLDP for $S_B \subseteq L$ as an m-simultaneous LD-Problem with the objective to find a b' in $S_B = \langle t_1, \ldots, t_n \rangle_*$ such that $b' * s_i = s'_i$ for all $i \leq m$.*
An oracle that solves the generalized m-simLDP and $*$-MSP for S_B is sufficient to break key establishment Protocol 1.

Proof. As outlined above, we perform an attack on Bob's private key. The generalized m-simLDP oracle provides a pseudo-key $b' \in S_B$ with $b' * s_i = s'_i = b * s_i$ for all $i = 1, \ldots, m$. Observe that this implies for any element $e_A \in S_A$ that $b' * e_A = b * e_A$. In particular, we have $b' * a_0 = b * a_0$. We feed this pseudo-key b' into a $*$-MSP oracle for S_B which returns a treeword $T'_*(u_1, \ldots, u_l) = b'$ (for some $l \in \mathbb{N}$ and $u_i \in \{t_j\}_{j \leq n}$). Now compute

$$\begin{aligned} T'_*(a * u_1, \ldots, a * u_l) * p_0 &\overset{LD}{=} (a * T'_*(u_1, \ldots, u_l)) * p_0 = (a * b') * (a * a_0) \\ &\overset{LD}{=} a * (b' * a_0) = a * (b * a_0) = K. \quad \square \end{aligned}$$

Note that here the situation is asymmetric - an attack on Alice's secret key requires the solution of the following problem.

n-modsimLDP (Modified n-Simultaneous LD-Problem):
Input: An element $p_0 \in L$ and pairs $(t_1, t'_1), \ldots, (t_n, t'_n) \in L^2$ with $t'_i = a * t_i$ $\forall 1 \leq i \leq n$ for some (unknown) $a \in L$.
Objective: Find elements $a'_0, a' \in L$ such that $p_0 = a' * a'_0$ and $a' * t_i = t'_i$ for all $i = 1, \ldots, n$.

Also here, even if an attacker finds Alice's original key $(a0, a)$ or a pseudo-key $(a'_0, a') \in S_A \times L$, then she still faces a $*$-submagma Membership Search Problem.

Proposition 3.3. *Let $(L, *)$ be an LD-system. We define the generalized n-modsimLDP for $S_A \subseteq L$ as a modified n-simultaneous LD-Problem with the objective to find a $a' \in L$ and a'_0 in $S_A = \langle s_1, \ldots, s_m \rangle_*$ such that $a' * t_i = t'_i$ for all $i \leq n$.*
An oracle that solves the generalized n-modsimLDP and $*$-MSP for S_A is sufficient to break key establishment Protocol 1.

Proof. As outlined above, we perform an attack on Alice's private key. The generalized n-simLDP oracle provides a pseudo-key $(a'_0, a')' \in S_A \times L$ such that $a' * a'_0 = p_0$ and $a' * t_i = a'_i = a * t_i$ for all $i = 1, \ldots, n$. Observe that this implies for any element $e_B \in S_B$ that $a' * e_B = a * e_B$. In particular, we have $a' * b = a * b$. We feed the first component $a'_0 \in S_A$ of this pseudo-key into a $*$-MSP oracle for S_A which returns a treeword $T'_*(r_1, \ldots, r_l) = a'_0$ (for some $l \in \mathbb{N}$ and $r_i \in \{s_j\}_{j \leq m}$). Now, we compute

$$\begin{aligned} a' * T'_*(b * r_1, \ldots, b * r_l) &\overset{LD}{=} a' * (b * T'_*(r_1, \ldots, r_l)) = a' * (b * a'_0) \\ &\overset{LD}{=} (a' * b) * (a' * a'_0) = (a * b) * p_0 = K. \quad \square \end{aligned}$$

Both appproaches described above require the solution of a $*$-submagma Membership Search Problem. Note that we assumed that the generalized m-simLDP (resp. n-modsimLDP) oracle already provides a pseudo-key in the

submagma S_B (resp. S_A) which we feed to the $*$-MSP oracle. But to check whether an element lies in some submagma, i.e. the $*$-submagma Membership Decision Problem, is already undecidable in general.
Fortunately, for the attacker, there are approaches which do not resort to solving the $*$-MSP.

Recall that we defined the generalized m-simLDP for $S_B \subseteq L$ as an m-simultaneous LD-Problem with the objective to find a b' in $S_B = \langle t_1, \ldots, t_n \rangle_*$ such that $b' * s_i = s'_i$ for all $i \leq m$.

Proposition 3.4. *An generalized simLDP oracle is sufficient to break key establishment Protocol 1. More precisely, an oracle that solves the generalized m-simLDP for S_B and the n-simLDP is sufficient to break KEP1.*

Proof. Here we perform attacks on Alice's and Bob's private keys - though we need only a pseudo-key for the second component a' of Alice's key. The n-simLDP oracle provides $a' \in L$ s.t. $a' * t_j = t'_j = a * t_j$ for all $j \leq n$. And the generalized m-simLDP oracle returns the pseudo-key $b' \in S_B$ s.t. $b' * s_i = s'_i = b * s_i$ for all $i \leq m$. Since $b' \in S_B$, we conclude that $a' * b' = a * b'$. Also, $a_0 \in S_A$ implies, of course, $b' * a_0 = b * a_0$. Now, we may compute

$$(a' * b') * p_0 = (a * b') * (a * a_0) \overset{LD}{=} a * (b' * a_0) = a * (b * a_0) = K. \quad \square$$

Recall that we defined the generalized n-modsimLDP for $S_A \subseteq L$ as an n-simultaneous LD-Problem with the objective to find a a'_0 in $S_A = \langle s_1, \ldots, s_m \rangle_*$ such that $a' * t_i = t'_i$ for all $i \leq n$.

Proposition 3.5. *An oracle that solves the generalized n-modsimLDP for S_A and the m-simLDP is sufficient to break KEP1.*

Proof. Also here we perform attacks on Alice's and Bob's private keys. The m-simLDP oracle provides $b' \in L$ s.t. $b' * s_j = s'_j = b * s_j$ for all $j \leq m$. And the generalized n-modsimLDP oracle returns the pseudo-key $(a'_0, a') \in S_A \times L$ s.t. $a' * t_i = t'_i = a * t_i$ for all $i \leq n$ *and* $a' * a'_0 = p_0$. Since $a'_0 \in S_A$, we conclude that $b' * a'_0 = b * a'_0$. Also, $b \in S_B$ implies, of course, $a' * b = a * b$. Now, we compute

$$a' * (b' * a'_0) = a' * (b * a'_0) \overset{LD}{=} (a' * b) * (a' * a'_0) = (a * b) * p_0 = K. \quad \square$$

4 Key establishment for leftdistributive systems

4.1 The protocol

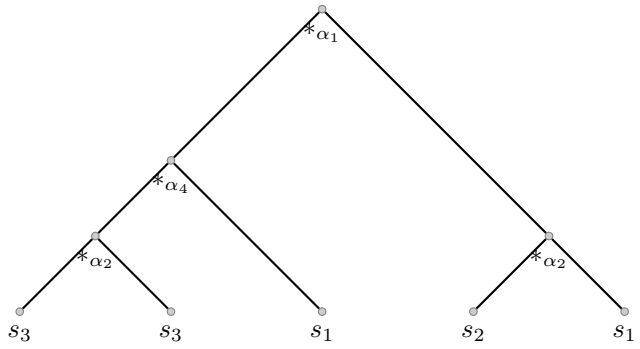
Here we describe a generalization of Protocol 1 that works for all multi-LD-systems. Actually, it suffices if L is only a partial multi-LD-system, i.e. some distributive laws hold. More precisely, consider a set L equipped with a pool of binary operations $O_A \cup O_B$ (O_A and O_B non-empty) s.t. the operations in O_A are distributive over those in O_B and vice versa, i.e. the following holds for all $x, y, z \in L$, $*_\alpha \in O_A$ and $*_\beta \in O_B$.

$$x *_\alpha (y *_\beta z) = (x *_\alpha y) *_\beta (x *_\alpha z), \text{ and} \tag{4}$$

$$x *_\beta (y *_\alpha z) = (x *_\beta y) *_\alpha (x *_\beta z). \tag{5}$$

Note that, if $O_A \cap O_B \neq \emptyset$, then $(L, O_A \cup O_B)$ is a multi-LD-system.
Let $s_1, \ldots, s_m, t_1, \ldots, t_n \in L$ be some public elements. We denote $S_A = \langle s_1, \cdots, s_m \rangle_{O_A}$ and $S_B = \langle t_1, \cdots, t_n \rangle_{O_B}$, two submagmas of $(L, O_A \cup O_B)$. For example, an element y of S_A can be described by a planar rooted binary tree T whose k leaves are labelled by these other elements $r_1, \ldots, r_k$ with $r_i \in \{s_i\}_{i \leq m}$. Here the tree contains further information, namely to each internal vertex we assign a binary operation $*_i \in O_A$. We use the notation $y = T_{O_A}(r_1, \ldots, r_k)$. The subscript O_A tells us that the grafting of subtrees of T corresponds to the operation $*_i \in O_A$. Consider, for example, the element $y = ((s_3 *_{\alpha_2} s_3) *_{\alpha_4} s_1) *_{\alpha_1} (s_2 *_{\alpha_2} s_1)$. The corresponding labelled planar rooted binary tree T is displayed in the following figure.

Figure 2: The element $y = ((s_3 *_{\alpha_2} s_3) *_{\alpha_4} s_1) *_{\alpha_1} (s_2 *_{\alpha_2} s_1) \in S_A$



Let $*_\alpha \in O_A$ and $*_\beta \in O_B$. By induction over the tree depth, it is easy to show that, for all elements $e, e_1, \ldots, e_l \in (L, O_A \cup O_B)$ and all planar rooted binary trees T with l leaves, the following equations hold.

$$e *_\alpha T_{O_B}(e_1, \ldots, e_l) = T_{O_B}(e *_\alpha e_1, \ldots, e *_\alpha e_l), \quad (6)$$

$$e *_\beta T_{O_A}(e_1, \ldots, e_l) = T_{O_A}(e *_\beta e_1, \ldots, e *_\beta e_l). \quad (7)$$

Now, we are going to describe a KEP that applies to any system $(L, O_A \cup O_B)$ as described above. We have two subsets of public elements $\{s_1, \cdots, s_m\}$ and $\{t_1, \cdots, t_n\}$ of L. Also, recall that $S_A = \langle s_1, \cdots, s_m \rangle_{O_A}$ and $S_B = \langle t_1, \cdots, t_n \rangle_{O_B}$. Alice and Bob perform the following protocol steps.

Protocol 2: KEY ESTABLISHMENT FOR THE PARTIAL MULTI-LD-SYSTEM $(L, O_A \cup O_B)$.

1. Alice generates her secret key $(a_0, a, *_\alpha) \in S_A \times L \times O_A$, and Bob chooses his secret key $(b, *_\beta) \in S_B \times O_B$.

2. Alice computes the elements $a *_\alpha t_1, \ldots, a *_\alpha t_n, p_0 = a *_\alpha a_0 \in L$, and sends them to Bob. Bob computes $b *_\beta s_1, \ldots, b *_\beta s_m \in L$, and sends them to Alice.

3. Alice, knowing $a_0 = T_{O_A}(r_1, \ldots, r_k)$ with $r_i \in \{s_1, \ldots, s_m\}$, computes from Bob's public key

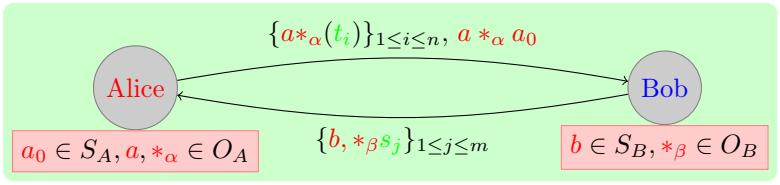
$$T_{O_A}(b *_\beta r_1, \ldots, b *_\beta r_k) = b *_\beta T_{O_A}(r_1, \ldots, r_k) = b *_\beta a_0.$$

And Bob, knowing $b = T'_{O_B}(u_1, \ldots, u_{k'})$ with $u_j \in \{t_1, \ldots, t_n\}$, computes from Alice's public key

$$T'_{O_B}(a *_\alpha u_1, \ldots, a *_\alpha u_{k'}) = a *_\alpha T'_{O_B}(u_1, \ldots, u_{k'}) = a *_\alpha b.$$

4. Alice computes $K_A = a *_\alpha (b *_\beta a_0)$. Bob gets the shared key by $K_B := (a *_\alpha b) * p_0 = (a *_\alpha b) *_\beta (a *_\alpha a_0) \overset{LD}{=} K_A$.

Figure 3: KEP FOR THE PARTIAL MULTI-LD-SYSTEM $(L, O_A \cup O_B)$.



Here the operations $*_\alpha \in O_A$ and $*_\beta \in O_B$ are part of Alice's and Bob's private keys. As in Protocol 1, explicit expressions of $a_0 \in S_A$ and $b \in S_B$ as treewords T, T' are also parts of the private keys - though we did not mention it explicitly in step 1 of the protocols. But here T_{O_A} and T'_{O_B} also contain all the information about the grafting operations (in O_A or O_B, respectively) at the internal vertices of T, T'.

4.2 Base problems

In order to break Protocol 2 an attacker has to find the shared key $K = K_A = K_B$. A successful attack on Bob's secret key $(b, *_\beta)$ requires (first) the solution of the following problem.

INPUT: Element pairs $(s_1, s'_1), \ldots, (s_m, s'_m) \in L^2$ with $s'_i = b *_\beta s_i$ $\forall 1 \le i \le m$ for some (unknown) $b \in L$, $*_\beta \in O_B$.
OBJECTIVE: Find $b' \in L$ and $*_{\beta'} \in O_B$ such that $b' *_{\beta'} s_i = s'_i$ for all $i = 1, \ldots, m$.

In order to clarify concepts we introduce the following notation, which also makes it easier to name our base problems at hand. For $e \in L$, let $\phi_{e,\alpha}(x) := e *_\alpha x$. Then, for $*_\alpha \in O_A$, $\phi_{e,\alpha}$ is by (4) a magma homomorphism on S_B. Analogeously, for $*_\beta \in O_B$, $\phi_{e,\beta} \in End(S_A)$ by (5). Now, we may reformulate the base problem for obtaining a pseudo-key on Bob's secret.

LDEndP (LD-endomorphism Search Problem for S_A):
INPUT: Element pairs $(s_1, s'_1), \ldots, (s_m, s'_m) \in L^2$ with $s'_i = \phi_{b,\beta}(s_i)$ $\forall 1 \le i \le m$ for some (unknown) magma endomorphism $\phi_{b,\beta} \in End(S_A)$ (with $*_\beta \in O_B$).
OBJECTIVE: Find magma endomorphism $\phi_{b',\beta'} \in End(S_A)$ $(*_\beta \in O_B)$ such that $\phi_{b',\beta'}(s_i) = s'_i$ for all $i = 1, \ldots, m$.

Recall that we work in the leftdistributive system $(L, O_A \cup O_B)$. We define the *generalized LDEndP* for (S_A, S_B) as an LD-endomorphism Search Problem

for S_A with the objective to find a magma endomorphism $\phi_{b',\beta'} \in End(S_A)$ with $*_{\beta'} \in O_B$ *and* b' in $S_B = \langle t_1, \ldots, t_n \rangle_{O_B}$.
Even if an attacker finds a pseudo-key endomorphism $\phi_{b',\beta'} \in End(S_A)$, then she still faces the following problem.

O_B**-MSP** (O_B-submagma Membership Search Problem for S_B):
INPUT: $t_1, \ldots, t_n \in L$, $b \in S_B = \langle t_1, \ldots, t_n \rangle_{O_B}$.
OBJECTIVE: Find an expression of b as a tree-word (with internal vertices labelled by operations in O_B) in the submagma S_B (notation $b = T_{O_B}(u_1, \ldots, u_k)$ for $u_i \in \{t_j\}_{j \leq n}$).

Proposition 4.1. *An oracle that solves the generalized LDEndP for* (S_A, S_B) *and* O_B*-MSP for* S_B *is sufficient to break key establishment Protocol 2.*

PROOF. As outlined above, we perform an attack on Bob's private key. The generalized LDEndP for (S_A, S_B) oracle provides a pseudo-key endomorphism $\phi_{b',\beta'} \in End(S_A)$ with $b' \in S_B$, $*_{\beta'} \in O_B$ such that $\phi_{b',\beta'}(s_i) = s'_i = \phi_{b,\beta}(s_i)$ for all $i = 1, \ldots, m$. Observe that this implies for any element $e_A \in S_A$ that $\phi_{b',\beta'}(e_A) = \phi_{b,\beta}(e_A)$. In particular, we have $\phi_{b',\beta'}(a_0) = \phi_{b,\beta}(a_0)$. Since $b' \in S_B$, we may feed b' into a O_B-MSP oracle for S_B which returns a tree-word $T'_{O_B}(u_1, \ldots, u_l) = b'$ (for some $l \in \mathbb{N}$ and $u_i \in \{t_j\}_{j \leq n}$). Now, we compute

$$\begin{aligned} & T'_{O_B}(a *_\alpha u_1, \ldots, a *_\alpha u_l) *_{\beta'} p_0 \overset{LD}{=} (a *_\alpha T'_{O_B}(u_1, \ldots, u_l)) *_{\beta'} p_0 \\ = \quad & (a *_\alpha b') *_{\beta'} (a *_\alpha a_0) \overset{LD}{=} a *_\alpha (b' *_{\beta'} a_0) = a *_\alpha (b *_\beta a_0) = K. \quad \square \end{aligned}$$

On the other hand, an attack on Alice's secret key requires (first) the solution of the following problem.

modLDEndP (Modified LD-endomorphism Search Problem for S_B):
INPUT: Element pairs $(t_1, t'_1), \ldots, (t_n, t'_n) \in L^2$ with $t'_i = \phi_{a,\alpha}(t_i)$ $\forall 1 \leq i \leq n$ for some (unknown) magma endomorphism $\phi_{a,\alpha} \in End(S_B)$ (with $*_\alpha \in O_A$). Furthermore, an element $p_0 \in \phi_{a,\alpha}(S_A)$, i.e. $p_0 = \phi_{a,\alpha}(a_0)$ for some $a_0 \in S_A$.
OBJECTIVE: Find $(a'_0, \phi_{a',\alpha'}) \in L \times End(S_B)$ ($*_\alpha \in O_A$) such that $\phi_{a',\alpha'}(t_i) = t'_i$ for all $i = 1, \ldots, n$ *and* $\phi_{a',\alpha'}(a'_0) = p_0$.

We define the *generalized modLDEndP* for (S_B, S_A) as a modified LD-endomorphism Search Problem for S_B with the objective to find $(a'_0, \phi_{a',\alpha'}) \in S_A \times End(S_B)$ ($*_\alpha \in O_A$) such that $\phi_{a',\alpha'}(t_i) = t'_i$ for all $i = 1, \ldots, n$ *and* $\phi_{a',\alpha'}(a'_0) = p_0$.
Even if an attacker finds a pseudo-key $(a'_0, \phi_{a',\alpha'}) \in S_A \times End(S_B)$ for Alice's secret, then she still faces a O_A-submagma Membership Search Problem for S_A.

Proposition 4.2. *An oracle that solves the generalized modLDEndP for* (S_B, S_A) *and* O_A*-MSP for* S_A *is sufficient to break key establishment Protocol 1.*

PROOF. As outlined above, we perform an attack on Alice's private key. The generalized modLDEndP oracle provides a pseudo-key $(a'_0, \phi_{a',\alpha'}) \in S_A \times End(S_B)$ such that $\phi_{a',\alpha'}(t_i) = t'_i = \phi_{a,\alpha}(t_i)$ for all $i = 1, \ldots, n$ *and* $\phi_{a',\alpha'}(a'_0) = p_0$. Observe that this implies for any element $e_B \in S_B$ that $\phi_{a',\alpha'}(e_B) = \phi_{a,\alpha}(e_B)$. In particular, we have $\phi_{a',\alpha'}(b) = \phi_{a,\alpha}(b)$. Since $a'_0 \in S_A$, we may feed a'_0 into a

O_A-MSP oracle for S_A which returns a tree-word $T'_{O_A}(r_1,\dots,r_l) = a'_0$ (for some $l \in \mathbb{N}$ and $r_i \in \{s_j\}_{j\leq m}$). Now, we may compute

$$\begin{aligned} & a' *_{\alpha'} T'_{O_A}(b *_\beta r_1,\dots,b *_\beta r_l) \overset{LD}{=} a' *_{\alpha'} (b *_\beta T'_{O_A}(r_1,\dots,r_l)) \\ = \;& a' *_{\alpha'} (b *_\beta a'_0) \overset{LD}{=} (a' *_{\alpha'} b) *_\beta (a' *_{\alpha'} a'_0) = (a *_\alpha b) *_\beta p_0 = K. \quad \square \end{aligned}$$

Now, we describe approaches to break Protocol 2 which do not resort to solving a submagma-MSP.

Proposition 4.3. *An generalized LDEndP oracle is sufficient to break key establishment Protocol 2. More precisely, an oracle that solves the generalized LDEndP for (S_A, S_B) and the LDEndP for S_B is sufficient to break KEP1.*

Proof. Here we perform attacks on Alice's and Bob's private keys - though we do not require a pseudo-key for the first component a_0 of Alice's key. The LDEndP oracle for S_B provides $\phi_{a',\alpha'}$ s.t. $\phi_{a',\alpha'}(t_j) = t'_j = \phi_{a,\alpha}(t_j)$ for all $j \leq n$. And the generalized LDEndP oracle for (S_A, S_B) returns the pseudo-key endomorphism $\phi_{b',\beta'}$ with $b' \in S_B$ s.t. $\phi_{b',\beta'}(s_i) = s'_i = \phi_{b,\beta}(s_i)$ for all $i \leq m$. Since $b' \in S_B$, we conclude that $\phi_{a',\alpha'}(b') = \phi_{a,\alpha}(b')$. Also, $a_0 \in S_A$ implies, of course, $\phi_{b',\beta'}(a_0) = \phi_{b,\beta}(a_0)$. Now, we compute

$$(a' *_{\alpha'} b') *_{\beta'} p_0 = (a *_\alpha b') *_{\beta'} (a *_\alpha a_0) \overset{LD}{=} a *_\alpha (b' *_{\beta'} a_0) = a *_\alpha (b *_\beta a_0) = K. \quad \square$$

Alternatively, one may choose the following approach.

Proposition 4.4. *An oracle that solves the generalized modLDEndP for (S_B, S_A) and the LDEndP for S_A is sufficient to break KEP1.*

Proof. Also here we perform attacks on Alice's and Bob's private keys. The LDEndP oracle for S_A provides $\phi_{b',\beta'} \in End(S_A)$ s.t. $\phi_{b',\beta'}(s_j) = s'_j = \phi_{b',\beta'}(s_j)$ for all $j \leq m$. And the generalized modLDEndP oracle for (S_B, S_A) returns the pseudo-key $(a'_0, \phi_{a',\alpha'}) \in S_A \times End(S_B)$ s.t. $\phi_{a',\alpha'}(t_i) = t'_i = \phi_{a',\alpha'}(t_i)$ for all $i \leq n$ *and* $\phi_{a',\alpha'}(a'_0) = p_0$. Since $a'_0 \in S_A$, we conclude that $\phi_{b',\beta'}(a'_0) = \phi_{b,\beta}(a'_0)$. Also, $b \in S_B$ implies, of course, $\phi_{a',\alpha'}(b) = \phi_{a,\alpha}(b)$. Now, we compute

$$a' *_{\alpha'} (b' *_{\beta'} a'_0) = a' *_{\alpha'} (b *_\beta a'_0) \overset{LD}{=} (a' *_{\alpha'} b) *_\beta (a' *_{\alpha'} a'_0) = (a *_\alpha b) *_\beta p_0 = K. \quad \square$$

5 Discussion

We leave this section short and confine our selves to some few remarks, mainly on shifted conjugacy in braid groups. For further discussion and open problems we refer the reader to section 5.2 in [Ka12].
(1) Note that in the non-associative setting the case $m = n = 1$ is particular interest, i.e. we may *abandon simultaneity* in our base problems since the submagmas generated by one element are still complicated objects.
(2) Consider the infinite braid group $(B_\infty, *)$ with shifted conjugacy as LD-operation. Then the LD-Problem ($m = n = 1$) is the *shifted conjugacy problem* (see e.g. [De06]) which was first solved in [KLT09]. If we replace shifted conjugacy by generalized shifted conjugacy, then the corresponding LD-problem still

reduces to a subgroup conjugacy problem for a standard parabolic subgroup of a braid group. Such problems were first solved (in a more general framework) in [KLT10]. Though these solutions provide only deterministic algorithms with exponential worst case complexity, they may still affect the security of Protocol 1 if we use such LD-systems in braid groups as platform LD-systems.
(3) Here we point out that the deterministic algorithms from [KLT09, KLT10] do not affect the security of Protocol 2 if we consider the following natural partial multi-LD-system $(B_\infty, O_A \cup O_B)$ in braid groups. Let $p = 2q$ be odd. Let any $*_\alpha \in O_A$ be of the form $x *_\alpha y = \partial^p(x^{-1})\alpha\partial(y)x$ with $\alpha = \alpha'\tau_{p,p}\alpha''$ for some $\alpha', \alpha'' \in B_q$. Analogously, any $*_\beta \in O_B$ is of the form $x *_\beta y = \partial^p(x^{-1})\beta\partial(y)x$ with $\beta = \beta'\tau_{p,p}\beta''$ for some $\beta', \beta'' \in \partial^q(B_q)$. Since α', α'' commute with β', β'', $(B_\infty, *_\alpha, *_\beta)$ is for any $*_\alpha \in O_A$ and $*_\beta \in O_B$ a bi-LD-system. The deterministic algorithms from [KLT09, KLT10] do not apply because the *operations are part of the secret*. Actually, the attacker has then to solve (still in the case $n = 1$) a very special *decomposition problem*, namely, given $t, t' \in B_\infty$ such that $t' = \partial^p(a^{-1})\tilde{\alpha}\tau_{p,p}\partial(t)a$ for some $a \in B_\infty$, $\tilde{\alpha} \in B_q \cdot \partial^p(B_q)$, find $a, \tilde{\alpha}$ or some pseudokeys in the same domain.
(4) The decomposition problem mentioned above appears to be inherently quadratic, i.e. we do not see how it may be linearized such that linear algebra attacks as the *linear centralizer attack* of B. Tsaban [Ts12] apply. It shares this feature with Y. Kurt's *Triple Decomposition Problem* (see section 4.2.5. in [MSU11]).

Acknowledgements. The first author acknowledges financial support by The Oswald Veblen Fund and by the Minerva Foundation of Germany.

References

[AAG99] Iris Anshel, Michael Anshel and Dorian Goldfeld, *An algebraic method for public-key cryptography*, Mathematical Research Letters **6** (1999), 1-5.

[De00] Patrick Dehornoy, *Braids and Self-Distributivity*, Progress in Math. **192** Birkhäuser (2000).

[De06] Patrick Dehornoy, *Using shifted conjugacy in braid-based cryptography.* In: L. Gerritzen, D. Goldfeld, M. Kreuzer, G. Rosenberger and V. Shpilrain (Eds.), *Algebraic Methods in Cryptography*, Contemporary Mathematics **418**, AMS (2006), 65-73.

[Ka07] Arkadius Kalka, *Representations of braid groups and braid-based cryptography*, PhD thesis, Ruhr-Universität Bochum (2007).
`www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/KalkaArkadiusG/`

[Ka12] Arkadius Kalka, *Non-associative public key cryptography*, preprint (2012), submitted.
`http://arxiv.org/abs/1210.8270`.

[KLT09] Arkadius Kalka, Eran Liberman, and Mina Teicher , *A Note on the Shifted Conjugacy Problem in Braid Groups*, Groups - Complexity - Cryptology **1** (2) (2009), 227-230.

[KLT10] Arkadius Kalka, Eran Liberman, and Mina Teicher, *Solution to the subgroup conjugacy problem for Garside subgroups of Garside groups*, accepted, to be published in: Groups – Complexity – Cryptology (2010).

[MSU11] A.G.Myasnikov, V. Shpilrain and A.Ushakov, *Non-commutative Cryptography and Complexity of Group-theoretic Problems*, Amer. Math. Soc. Surveys and Monographs, 2011.

[Ts12] Boaz Tsaban, *Polynomial time solutions of computational problems in noncommutative-algebraic cryptography*,
preprint: `http://arxiv.org/abs/1210.8114` 2012.

E-mail addresses: `arkadius.kalka@rub.de, teicher@macs.biu.ac.il`

DEPARTMENT OF MATHEMATICS, BAR-ILAN UNIVERSITY, RAMAT GAN, ISRAEL