New Cryptosystems From CSP-Based Self-Distributive Systems

Licheng Wang[1,2], Lihua Wang[2], Zhenfu Cao[3], Eiji Okamoto[4], and Jun Shao[5]

[1] Information Security Center, Beijing University of Posts and Telecommunications
Beijing, P.R. China 100876
[2] National Institute of Information and Communications Technology (NICT)
4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
[3] Trusted Digital Technology Laboratory, Shanghai Jiao Tong University
Shanghai 200240, China
[4] University of Tsukuba, Tsukuba 305-8573, Japan
[5] College of Information Science and Technology, Pennsylvania State University
University Park 16803, USA

Abstract. We propose new cryptosystems based on self-distributive systems that are defined by conjugator searching problems (CSP) in noncommutative groups. Under the newly developed cryptographic assumptions, our basic construction is proven IND-CPA secure in the standard model. Then, we describe two extensions: The first is proven IND-CCA secure in the random oracle model, while the second achieves the IND-CCA security in the standard model. Moreover, our proposal is instantiated with braid groups, and leads to a new braid-based encryption scheme and its security is directly rooted in the intractability assumption of CSP in braid groups.

Keywords: Self-distributive systems, conjugator searching problems, braid groups, cryptosystems

1 Introduction

Most public-key cryptosystems that remain unbroken are based on the perceived difficulty of solving certain problems in large finite (abelian) groups. The theoretical foundations of these cryptosystems are related to the intractability of problems that are *closer to* number theory *than to* group theory [MST02]. In a quantum computer, most of these problems on number theory can be efficiently solved by using algorithms developed by Shor [Sho97], Kitaev [Kit95] and Proos-Zalka [PZ03]. Although the quantum computation is still in its infancy, the knowledge regarding their potential will soon create distrust in the current cryptographic methods [Lee04]. In order to enrich cryptography and not to put all eggs in one basket [Lee04], many attempts have been made to develop alternative public-key cryptography (PKC) based on different kinds of problems [AAG99,KLCH00,Lee04,MST02].

Under this background, some noncommutative groups have been attracted considerable attentions. One of the most popular groups in this category is the braid group. In 1999, Anshel et al. [AAG99] proposed an algebraic method for PKC. Shortly afterward, Ko et al. [KLCH00] published a fully fledged braid-based encryption scheme using braid groups. The security of this scheme is based on the

so-called Diffie-Hellman like conjugacy problem (DHCP), which can be viewed as a *weaker* variant of the conjugator searching problem (CSP). Unfortunately, Cha et al.'s algorithm [CJ03] announced the breaking of Ko et al.'s braid-based encryption. *After then, finding new braid-based encryption becomes an interesting challenge.*

Instead of giving a direct answer for the above challenge, in this study, we at first propose some properties of the CSP-based left self-distributive systems (though these properties could be obtained easily from the mathematic viewpoint, they are very useful from the cryptographic perspective), and then propose new cryptosystems based on these properties. Our proposal is suitable for arbitrary noncommutative group G, providing that the intractability assumption of CSP in G holds. Under a further assumption, the ciphertext of our basic scheme is proven indistinguishable against chosen plaintext attacks (IND-CPA). Then two extended schemes that achieve the IND-CCA security are described. When G is instantiated with the braid group B_n for example, we can immediately derive a new braid-based encryption scheme based on the intractability of CSP. *This scheme can be viewed as an affirmative answer for the aforementioned challenge that has been remained open over years.* In fact, our originality is enlightened by Dehornoy's previous work. In 2006, Dehorney [Deh06] proposed an authentication scheme based on self-distributive systems in braid groups. Although some cryptanalysis on Dehorney's authentication scheme were reported [LU08], we find that Dehorney's work is still meaningful at least in the following two aspects: First, self-distributive systems can be defined over arbitrary noncommutative groups, rather than braid groups only; second, self-distributive systems have the potential for building variety of cryptographic schemes, rather than authentication schemes only.

The rest of contents are organized as follows: In Section 2, we give a review on the concept of self-distributive system and the related assumption; in Section 3, we propose some properties for the CSP-based self-distributive system; more fledged cryptographic assumptions based on the intractability assumption of CSP over the underlying noncommutative group were developed in Section 4; based on these newly developed assumptions, a Diffie-Hellman-like key agreement protocol, an ElGamal-like encryption scheme and its hashed extension, as well as a Cramer-Shoup-like encryption scheme are proposed in Section 5; meanwhile, the provable security theorem, the efficiency, and other related discussions are also presented. Concluding remarks are given in Section 6.

2 Left Self-Distributive System and Hardness Assumption

Suppose that S is a non-empty set, and $F : S \times S \to S$ is a well-defined function. If the following rewrite formula holds,

$$F_r(F_s(p)) = F_{F_r(s)}(F_r(p)), \quad (\forall p, r, s \in S) \tag{1}$$

then we call $F.(\cdot)$ a *left self-distributive system*, abbreviated as LD system (See [Deh06]). The terminology "left self-distributive" arises from the following analogical

observation: If we consider $F_r(s)$ as a binary operation $r * s$, then the formula (1) becomes

$$r * (s * p) = (r * s) * (r * p), \tag{2}$$

i.e., the operation "$*$" is left distributive with respect to itself.

Given a LD system $F_\cdot(\cdot)$ defined as (1), if it is hard to retrieve s' from the given pair $(p, F_s(p))$ such that $F_s(p) = F_{s'}(p)$, then we say that this LD system is *hard*. Note that s' is not necessarily the original s. Based on a hard LD system in braid groups, Dehornoy proposed an authentication protocol [Deh06].

Remark 1. The hardness of a LD system implies the one-wayness (OW) or pre-image resistance (PR) of the same LD system, i.e., intractability of retrieving s from the pair $(p, F_s(p))$.

3 CSP-based Left Self-Distributive System and its Properties

In his seminal paper, Dehornoy[Deh06] defined a non-trivial left self-distributive systems based on the conjugate operation in braid groups. If we define a binary function F as follows,

$$F : B_n \times B_n \to B_n, \quad (a, b) \mapsto aba^{-1}, \tag{3}$$

and denote $F(a, b)$ by $F_a(b)$, then, we can see that F caters to the definition of the formula (1). Moreover, under the intractability assumption of the conjugator searching problem (CSP) in braid groups, this LD system is hard. That is, it is hard to find s for a given pair $(p,\ sps^{-1})$, where s and p are two braids belonging to the braid group B_n.

Apparently, Dehornoy's definition on LD system is suitable for arbitrary non-commutative group G, rather then braid groups only. Thus, let us name all this category of LD systems as CSP-based LD systems. We have developed some properties of CSP-based LD systems. From the cryptographic perspective, these properties are very useful under the aforementioned hardness assumption.

Remark 2. We must take care of the relationship between the intractability assumption of CSP and the hardness assumption of LD systems. The CSP problem is defined here as a *worst-case* problem, whereas from the cryptographic perspective, one needs average-case hardness for a LD system. Therefore, we need a special sample algorithm that can produce the hardest instances of CSP in a particular non-commutative G. For a general noncommutative group G, it is difficult to discuss whether we can sample hardest instances. But for braid groups, Ko et al. [KLT08] proposed some methods for generating hard CSP instances for braid cryptography. In sequel, the CSP instances used in our proposal are always assumed to be hard.

Proposition 1. *Suppose that F is a CSP-based LD system defined over a noncommutative group G. Then, for arbitrary $a, b, c \in G$, F satisfies the following properties:*

(i) F is idempotent in the sense of $F_a(a) = a$;
(ii) F is mutual inverse in the sense of $F_a(b) = c \Leftrightarrow F_{a^{-1}}(c) = b$;
(iii) F is homomorphic in the sense of $F_a(bc) = F_a(b)F_a(c)$;
(iv) F is self-reflective in the sense of $F_a(b) = F_a^{-1}(b^{-1})$.

Proof. See Appendix A.

Combining all above properties together, we obtain a new property, named as *power law*, for CSP-based LD systems.

Proposition 2 (Power Law of CSP-based LD Systems). *Suppose that F is a CSP-based left self-distributive system defined over a noncommutative group G. Then, for arbitrary three integers m, s, t such that $m = s + t$, we have*

$$F_a(b^m) = F_a(b^s)F_a(b^t) = F_a^m(b) \quad \textit{and} \quad F_{a^m}(b) = F_{a^s}(F_{a^t}(b)). \tag{4}$$

Proof. It is easy to obtain by combining the property (iii) in Proposition 1, and the definition of the CSP-based LD system given by the formula (3).

4 New Cryptographic Assumptions over CSP-based Left Self-Distributive Systems

4.1 The CSP-DDH assumption

Recall that the decisional Diffie-Hellman (DDH) problem is developed from the discrete logarithm problems (DLP) over a cyclic group. Similarly, given the underlying noncommutative group G, for arbitrary $a, b \in G \setminus \{1_G\}$ (where 1_G is the identity of G), the CSP-based decisional Diffie-Hellman (CSP-DDH) problem is to distinguish the following two distributions:

$$D_{a,b} = \{(F_{a^i}(b),\ F_{a^j}(b),\ F_{a^{i+j}}(b)) \in G^3 : i, j \xleftarrow{\$} \mathbb{Z}\},$$

and

$$\widetilde{D}_{a,b} = \{(F_{a^i}(b),\ F_{a^j}(b),\ F_{a^k}(b)) \in G^3 : i, j, k \xleftarrow{\$} \mathbb{Z}\}.$$

Here and in sequel, the symbol "$\xleftarrow{\$} \mathbb{Z}$" always indicates a random integer sampling process. In practice, we should randomly pick integers from an interval that is large enough to resist exhaustive attacks. *The CSP-DDH assumption* says that the CSP-DDH problem over the given noncommutative group G is intractable.

Remark 3. At present, it is unclear if the CSP-DDH problem is actually hard. But at least in the so-called generic group model, we have no means to solve the CSP-DDH problem without solving the CSP problem in the corresponding non-commutative group G. To see this, we merely need to view the CSP problem and the CSP-DDH problem over a general noncommutative group G as the analogies of the DLP problem and the DDH problem over a general cyclic group with order q (where q is a large prime), respectively. According to [Mau05], with access to a DDH oracle, one can prove an $O(\sqrt[3]{q})$ bound for solving the DLP problem in the generic cyclic group. This suggests that from the perspective of complexity, the DLP problem and the DDH problem are polynomially equivalent in a generic cyclic group. Therefore, by an analogical manner, we have that

Conjecture 1. From the perspective of complexity, the CSP problem and the CSP-DDH problem in a generic noncommutative group are polynomially equivalent.

4.2 The Strong and Twin CSP-DH Assumptions

Enlightened by the work in [CKS08], we would like to give a reformulation on the above CSP-DDH problem and then propose the so-called strong and twin CSP-DH assumptions over the underlying noncommutative group G.

For arbitrary $a, b \in G \setminus \{1_G\}$ (where 1_G is the identity of G), let us define the CSP-DH function (w.r.t. (a, b)) $dh : G^2 \to G$ by

$$dh(X, Y) := Z, \text{ where } X = F_{a^s}(b),\ Y = F_{a^t}(b), \text{ and } Z = F_{a^{s+t}}(b) \tag{5}$$

for some unknown $s, t \xleftarrow{\$} \mathbb{Z}$. The problem of computing $dh(X, Y)$ given random $X, Y \in G$ such that $X = F_{a^s}(b)$, and $Y = F_{a^t}(b)$ for some unknown $s, t \xleftarrow{\$} \mathbb{Z}$ is the CSP-DH problem. The *CSP-DH assumption* asserts that this problem is hard. For $X, \hat{Y}, \hat{Z} \in G$ such that $X = F_{a^s}(b)$, $\hat{Y} = F_{a^{\hat{t}}}(b)$, and $\hat{Z} = F_{a^{\hat{u}}}(b)$ for some unknown $s, \hat{t}, \hat{u} \xleftarrow{\$} \mathbb{Z}$, define the CSP-DH predicate $dh : G^3 \to \{0, 1\}$ by

$$dhp(X, \hat{Y}, \hat{Z}) :=\ dh(X, \hat{Y}) \stackrel{?}{=} \hat{Z}. \tag{6}$$

Apparently, the CSP-DH predicate is an exact reformulation of the aforementioned CSP-DDH problem. Then, the *strong CSP-DH assumption* says that it is hard to compute $dh(X, Y)$, given $X, Y \in G$ such that $X = F_{a^s}(b)$ and $Y = F_{a^t}(b)$ for some unknown $s, t \xleftarrow{\$} \mathbb{Z}$, alone with access to a decision oracle for the CSP-DH predicate $dhp(X, \cdot, \cdot)$, which on input $(\hat{Y}, \hat{Z})$ such that $\hat{Y} = F_{a^{\hat{t}}}(b)$ and $\hat{Z} = F_{a^{\hat{u}}}(b)$ for some unknown $\hat{t}, \hat{u} \xleftarrow{\$} \mathbb{Z}$, and returns $dhp(X, \hat{Y}, \hat{Z})$.

Furthermore, for arbitrary $a, b \in G \setminus \{1_G\}$, let us define the twin CSP-DH function (w.r.t. (a, b)) $2dh : G^3 \to G^2$ by

$$2dh(X_1, X_2, Y) :=\ (dh(X_1, Y), dh(X_2, Y)), \tag{7}$$

where $X_1 = F_{a^s}(b)$, $X_2 = F_{a^t}(b)$, and $Y = F_{a^u}(b)$ for some unknown random integers $s, t, u \xleftarrow{\$} \mathbb{Z}$. Similarly, the twin CSP-DH predicate (w.r.t. (a, b)) is defined by

$$2dhp(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2) := \ 2dh(X_1, X_2, \hat{Y}) \stackrel{?}{=} (\hat{Z}_1, \hat{Z}_2), \tag{8}$$

where $X_1 = F_{a^s}(b)$, $X_2 = F_{a^t}(b)$, $Y = F_{a^{\hat{u}}}(b)$, $Z_1 = F_{a^{\hat{v}}}(b)$ and $Z = F_{a^{\hat{w}}}(b)$ for some unknown random integers $s, t, \hat{u}, \hat{v}, \hat{w} \xleftarrow{\$} \mathbb{Z}$. Then, the *twin CSP-DH assumption* states it is hard to compute $2dh(X_1, X_2, Y)$, given random $(X_1, X_2, Y) \in G$ such that $X_1 = F_{a^s}(b)$, $X_2 = F_{a^t}(b)$, and $Y = F_{a^u}(b)$ for some unknown $s, t, u \xleftarrow{\$} \mathbb{Z}$, while the *strong twin CSP-DH assumption* states that the twin CSP-DH assumption holds even with access to a decision oracle for the twin CSP-DH predicate $2dhp(X_1, X_2, \cdot, \cdot, \cdot)$, which on input $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$ such that $Y = F_{a^{\hat{u}}}(b)$, $Z_1 = F_{a^{\hat{v}}}(b)$ and $Z = F_{a^{\hat{w}}}(b)$ for some unknown random integers $\hat{u}, \hat{v}, \hat{w} \in \mathbb{Z}$, and returns $2dhp(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$. Analogically, enlightened by Theorem 1 in [CKS08], we have that

Conjecture 2. The CSP-DH assumption over the underlying noncommutative group G holds if and only if the strong twin CSP-DH assumption over G holds.

5 Cryptosystems from CSP-based Left Self-Distributive Systems

The CSP-DDH assumption over a noncommutative group G immediately implies a Diffie-Hellman-like key agreement protocol, which in turn implies an ElGamal-like encryption scheme over G. Furthermore, the strong twin CSP-DH assumption implies a Cramer-Shoup-like encryption scheme over the underlying noncommutative group G.

5.1 Constructions

Suppose that F is a hard CSP-based left self-distributive system defined over a noncommutative group G. Let $a, b \in G$ be two public elements. Assume that Alice and Bob want to negotiate a common session key. Then, Alice (resp. Bob) picks at random an integer s (resp. t) and then sends $F_{a^s}(b)$ (resp. $F_{a^t}(b)$) to Bob (resp. Alice). Finally, both of them can compute $F_{a^{s+t}}(b)$, by which a session key can be defined as

$$K_{session} = Kdf(F_{a^{s+t}}(b)), \tag{9}$$

where $Kdf(\cdot)$ is a key derivation function, such as KDF1 defined in IEEE Std 1363-2000. Note that we have not specified the magnitudes of s, t and the order of the group G. In fact, all of them should be large enough to resist exhaustive attacks. We will see that the order of the group G could be even infinite (cf. Section 5.4).

The Basic Scheme—CSP-ElG. The above interactive is a natural analogy of the Diffie-Hellman key agreement protocol [DH76]. Similarly, we can define the following encryption scheme (denoted by CSP-ElG) that is an analogy of the ElGamal cryptosystem [ElG85]. Our basic construction consists of the following four algorithms:

- **Setup.** The message space is $\mathcal{M} = G$, while the ciphertext space is $\mathcal{C} = G^2$. Picks $a, b \in G$ and publishes them as the system parameters.
- **KeyGen.** Picks an integer $s \in \mathbb{Z}$ at random. The public key is $pk = F_{a^s}(b)$, while the secret key is $sk = s$.
- **Enc.** Picks an integer $t \in \mathbb{Z}$ at random, the ciphertext on a message $m \in G$ is

$$c = (F_{a^t}(b),\ mF_{a^t}(pk)),$$

- **Dec.** $m = c_2 F_{a^s}^{-1}(c_1)$.

Under certain assumptions, the above scheme is IND-CPA secure (See Theorem 1 in Section 4). It is easy to derive a CCA secure encryption scheme by employing the Fujisaki-Okamoto transformation [FO99]. However, enlightened by the work in [CKS08,ABR01,CS03], we would like to give the following two different extensions: the first is called the hashed ElGamal variant that is IND-CCA secure in the random oracle model, while the second is called the Cramer-Shoup-like variant that is IND-CCA secure even in the standard model (cf. Section 4).

The First Extended Scheme—CSP-hElG. Our first extension is the hashed version of the above CSP-ElG scheme and thus denoted by CSP-hElG. The CSP-hElG scheme consists of the following four algorithms:

- **Setup.** Suppose that $\Pi = (E, D)$ is a symmetric cipher with the key space $\mathcal{K}$, and $H : G^2 \to \mathcal{K}$ be a hash function. Pick $a, b \in G$ randomly and publish them as the system parameters.
- **KeyGen.** The public key is X, with the corresponding secret key x, where $X = F_{a^x}(b)$.
- **Enc.** To encrypt a message $m \in \mathcal{M}$, one chooses a random $y \in \mathbb{Z}$, computes

$$Y := F_{a^y}(b),\ Z := F_{a^y}(X),\ k := H(Y, Z),\ c := E_k(m),$$

  and the ciphertext is (Y, c).
- **Dec.** Given the ciphertext (Y, c), and the secret key x, one computes

$$Z := F_{a^x}(Y),\ k := H(Y, Z),\ m := D_k(c).$$

The Second Extended Scheme—CSP-CS. Our second extension, denoted by CSP-CS, is an analogy of the well-known cryptosystem due to Cramer and Shoup [CS03]. The CSP-CS scheme consists of the following four algorithms:

– **Setup.** Suppose that $\Pi = (E, D)$ is a symmetric cipher with the key space G [6], and $T : G^2 \to \mathbb{Z}$ be a hash function. Pick $a, b \in G$ randomly and publish them as the system parameters.
– **KeyGen.** The public key is (X_1, X_2, X_3, X_4), with the corresponding secret key (x_1, x_2, x_3, x_4), where $X_i = F_{a^{x_i}}(b)$ for $i = 1, 2, 3, 4$.
– **Enc.** To encrypt a message $m \in \mathcal{M}$, one chooses a random $y \in \mathbb{Z}$, computes

$$Y := F_{a^y}(b),\ Z_1 := F_{a^y}(X_1),\ t := T(Y, Z_1),\ Z_2 := F_{a^y}(F_{a^t}(X_2)X_3),$$

$$k := F_{a^y}(X_4),\ c := E_k(m)$$

and the ciphertext is (Y, Z_1, Z_2, c).
– **Dec.** Given the ciphertext (Y, Z_1, Z_2, c), and the secret key (x_1, x_2, x_3, x_4), one computes $t := T(Y, Z_1)$ and tests if

$$F_{a^{x_1}}(Y) \stackrel{?}{=} Z_1 \text{ and } F_{a^{t+x_2}}(Y)F_{a^{x_3}}(Y) = Z_2.$$

If not, reject. Otherwise, compute $k := F_{a^{x_4}}(Y)$ and output $m := D_k(c)$.

5.2 Securities

Theorem 1 (IND-CPA of CSP-ElG). *Based on the CSP-DDH assumption, the ciphertexts of the encryption scheme CSP-ElG are indistinguishable under chosen plaintext attacks in the standard model.*

Proof. See Appendix B.

Theorem 2 (IND-CCA of CSP-hElG). *If H is modeled as a random oracle, and the underlying symmetric cipher Π is itself secure against chosen ciphertext attacks, then the hashed ElGamal encryption scheme CSP-hElG is secure against chosen ciphertext attacks under the strong CSP-DH assumption.*

Proof. Analogically implied by Theorem 1 and the claims in [CKS08].

Theorem 3 (IND-CCA of CSP-CS). *Suppose T is a target collision resistant hash function. Further, suppose the CSP-DDH assumption holds, and the symmetric cipher $\Pi = (E, D)$ is secure against chosen ciphertext attack. Then CSP-CS is secure against chosen ciphertext attack.*

Proof. Analogically implied by Theorem 13 in [CKS08].

[6] Similar to the reformulation in [CKS08], for simplicity we assume that the cipher's secret key consists of a random group element in G, but this assumption can be removed using standard techniques, c.f. [CS03].

5.3 Efficiency for Computing and Representing $F_{a^t}(b)$

For computing $F_{a^t}(b)$, we should at first compute a^t, and then plus one inversion and two multiplications. When t is large, say several hundreds of digits, rather than to multiply a t times, a similar successive doubling method should be employed, and thus a factor of $\log t$ would be introduced in the following performance evaluation. At present, it is enough to set t as an integer with 128 bits to resist exhaustive attacks.

It is necessary to assume that the basic group operations, i.e., multiplication of two elements and the inversion, can be finished efficiently. This assumption implies that the lengths of the representations of all elements in G, including a, b, a^t and $F_{a^t}(b)$, should be polynomial in the system security parameters, since the results have to be output bit-by-bit by using classical computers.

5.4 Potential Implementations and Evaluations

Now, let us proceed to give an implementation on our proposal by using braid groups.

Intractability of CSP in braid groups. Although some algorithms for solving CSP in braid groups were proposed [EH94,GM02,FGM03,Geb05], none of them has ever been proven in polynomial time (with respect to the braid index n). As far as we know, Gebhardt's algorithm [Geb05], which was proposed in 2003 but formally published in 2005, is the most efficient method for solving CSP in braid groups. The algorithm has not been proven to be polynomial-time, yet. After then, CSP in braid groups is classified for further study. According to Garber's report [Gar07], at present we can merely solve CSP for periodic braids within polynomial time. From ultimate solutions for CSP in braid groups, we are still facing two kinds of challenges: One is how to solve CSP for rigid braids within polynomial time, and the other is how to find a polynomial boundary for Gebhardt's method. (Please see [BGGM07a,BGGM07b,BGGM08] for more details.) According to [Deh04] and [KLT08], most of known attacks against braid-based cryptosystems take advantage of the way that the keys are generated, rather than solve CSP itself. So in ref. [KLT08], Ko et al. also proposed some new methods for generating hard CSP instances for the braid cryptography. According to Shpilrain's latest claim in [SU08], there is no deterministic polynomial-time algorithm for solving CSP in braid groups up to 2008. This gives us certain confidence for using braid groups as the platform to implement our proposal. Moreover, the capability of CSP assumption to resist currently known quantum attacks is also discussed from the perspective of hidden subgroup problems [WWC+10].

Parameter Suggestion. Note that the braid group B_n is infinite, and it is not convenient to work with an infinite group. So in practice, we always choose two positive

integers n and l as the system parameters, and assume that all the braids involved in our schemes are randomly chosen from the following finite subset [KCCL02]

$$B_n(l) = \{b \in B_n | \ell(b) \leq l\} \subseteq B_n, \tag{10}$$

where $\ell(b)$ is the canonical length of b. According to ref. [KCCL02], $|B_n(l)| \leq \left(\lfloor \frac{n-1}{2} \rfloor !\right)^l$. Further, if the keys are selected properly, say by employing Ko et al.'s method [KLT08], then according to Maffre's suggestion [Maf06], it is enough to set $n = 50$ and $l = 10$ to resist all known classical attacks.

Ciphertext Expansion. Apparently, in the CPA-secure scheme, CSP-ElG, the length of the ciphertexts is double of the length of the message to be encrypted. This means the expansion factor is 2, exactly the same situation of ElGamal cryptosystems. Regarding the extended CCA-secure schemes, i.e. CSP-hElG and CSP-CS, the ciphertext expansion factors are $(\kappa + \iota(G))/\kappa$ and $(\kappa + 3 \cdot \iota(G))/\kappa$ respectively, where κ is the block size of the underlying symmetric cipher and $\iota(G)$ indicates the length of the representation of an element in G. If G is finite, say Z_N, then in general we can assume $\iota(G) = \log |G|$. Now, new problems arise:

- First, when $\iota(G)$ is fixed, the expansion factors of these extensions could be strictly less than 2 for sufficiently large κ. This in turn suggests that these extensions might be more efficient in space than the basic CPA construction. What is the expense for achieving this? Our answer is: We achieve this at the expense of more computations.
- Second, when G is infinite, what is $\iota(G)$? In fact, we never need to represent infinite elements in practice. We just need to represent the involved elements. For example, if G is instantiated with the braid group B_n and the canonical length of all involved braids are bounded by l, then a braid in $B_n(l)$ (cf. the previous subsection) can be represented by a bit string of the length $\iota(B_n(l)) = ln \log n$ [KLCH00,CKL$^+$01].

Performance. Combining the above discussions together, let us give a performance evaluation on our proposal. Let us merely focus on braid groups, since for arbitrary group G, we have no method to evaluate the complexity of group operations, such as multiplication, inversion, etc.

According to refs. [CKL$^+$01] and [Maf06], the complexities of the braid operations such as multiplication, inversion, canonical form computation, etc., are bounded by $\mathcal{O}(l^2 n \log n)$ in the sense of bit operations, where n and l are the braid index and the canonical length of involved braids, respectively. Then, taking CSP-ElG for example and combining the issues discussed in Section 5.3, the complexity for encryption and decryption can be concluded in Table 1.

Now, if we neglect small constant factors and the cost for pre-computations, then the encryption and decryption of the scheme CSP-ElG can be finished with

Table 1. Computation Cost of CSP-ElG

	Comp. content	Comp. cost	Explanation
Enc	a^t	$\log t \cdot l^2 n \log n$	successive doubling
	a^{-t}	$l^2 n \log n$	1 inversion
	$c_1 = F_{a^t}(b)$	$2 \cdot l^2 n \log n$	2 multiplications
	$c_2 = mF_{a^t}(pk)$	$3 \cdot l^2 n \log n$	3 multiplications
Dec	a^s	$\log s \cdot l^2 n \log n$	successive doubling, pre-computing
	a^{-s}	$l^2 n \log n$	1 inversion, pre-computing
	$m = c_2 F^{-1}_{a^s}(c_1)$	$4 \cdot l^2 n \log n$	3 multiplications, 1 inversion

the complexities of $\mathcal{O}(\log t \cdot l^2 n \log n)$ and $\mathcal{O}(l^2 n \log n)$, respectively. When $\log t \leq 128 = 2^7$ (cf. Section 5.3) and Maffre's suggestion [Maf06] is adopted (i.e., $n = 50$ and $l = 10$), the number of bit operations for performing one time encryption is directly proportional to 2^{22}, while the number of bit operations for one time decryption is directly proportional to 2^{15}. We know that at present, the modulus of a secure RSA cryptosystem should be at least 1024 bits. Then, the number of bit operations for performing a modular exponential operation is directly proportional to 2^{30}. This suggests that our proposal is much efficient than RSA cryptosystem. Of course, every coin has two sides. The disadvantage of our newly derived braid-based encryption scheme is that the length of the public-key is considerably large—about $12K$ bits (where $K = 1024$) [WWC$^+$10].

6 Conclusion

Although CSP-based LD systems over any noncommutative groups are non-trivial, only those based on a noncommutative group with the intractability assumption of CSP are suitable for our proposal. It is worth mentioning that even if the CSP assumption in braid groups were broken in future, our method is still significant in building cryptosystems in other noncommutative groups provided that CSP in these groups is intractable. For example, given a ring R with identity, the general linear group $GL_n(R)$, i.e., the group of $n \times n$ invertible matrices with elements in R, can be taken into account. Especially, considering that multiplications over 0-1 matrices can be implemented much efficiently and these matrices have tightly relation with certain lattice problems, it seems very promising to implement our proposal in $GL_n(\mathbb{Z}_2)$.

References

[ABR01] M. Abdalla, M. Bellare, and P. Rogaway. The oracle diffie-hellman assumptions and an ananlysis of dhies. In *CT-RSA'01, LNCS 2020*, pages 143–158. Springer, 2001.

[AAG99] I. Anshel, M. Anshel, and D. Goldfeld. An algebraic method for public key cryptography. *Math. Research Letters*, (6):287–291, 1999.

[BGGM07a] J.S. Birman, V. Gebhardt, and J. González-Meneses. Conjugacy in garside groups i: Cyclings, powers, and rigidity. *Groups, Geometry and Dynamics*, 1(3):221–279, 2007.

[BGGM07b] J.S. Birman, V. Gebhardt, and J. González-Meneses. Conjugacy in garside groups iii: periodic braids. *J. Algebra*, 316(2):746–776, 2007.

[BGGM08] J.S. Birman, V. Gebhardt, and J. González-Meneses. Conjugacy in garside groups ii: Structure of the ultra summit set. *Groups, Geometry and Dynamics*, 2(1):16–31, 2008.

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The twin diffie-hellman problem and applications. In *Advances in Cryptology - EUROCRYPT 2008, LNCS 4965*, pages 127–145. Springer, 2008.

[CKL+01] J.C. Cha, K.H. Ko, S.J. Lee, J.W. Han, and J.H. Cheon et al. An efficient implementation of braid groups. In *ASIACRYPT 2001, LNCS 2248*, pages 144–156. Springer, 2001.

[CJ03] J.-H. Cheon and B. Jun. A polynomial time algorithm for the braid diffie-hellman conjugacy problem. In *CRYPTO 2003, LNCS 2729*, pages 212–225. Springer, 2003.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[Deh04] P. Dehornoy. Braid-based cryptography. *Contemp. Math., Amer. Math. Soc.*, 360:5–33, 2004.

[Deh06] P. Dehornoy. Using shifted conjugacy in braid-based cryptography. *Contemporary Mathematics*, 0(0):1–9, 2006. http://www.math.unicaen.fr/ dehornoy/papers.html.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(5):644–654, 1976.

[ElG85] ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEETIT: IEEE Transactions on Information Theory*, 31, 1985.

[EH94] E.A. Elrifai and H.R.Morton. Algorithms for positive braids. *Quart. J. Math. Oxford Ser.*, 45(2):479–497, 1994.

[FGM03] N. Franco and J. Gonzales-Menses. Conjugacy problem for braid groups and garside groups. *Journal of Algebra*, 266:112–132, 2003.

[FO99] E. Fujisaki and T. Okamoto. How to enhance the security of public key encryption at minimum cost. In *In Public Key Cryptography – PKC'99, LNCS 1560*, pages 53–68. Springer, 1999.

[Gar07] David Garber. Braid group cryptography. *Report, PRIMA school and conference ob braids, Singapore*, pages 1–75, 2007.

[Geb05] V. Gebhardt. A new approach to the conjugacy problem in garside groups. *Journal of Algebra*, 292:282–302, 2005.

[GM02] J. Gonzales-Meneses. Improving an algorithm to solve the multiple simultaneous conjugacy problems in braid groups. *Preprint, http://arxiv.org/abs/math.GT/0212150*, 2002.

[Kit95] A. Kitaev. Quantum measurements and the abelian stabilizer problem. *Preprint, quant-ph/9511026*, 1995.

[KLT08] K. Ko, J. Lee, and T. Thomas. Towards generating secure keys for braid. *Cryptography, Designs, Codes and Cryptography*, 2008.

[KCCL02] K.H. Ko, D.H. Choi, M.S. Cho, and J.W. Lee. New signature scheme using conjugacy problem. *Preprint, http://eprint.iacr.org/2002/168*, 2002.

[KLCH00] K.H. Ko, S.J. Lee, J.H. Cheon, and J.W. Han. New public-key cryptosystem using braid groups. In *CRYPTO 2000, LNCS 1880*, pages 166–183. Springer, 2000.

[Lee04] E. Lee. Braig groups in cryptography. *IEICE Trans. Fundamentals*, E87-A(5):986–992, May 2004.

[LU08] J. Longrigg and A. Ushakov. Cryptanalysis of shifted conjugacy authentication protocol. *Journal of Math. Cryptology*, (2):107–114, 2008.

[Maf06] Samuel Maffre. A weak key test for braid based cryptography. *Des. Codes. Crypt.*, 39:347–373, 2006.

[MST02] S.S. Magliveras, D.R. Stinson, and T. Trung. New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups. *Journal of Cryptography*, 15:285–297, 2002.

[Mau05] Ueli Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding 2005, LNCS 3796*, pages 1–12. Springer, 2005.

[PZ03] J. Proos and C. Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Information and Computation*, 3:317–344, 2003.

[Sho97] P. Shor. Polynomail-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 5:1484–1509, 1997.

[SU08] V. Shpilrain and A. Ushakov. An authentication scheme based on the twisted conjugacy problem. In *ACNS 2008, LNCS 5037*, pages 366–372. Springer, 2008.

[WWC+10] L.C. Wang, L.H. Wang, Z.F. Cao, Y.X. Yang, and X.X. Niu. Conjugate adjoining problem in braid groups and new design of braid-based signatures. *Sci China Ser F-Inf Sci*, (3):(In presss), 2010.

A Proof of Proposition 1

Proof. Suppose F is a CSP-based LD system defined over noncommutative group G. Then, for arbitrary $a, b, c \in G$, we have that

- Property (i): Idempotent. Since $aaa^{-1} = a$, i.e., a will remain unchanged when it conjugates to itself. By using the notation as in the formula (1), we have $F_a(a) = a$.
- Property (ii): Mutual inverse. According to the definition of F (cf. (3)), we have

$$\begin{aligned} F_a(b) = c &\Leftrightarrow c = aba^{-1} \\ &\Leftrightarrow a^{-1}ca = b \\ &\Leftrightarrow F_{a^{-1}}(c) = b. \end{aligned}$$

- Property (iii): Homomorphic.

$$\begin{aligned} F_a(bc) &= a(bc)a^{-1} \\ &= (aba^{-1})(aca^{-1}) \\ &= F_a(b)F_a(c). \end{aligned}$$

- Property (iv): Self-reflective.

$$\begin{aligned} F_a(b) &= aba^{-1} \\ &= (ab^{-1}a^{-1})^{-1} \\ &= (F_a(b^{-1}))^{-1} \\ &\triangleq F_a^{-1}(b^{-1}). \end{aligned}$$

This concludes the proposition. □

B Proof of Theorem 1

Proof. Assume that the CSP-DDH assumption holds for the underlying noncommutative group G. We will prove by contradiction that CSP-ElG is IND-CPA. Suppose that CSP-ElG is not IND-CPA, and let $\mathcal{A}$ be an algorithm which, on the system parameters $a, b \in G$ and a random public key $F_{a^s}(b)$, has probability non-negligibly greater than $1/2$ of distinguishing random encryptions $Enc(m_0)$ and $Enc(m_1)$ of two messages m_0, m_1 of its choice. Let $(F_{a^s}(b),\ F_{a^t}(b),\ F_{a^u}(b)) \in G^3$ be either a random CSP-DDH triple or a random triple, with equal probability. We will produce an algorithm $\mathcal{B}$ which can distinguish between the two cases, using $\mathcal{A}$ as an oracle, with probability close to 1. (This result is stronger than what we need to prove.) The algorithm $\mathcal{B}$ picks two random integers v, w and constructs the triple of group elements

$$T = (F_{a^{s+v}}(b),\ F_{a^{t+w}}(b),\ F_{a^{u+v+w}}(b)),$$

which is easy to do since $\mathcal{B}$ knows $F_{a^s}(b), F_{a^t}(b), F_{a^u}(b), a, b, v$ and w. The algorithm $\mathcal{B}$ then calls $\mathcal{A}$ with the public key $F_{a^{s+v}}(b)$, which is guaranteed to be random since v is chosen randomly. Afterwards, the algorithm $\mathcal{A}$ selects two messages $m_0, m_1 \in G$, and $\mathcal{B}$ replies with the ciphertext $c^*_\beta = (F_{a^{t+w}}(b), m_\beta \cdot F_{a^{u+v+w}}(b))$ for randomly picked $\beta \in \{0, 1\}$. There are now two cases to consider.

1. Suppose that $u = s + t$ holds. Then $u + v + w = (s + v) + (t + w)$, so T is a CSP-DDH triple. Moreover, all possible CSP-DDH triples w.r.t. (a, b) are equally likely to occur as T, since v and w are random. Therefore c^*_β is a valid random encryption of m_β (random since $F_{a^{t+w}}(b)$ is a random group element). Under these conditions, the algorithm $\mathcal{A}$ by hypothesis will succeed in outputting β with probability exceeding $1/2$ by a non-negligible quantity.
2. Suppose that u is random. Then T is a random triple of group elements, and all possible triples of group elements occur with equal probability. In this situation, the probability distribution of c^*_0 is *identical* to that of c^*_1, over all possible random choices of v and w. It follows that the algorithm $\mathcal{A}$ cannot exhibit different behavior for $\beta = 0$ and $\beta = 1$. Note that we can arrive at this conclusion even though the expression c^*_β is an invalid encryption of m_β —that is, even though we have no information about how $\mathcal{A}$ behaves on invalid inputs, we know for certain that $\mathcal{A}$ cannot behave differently depending on the value of β.

The above analysis reveals that if $(F_{a^s}(b),\ F_{a^t}(b),\ F_{a^u}(b))$ is a CSP-DDH triple then $\mathcal{A}$ with non-negligible probability exhibits different behavior depending on whether $\beta = 0$ or $\beta = 1$, whereas if $(F_{a^s}(b),\ F_{a^t}(b),\ F_{a^u}(b))$ is not a CSP-DDH triple then $\mathcal{A}$ must behave identically regardless of the value of β. Hence, by repeating this process with several different choices of random integers v, w, the algorithm $\mathcal{B}$ can determine with high probability whether or not $\mathcal{A}$ can determine the value of β. In this way $\mathcal{B}$ can determine whether or not $(F_{a^s}(b),\ F_{a^t}(b),\ F_{a^u}(b))$ is a CSP-DDH triple, thus violating the CSP-DDH assumption for G. □