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 certain assumptions, the ciphertexts of our basic construction are proven indistinguishable against chosen plaintext attacks (IND-CPA) in the standard model, and two extended schemes achieve the IND-CCA security in the random oracle model. Then, our proposal is instantiated with braid groups, and leads to a new braid-based encryption scheme that is directly based on the intractability assumption of CSP in braid groups. Furthermore, we quote an analysis to manifest that our newly derived braid-based cryptosystem has the potential to resist currently known quantum attacks.

Keywords: Self-distributive systems, conjugator searching problem, braid groups, resistance to quantum attacks

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. Owing to their pioneering work, braid groups gained attention in the field of modern cryptography. 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.

Beyond to give 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, but 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 ciphertexts of our basic scheme are proven indistinguishable against chosen plaintext attacks (IND-CPA). Then, two extended schemes that achieve 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 that is directly based on the intractability of CSP. At the time of this writing, no polynomial-time algorithm for solving CSP in braid groups has been reported yet [SU08] (cf. Section 4.4). Moreover, we quote some detailed analysis in [WWC$^+$10] to argue that the resulted braid-based encryption is secure against currently known quantum attacks (cf. Section 4.5).

In fact, our originality is rooted in 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 in at least 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; based on these newly developed properties, a Diffie-Hellman-like key agreement protocol, an ElGamal-like encryption scheme and its CCA-secure extensions are proposed in Section 4; meanwhile, the provable security theorem, the efficiency, and other related discussions are also presented. Concluding remarks are given in Section 5. Related proofs are arranged in appendixes.

2 Left Self-Distributive System and the One-Wayness 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.

If the LD system $F_{\cdot}(\cdot)$ defined as (1) possesses the following one-wayness property [Deh06], it is useful for designing cryptographic protocols: It is hard to retrieve 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 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 a one-way function. That is, it is hard to find s for given $(p, F_s(p) = sps^{-1})$, 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 noncommutative 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 should be very useful under the aforementioned one-wayness assumption.

Proposition 1. *Suppose F be 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 F be 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*

$$F_a(b^m) = F_a^m(b) \quad \textit{and} \quad F_{a^m}(b) = F_a(F_{a^{m-1}}(b)) = F_{a^{m-1}}(F_a(b)) = F_{a^s}(F_{a^t}(b)). \tag{4}$$

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 Cryptosystems from CSP-based Left Self-Distributive Systems

The power law of CSP-based LD system immediately implies a Diffie-Hellman-like key agreement protocol, which in turn implies an ElGamal-like encryption scheme. Of course, certain cryptographic assumptions and some computational issues should be taken into account.

4.1 Constructions

Suppose F be a one-way 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)—Please refer [GPV08] for exact meaning of sampling an integer from an infinite space—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

$$K_{session} = Kdf(F_{a^{s+t}}(b)), \tag{5}$$

where $Kdf(\cdot)$ is a key derivation function, such as KDF1 defined in IEEE Std 1363-2000. Note that we have not mentioned the order of the group G. In fact, it should be large enough to resist exhaustive attacks. We will see that it could be even infinite (cf. Section 4.4).

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

- **Setup.** The massage 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 4.2). It is easy to derive a CCA secure encryption scheme by employing the Fujisaki-Okamoto transformation [FO99]. However, enlightened by the work in [CKS08], we would like to give the following two different extensions: the first is called the hashed ElGamal variant, denoted by CSP-hElG, and the second is called the twin ElGamal variant, denoted by CSP-tElG. Both of them can be proven to be IND-CCA secure under certain assumptions (cf. Section 4.2).

The First Extended Scheme—CSP-hElG. Our first extension is enlightened by the work in [CKS08] and [ABR01]. It 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 $\Pi = (E, D)$ be a symmetric cipher with the key space $\mathcal{K}$, and $H : G^2 \to \mathcal{K}$ be a hash function. Picks $a, b \in G$ and publishes 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 CSP-hElG can be proven secure against chosen ciphertext attacks (See Theorem 2 in Section 4.2).

The Second Extended Scheme—CSP-tElG. Our second extension is also enlightened the work in [CKS08]. It is the twin version of the above CSP-ElG scheme and thus denoted by CSP-tElG. The CSP-tElG scheme consists of the following four algorithms:

- **Setup.** Suppose $\mathit{\Pi} = (E, D)$ be a symmetric cipher with the key space $\mathcal{K}$, and $H : G^3 \to \mathcal{K}$ be a hash function. Picks $a, b \in G$ and publishes them as the system parameters.
- **KeyGen.** The public key is (X_1, X_2), with the corresponding secret key (x_1, x_2), where $X_i = F_{a^{x_i}}(b)$ for $i = 1, 2$.
- **Enc.** To encrypt a message $m \in \mathcal{M}$, one chooses a random $y \in \mathbb{Z}$, computes

$$Y := F_{a^y}(b),\ Z_1 := F_{a^y}(X_1),\ Z_2 := F_{a^y}(X_2),\ k := H(Y, Z_1, Z_2),\ c := E_k(m),$$

  and the ciphertext is (Y, c).
- **Dec.** Given the ciphertext (Y, c), and the secret key (x_1, x_2), one computes

$$Z_1 := F_{a^{x_1}}(Y),\ Z_2 := F_{a^{x_2}}(Y),\ k := H(Y, Z_1, Z_2),\ m := D_k(c).$$

The CSP-tElG can be proven secure against chosen ciphertext attacks (See Theorem 4 in Section 4.2).

The Third Extended Scheme—CSP-rElG. Our third extension is denoted by CSP-rElG since it is obtained by combining the above CSP-ElG scheme and a randomly padding diagram. Compared with the CSP-ElG scheme, we just introduce a hash function H in CSP-rElG. The CSP-rElG scheme consists of the following four algorithms:

- **Setup.** The message space is $\mathcal{M} = \{0, 1\}^{\kappa}$, while the ciphertext space is $\mathcal{C} = G \times \{0, 1\}^{\kappa+\kappa_1}$, where κ are κ_1 are two positive integers that are discussed later. Suppose $H : G \to \{0, 1\}^{\kappa+\kappa_1}$ be a hash function. Picks $a, b \in G$ and publishes them as the system parameters.
- **KeyGen.** Picks an integer $s \in Z$ at random. The public key is $pk = F_{a^s}(b)$, while the secret key is $sk = s$.
- **Enc.** Picks $r \in \{0, 1\}^{\kappa_1}$ at random, then the ciphertext on a message $m \in \{0, 1\}^{\kappa}$ is

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

  where $t = (m||r)_2$ is the integer obtained by concatenating the bit-representations of m and r.

- **Dec.** Computes $(m'||r')_2 = c_2 \oplus H(F_{a^s}(c_1))$, then outputs m' if $c_1 = F_{a^{(m'||r')_2}}(b)$ and $\perp$ otherwise.

Although we cannot prove that CSP-rElG is IND-CCA secure with the assumptions developed in this paper (even if H is modeled as the random oracle model), the scheme has certain advantages, say intuitive construction, small expansion of the ciphertexts, without symmetric ciphers involving, etc. Anyway, we have not conceived a successful attack towards CSP-rElG, considering that there is no easy way to modify a ciphertext without being detected. Note that Shoup described another hashed ElGamal encryption (denoted by Shoup-hElG) that is merely IND-CPA secure in [Sho04]. The difference between the Shoup-hElG and the CSP-rElG scheme is that in CSP-rElG, we introduce the validation on the ciphertexts after decryption. However, we fail to answer whether this validation mechanism is enough for assuring IND-CCA security.

4.2 Security Requirements and Cryptographic Assumptions

Similar to the decisional Diffie-Hellman (DDH) assumption for the ElGamal cryptosystem, the security of the encryption scheme CSP-ElG described in the above subsection is based on the following assumption (denoted by CSP-DDH) over the underlying noncommutative group G: For arbitrary $a, b \in G \setminus \{1_G\}$, it is hard to distinguish the 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

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

where 1_G is the identity of G, and the symbol "$\xleftarrow{\$} \mathbb{Z}$" indicates the integer sampling algorithm **SampleD** defined in [GPV08].

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.

However, in order to prove the CCA security of the extended schemes descried in the above subsection, we need further assumptions. Enlightened by the work in [CKS08], for arbitrary $a, b \in G \setminus \{1_G\}$, let us define the CSP-DH function (w.r.t. (a, b)) $dh : G^2 \to G$ by

$$dh(X, Y) := Z, \text{ where } X = F_{a^s}(b),\ Y = F_{a^t}(b), \text{ and } Z = F_{a^{s+t}}(b). \tag{6}$$

Here, $s, t \in \mathbb{Z}$ are random integers. 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 $s, t \in \mathbb{Z}$ 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 $s, \hat{t}, \hat{u} \in \mathbb{Z}$, define the CSP-DH predicate

$$dhp(X, \hat{Y}, \hat{Z}) :=\ dh(X, \hat{Y}) \stackrel{?}{=} \hat{Z}. \tag{7}$$

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 $s, t \in \mathbb{Z}$, 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 $\hat{t}, \hat{u} \in \mathbb{Z}$, and returns $dhp(X, \hat{Y}, \hat{Z})$.

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].

Further, we can develop the CSP-based twin Diffie-Hellman assumption over the underlying noncommutative group G. 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

$$2dh(X_1, X_2, Y) := \; (dh(X_1, Y), dh(X_2, Y)), \tag{8}$$

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

$$2dhp(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2) := \; 2dh(X_1, X_2, \hat{Y}) \stackrel{?}{=} (\hat{Z}_1, \hat{Z}_2), \tag{9}$$

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 $s, t, \hat{u}, \hat{v}, \hat{w} \in \mathbb{Z}$. 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 $s, t, u \in \mathbb{Z}$, 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, we have

Theorem 3 ([CKS08]). *The CSP-DH assumption over the underlying noncommutative group G holds if and only if the strong twin CSP-DH assumption over G holds.*

Theorem 4 (IND-CCA of CSP-tElG). *If H is modeled as a random oracle, and the underlying symmetric cipher Π is itself secure against chosen ciphertext attacks, then the twin ElGamal encryption scheme CSP-hElG is secure against chosen ciphertext attacks under the CSP-DH assumption.*

Proof. Analogically implied by Theorem 1, Theorem 3, and Theorem 4 in [CKS08].

4.3 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. Then, 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.

4.4 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 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 give us certain confidence for using braid groups as the platform to implement our proposal. Especially in Section 4.5, we will quote an detailed analysis on the capability of our proposal in resiting currently known quantum attacks.

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]

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

where $\ell(b)$ is the canonical length of b. According to ref. [KCCL02], $|B_n(l)| \le \left(\lfloor \frac{n-1}{2} \rfloor !\right)^l$. 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. As regards the extended CCA-secure schemes, i.e. CSP-hElG and CSP-tElG, the ciphertext expansion factor is $(\kappa + \iota(G))/\kappa$, 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. As regards the randomly padding extension, i.e. CSP-rElG, the ciphertext expansion factor is $(\kappa + \kappa_1 + \iota(G))/\kappa$. 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. As for CSP-hElG and CSP-tElG, when κ increases, the cost of computing $E_k(m)$ and $D_k(c)$ increases. As regards CSP-rElG, when κ increases, $t = (m||r)_2$ also increases, leading to the increment of the cost for computing a^t; Moreover, when κ increase *linearly*, the value of t increases *exponentially* and in turn $\log t$ increase *linearly*, resulting in a *linear* increment of the computation cost.
- Second, when G is infinite, what is $\iota(G)$? In fact, we need never 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 4.3, the complexity for encryption and decryption can be concluded in Table 1.

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
	$m = c_2 F_{a^s}^{-1}(c_1)$	$4 \cdot l^2 n \log n$	3 multiplications, 1 inversion

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 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 4.3) and Maffre's suggestion [Maf06] is adopted (i.e., $n = 50$ and $l = 10$), the encryption can be implemented in $\mathcal{O}(2^{22})$ bit operations, while the decryption merely needs $\mathcal{O}(2^{15})$ bit operations. We know that at present, the modulus of a secure RSA cryptosystem should be at least 1024 bits. Then, the complexity of modular exponential operation is about $\mathcal{O}(2^{30})$. 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].

4.5 Can we efficiently solve CSP in braid groups by using quantum computers?

The content in this subsection was partially contained in ref. [WWC$^+$10]. However, the literature ***[WWC$^+$10] has not been published****, yet. Thus, we would like to restate the related materials to support the corresponding claims in this paper.*

The known quantum algorithms are roughly divided into two types, namely hidden subgroup algorithms and amplitude amplification algorithms [Röt06]. The latter can merely obtain polynomial speed-up ratios as compared to classical algorithms, while the former deals with the so-called hidden subgroup problem (HSP) and has the promise to obtain exponential speed-up ratios [Röt06]. In the HSP, one is given a black-box function $f : G \to S$ from a group G to a set S with the promise that there exists a subgroup $H \subseteq G$ such that f is constant on the cosets of H and takes distinct values for different cosets. The task is to determine H, or equivalently, to find a set of generators of H by making the minimum possible queries to f [Röt06]. HSP provides a unified framework to study problems of group theoretical nature. We have already obtained efficient solutions of the HSP for any abelian group [Kit95,Sho97,KNP07] and for some non-abelian groups [FIM$^+$03,BCvD05b,BCvD05a,Bac06,KR08]. Thus, a natural question arises: Can we efficiently solve the conjugator searching problem (CSP) by using quantum computers? This is partially equivalent to answering all of the following three sub-problems:

(1) Can we model CSP by using the framework of HSP?
(2) Can we solve the HSP obtained in (1) quantumly?
(3) Can we implement the algorithm obtained in (2) efficiently in terms of elementary quantum gates?

In spite of conducting an extensive investigation on the available literatures about HSP, including surveys [Joz01,LJK02,Lom04,Röt06] and some newly proposed but not formally published results such as those mentioned in refs. [MRV07], [KR08], and [DMR09], we could not find any answers for (1), (2) or (3). Here, we present three speculations that may partially provide negative answers for the aforementioned subproblems:

- First, we have no evident solution for (1) at present. Especially, the solution (space) for a given CSP instance is not a subgroup of B_n.

- Second, supposing (1) is solved, we still have no easy solution for (2) at present. The standard approach of a quantum algorithm for HSP is to use the oracle f to create coset states $\rho_H = \frac{1}{|G|}\sum_{g\in G}|gH\rangle\langle gH|$, where $|gH\rangle = \frac{1}{\sqrt{|H|}}\sum_{h\in H}|gh\rangle$ [DMR09]. This means that $|G|$ and $|H|$ should be finite, and thus the standard approach cannot be directly used for braid group B_n, considering $|B_n| = \infty$. Although Shor [Sho97] worked with a finite additive cyclic group $\mathbb{Z}_Q$ with $N^2 \leq Q < 2N^2$ for factoring N, at present we do not know how to translate Shor's method to braid groups, taking into account the fact that we do not know how to define a modular operation for braids.
- Third, supposing (2) is solved, we might face new challenges in solving (3). The (quantum computational) progress on HSP in the symmetric group S_n is rather limited and so far dominated by negative results [Röt06]. It has been shown [GSVV04] that exponentially many repetitions of quantum Fourier sampling are necessary for obtaining a non-negligible probability of distinguishing between the cases $|H| = 2$ and $|H| = 1$ (where $H \subseteq S_n$). Recently, Moore et al. [MRV07,DMR09] speculated that HSP in the groups that contain the symmetric group S_n as a subgroup may be resistant to all known quantum techniques. Although S_n is not a subgroup of the braid group B_n, there is a one-to-one correspondence between the set of the permutation braids in B_n and the set of the permutations in S_n. This suggests that there might exist some HSP instances in braid groups to be intractable even by using all known quantum techniques.

In summary, the currently known quantum technique cannot break the one-wayness of the left self-distributive system based on the intractability assumption of CSP in braid groups.

We claim again that the analysis in this subsection is not *our contribution. But we need to quote it in this paper for supporting the claimed advantages of our proposal.*

5 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 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.

[Bac06] Dave Bacon. How a clebsch-gordan transform helps to solve the heisenberg hidden subgroup problem. *Preprint, quant-ph/0612107*, September 20 2006.

[BCvD05a] Dave Bacon, Andrew M. Childs, and Wim van Dam. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Proc. 46th IEEE Symposium on Foundations of Computer Science (FOCS 2005), pp. 469-478; doi:10.1109/SFCS.2005.38*, pages 469–478, April 26 2005.

[BCvD05b] Dave Bacon, Andrew M. Childs, and Wim van Dam. Optimal measurements for the dihedral hidden subgroup problem. *Journal of Theoretical Computer Science (2006), no. 2*, (2), April 26 2005.

[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.

[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. *In: L. Gerritzen, D. Goldfeld, M. Kreuzer, G. Rosenberger and V. Shpilrain (Eds.), Algebraic Methods in Cryptography, Contemporary Mathematics*, 418:65–73, AMS (2006).

[DMR09] Aaron Denney, Cristopher Moore, and Alexander Russell. Finding conjugate stabilizer subgroups of PSL(2; q). *Preprint, math/0809.2445*, September 30 2009.

[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.

[FIM$^+$03] Katalin Friedl, G Abor Ivanyos, Fr Eric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and orbit coset. March 22 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.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 187–196. ACM, 2008.

[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.

[GSVV04] Michelangelo Grigni, Leonard J. Schulman, Monica Vazirani, and Umesh V. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. *Combinatorica*, 24(1):137–154, 2004.

[Jao08] David Jao. Public-key cryptography. *http://djao.math.uwaterloo.ca/w/CO485/685*, 2008.

[Joz01] R. Jozsa. Quantum factoring, discrete logarithms, and the hidden subgroup problem. *Computing in Science and Engineering (CSE)*, 3(2):34–43, March-April 2001.

[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.

[KNP07] Pascal Koiran, Vincent Nesme, and Natacha Portier. The quantum query complexity of the abelian hidden subgroup problem. *Theor. Comput. Sci*, 380(1-2):115–126, 2007.

[KR08] Hari Krovi and Martin Roetteler. An efficient quantum algorithm for the hidden subgroup problem over weyl-heisenberg groups, October 20 2008.

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

[LJK02] Samuel J. Lomonaco, Jr., and Louis H. Kauffman. Quantum hidden subgroup problems: A mathematical perspective, June 17 2002.

[Lom04] Chris Lomont. The hidden subgroup problem - review and open problems, November 04 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.

[MRV07] Cristopher Moore, Alexander Russell, and Umesh Vazirani. A classical one-way function to confound quantum adversaries, January 19 2007.

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

[Röt06] Martin Rötteler. Quantum algorithms: A survey of some recent results. *Informatik - Forschung und Entwicklung 21(1), 3-20. (2006)*, 2006.

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

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs, 2004.

[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] Licheng Wang, Lihua Wang, Zhenfu Cao, Yixian Yang, and Xinxin Niu. Conjugate adjoining problem over braid groups and new design of braid-based signatures. *Sci China Ser F-Inf Sci*, (to appear) 2010.

A Proof of Proposition 1

Proof. Suppose F be 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. The proof for this theorem can be easily sketched in imitation of Jao's[Jao08] proof for the IND-CPA security of ElGamal scheme. Of course, special cautiousness should be taken into consideration since we now work with a noncommutative group G. 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 $u = s + t$. 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. □