New Public Key Cryptosystems Using Polynomials over Non-commutative Rings

Zhenfu Cao, Xiaolei Dong and Licheng Wang

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, P. R. China
zfcao@cs.sjtu.edu.cn

Abstract. In this paper, we propose a new method for designing public key cryptosystems based on general non-commutative rings. The key idea of our proposal is that for a given non-commutative ring, we can define polynomials and take them as the underlying work structure. By doing so, it is easy to implement Diffie-Helman-like key exchange protocol. And consequently, ElGamal-like cryptosystems can be derived immediately. Moreover, we show hot to extend our method to non-commutative groups (or semi-groups).

1 Introduction

1.1 Background of Public-Key Cryptography and Proposals Based on Commutative Rings

Since the idea of public key cryptography (PKC) was introduced by Diffie and Hellman [20] in 1976, many PKC schemes have been proposed and broken. The trapdoor one-way functions play the key roles in the idea of PKC. Today, most succusseful PKC schemes are based on the perceived difficulty of certain problems in particular large finite *commutative* rings. For example, the difficulty of solving the integer factoring problem (IFP) defined over the ring Z_n (where n is the product of two large primes) forms the ground of the basic RSA cryptosystem [54] and its variants, such as Rabin-Williams [52,65,66] schemes, LUC's scheme[60], Cao's schemes [13,15] and elliptic curve version of RSA like KMOV [37]. The extended multi-dimension RSA cryptosystem [14], which can efficiently resist low exponent attacks, is also defined over the commutative ring $\mathbb{Z}_N[x]$. Another good case is that ElGamal-type PKC family, including the basic ElGamal scheme [22], elliptic curve cryptosystem, DSS and McCurley scheme [45], is based on the difficulty of solving the discrete logarithm problem (DLP) defined over a finite field Z_p (where p is a large prime), of course a commutative ring.

1.2 PKC Proposals Based on Generic Group Theory

The theoretical foundations for the above cyrptosystems lie in the intractability of problems closer to number theory than group theory [42]. On quantum computer, IFP and DLP, as well as DLP over elliptic curves (ECDLP), turned out to be efficiently solved by algorithms due to Shor [56], Kitaev [34] and Proos-Zalka [51]. Although practical quantum computers are as least 10 years away, their potential weakness will soon create distrust in current cryptographic methods [38].

As addressed in [38], in order to enrich cryptography as well as not to put all eggs in one basket, there have been many attempts to develop alternative PKC based on different kinds of problems [38]:

- In 1984, Wagner et al. [64] proposed an approach to design public-key cryptosystems based on the undecidable word problem for groups and semigroups. In 2005, Birget et al. [7] pointed out that Wagner's idea is actually not based on word problem, but on another, generally easier, premise problem. And finally, Birget et al. proposed a new public-key cryptosystem which is based on finitely presented groups with hard word problem.
- In 1999, Anshel et al. [2] proposed a compact algebraic key establishment protocol. The foundation of their method lies in the difficulty of solving equations over algebraic structures, in particular non-commutative groups [2]. In their pioneering paper, they also suggested that braid groups maybe are good alternative platforms for PKC.
- Subsequently, Ko et al. [36] firstly proposed new PKC by using braid groups in 2000. The security foundation is that the conjugator search problem (CSP) is intractable when the system parameters, such as braid index and the canonical length of the working braids, are selected properly. After that, the subject has met with a quick success [1,3,16,35,40,61,62]. However, from 2001 to 2003, repeated cryptanalytic success [17,31,32,33,39,46] also diminished the initial optimism on the subject significantly [10]. Some authors even announced the premature death of the braid-based PKC [19]. Dehornoy's paper [19] gives a good survey on the state of the subject, and evidently significant research is still needed to reach a definite conclusion on cryptographic potential of braid groups [10].
- In 2001, Paeng et al. [49] also published a new PKC built on finite non-abelian groups. Their method is based on the DLP in the inner automorphism group defined via the conjugate action. Their systems was later improved to the so-called MOR systems [50].
- Meanwhile, Magliveras et al. [41] developed new approaches to design public key cryptosystems using one-way functions and trapdoors in finite groups. It is worth remarking that their method originates in group theory. Two public key cryptosystems based on the difficulty of computing certain factorizations in finite groups, have been introduced: MST1 and MST2. Subsequently, in 2002, Vasco et al. [63] demonstrated that, after a suitable generalization, the factorization concepts used in MST1 and MST2 allow a uniform description of several cryptographic primitives. Also, it turned out that a generalization of MST2 can serve as a unifying framework for several proposed public key cryptosystems, including the ElGamal public key system, the braid group based system [36] and the MOR cryptosystem [50].
- In 2002, certain homomorphic cryptosystems were constructed for the first time for non-abelian groups due to Grigoriev and Ponomarenko [29]. Shortly afterwards, Grigoriev and Ponomarenko [30] extended their method to arbitrary nonidentity finite groups based on the difficulty of the membership problem for groups of integer matrices.
- Enlightened by the idea in the arithmetic key exchange [2], in 2004, Eick and Kahrobaei [21] proposed a new cryptosystem based on polycyclic groups. Polycyclic groups are a natural generalization of cyclic groups, but they are much more complex in their structure than cyclic groups. Hence their algorithmic theory is more difficult and thus it seems promising to investigate classes of polycyclic groups as candidates to have a more substantial platform perhaps more secure.
- In 2005, Shpilrain and Ushakov [57] suggested that R. Thompson's group maybe is a good platform for constructing public-key cryptosystems. In their contribution, the key assumption is the intractability of the decomposition problem, which is more general than the conjugator search problem, defined over R. Thompson's group, also a infinite non-abelian group given by finite presentation.

Among the above cryptosystems, those based on generic algebraic systems, especially *non-commutative* ones, attract more and more attentions. So far, most cryptosystems using non-commutative algebraic systems are related to the difficulty of solving CSP over certain non-abelian groups. Although there are algorithms for solving some *variants* of CSP in certain groups, such as braid groups [8,9,23,26,28], none of them can solve CSP *itself* defined over general non-abelian group in polynomial time with respect to the system parameters. However, non-commutative is a double-edged sword: on the one hand, it makes CSP meaningful; on the other hand, it brings some inconvenience for designing PKC schemes, for example, in Diffie-Hellman-like key agreement protocol, we require that the operations executed by both of the participants are symmetrical and commutable. How to utilize non-commutative and overcome its inconvenience is the key problem for developing PKC over non-commutative algebraic systems.

1.3 Motivations and Organization

In this paper, we would like to propose a new method for designing public key cryptosystems based on general non-commutative rings. The key idea of our proposal is that for a given non-commutative ring, we can define polynomials and take them as the underlying work structure. By doing so, it is much easy to implement the Diffie-Helman-like key exchange protocol and consequently ElGamal-like cryptosystems can also be derived immediately. In addition, in [58], Shpilrain et al. gave *six criteria* for choosing alternative non-commutative group G as PKC platforms. The firth criteria is: It is easy to produce some pairs $(a, \{a_1, \cdots, a_k\})$ such that $aa_i = a_i a$,$(i = 1, \cdots, k)$. We find that our proposal indeed provides a general way to produce pairs that meet Shpilrain's firth criteria for general non-commutative group G.

The rest of the paper is organized as follows. In Section 2, preliminaries on security models are introduced; In Section 3, we survey necessary cryptographic assumptions over non-commutative groups and then develop some new assumptions; In Section 4, we develop our method step by step: At first, we define polynomial over an arbitrary non-commutative ring and prove necessary propositions that supports our later design; Then, we describe a Diffie-Hellman-like key agreement protocol and two ElGamal cryptosystems based on new underlying structure and new developed assumptions; In Section 5, we extend our method to non-commutative groups and non-commutative semi-groups. Meanwhile, concrete examples are provided to support our method in practice. Finally, concluding remarks are made in Section 6.

2 Preliminaries

2.1 Notations

Throughout this paper, if x is a string then $|x|$ denotes its length, and if S is a set then $|S|$ denotes its size. We denote by $\mathbb{N}$ the set of positive integers, the integer $k \in \mathbb{N}$ denotes the security parameter. We say a function $\epsilon(k) : \mathbb{N} \mapsto [0, 1]$ is *negligible* if for all $\alpha > 0$, $\epsilon(k) < 1/k^\alpha$ for all sufficiently large k [43]. Assume that $\mathcal{A}$ is a probabilistic algorithm that runs in polynomial time with respect to the security parameter k. Then we denote $z \leftarrow \mathcal{A}(x, y, \cdots)$ the operation of running $\mathcal{A}$ with inputs $x, y, \cdots$ and letting z be the output, $z \leftarrow \mathcal{A}(x, y, \cdots, \mathcal{O}_1, \mathcal{O}_2, \cdots,)$ the operation of running $\mathcal{A}$ with inputs $x, y, \cdots$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \cdots$ and letting z be the output.

2.2 Public Key Encryption

In this subsection, we recall the formal definition for public key encryption schemes, together with the security notions.

Definition 1. *A public key encryption scheme $\varPi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ consists of the following three polynomial-time (in k) algorithms:*

- *The key generation algorithm – KGen: On input 1^k (unary representation of k), the algorithm KGen produces a pair (pk, sk) of matching public and private keys. Algorithm KGen is probabilistic.*
- *The encryption algorithm – Enc: Given a message m and a public key pk, Enc produces a ciphertext $c = \varPi(m)$ of m. This algorithm may be probabilistic.*
- *The decryption algorithm – Dec: Given a ciphertext $c = \varPi(m)$ and the private key sk. $Dec(sk, c)$ gives back the plaintext m. This algorithm is necessarily deterministic.*

In addition, for every pair (pk, sk) generated by $\mathbf{KGen}(1^k)$, and for every α, algorithms Enc and Dec satisfy

$$\Pr[\mathbf{Dec}(sk, \mathbf{Enc}(pk, m)) = m] = 1$$

where the probability is taken over the internal coin tosses of algorithm Enc and Dec.

Adversarial goals. The basic security notion required from a public key encryption scheme is the one-wayness (OW), which roughly means that one can't recover the whole plaintext from a given ciphertext.

Definition 2 (One-Wayness). *A public key encryption scheme $\varPi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ is said to be one-way if for all probabilistic polynomial time algorithms $\mathcal{A}$, for every $\alpha > 0$ and sufficiently large k,*

$$\Pr[\mathcal{A}(pk, c) = \mathbf{Dec}(sk, c) = m] < \frac{1}{k^\alpha}$$

where $c = \varPi(m) \leftarrow \mathbf{Enc}(pk, m)$, $(pk, sk) \leftarrow \mathbf{KGen}(1^k)$ and m is any message in message space.

A stronger security notion for a public key encryption scheme is the so-called semantic security (a.k.a. indistinguishability (IND) of encryption) [27]. This security notion requires computational impossibility to distinguish between two messages chosen by an adversary, which one has been encrypted, with a probability significantly better than 1/2.

Definition 3 (Semantic Security). *A public key encryption scheme $\varPi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ is said to be semantic security if for all probabilistic polynomial time algorithms $\mathcal{A}$, for every $\alpha > 0$ and sufficiently large k,*

$$\Pr[\mathcal{A}(pk, m_0, m_1, c) = m] < \frac{1}{2} + \frac{1}{k^\alpha}$$

where (m_0, m_1) is chosen by $\mathcal{A}$, $m \leftarrow \{m_0, m_1\}$, $c = \varPi(m) \leftarrow \mathbf{Enc}(pk, m)$, $(pk, sk) \leftarrow$ $\mathbf{KGen}(1^k)$.

Adversarial models. Currently, there are several types of attacks models for public key encryption, namely the chosen-plaintext attack (CPA), non-adaptive chosen-ciphertext attacks (CCA1) [47] and adaptive chosen-ciphertext attacks (CCA2) [53]. In a CPA, an adversary can access an encryption oracle. This scenario clearly cannot be avoided. In a CCA1, an adversary also can access a decryption oracle before being given the challenge ciphertext. While in a CCA2, an adversary can access a decryption oracle before and after being challenged; and the only restriction for him is that he cannot feed the oracle with the challenge ciphertext himself. This is the strongest known attack scenario.

Security levels are usually defined by pairing each goal (OW, IND) with an attack model (CPA, CCA1 or CCA2); *i.e.*, OW-CPA, OW-CCA1, OW-CCA2; IND-CPA, IND-CCA1 and IND-CCA2. Among each security level, the following relations are satisfied.

$$\begin{tikzcd}
\mathsf{OW\text{-}CPA} & \mathsf{OW\text{-}CCA1} \arrow[l] & \mathsf{OW\text{-}CCA2} \arrow[l] \\
\mathsf{IND\text{-}CPA} \arrow[u] & \mathsf{IND\text{-}CCA1} \arrow[l] \arrow[u] & \mathsf{IND\text{-}CCA2} \arrow[l] \arrow[u]
\end{tikzcd}$$

Definition 4 (OW-ATK). *Let $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ be a public key encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be any probabilistic polynomial time algorithm. For ATK $\in$ {CPA, CCA1, CCA2}, under sufficiently large k, let define*

$$\mathbf{Succ}^{\mathsf{OW\text{-}ATK}}_{\mathcal{A},\Pi} := \Pr\left[\begin{array}{c}(pk, sk) \leftarrow \mathbf{KGen}(1^k);\\ s \leftarrow \mathcal{A}_1(pk, \mathcal{O}_1)\\ c = \mathbf{Enc}(pk, m):\\ \mathcal{A}_2(s, c, \mathcal{O}_2) = m\end{array}\right]$$

where s is $\mathcal{A}$'s inner statement information, $\mathcal{O}_1, \mathcal{O}_2$ are oracles that $\mathcal{A}$ can access. According to each attack, $\mathcal{O}_1, \mathcal{O}_2$ are defined as follows:

- *If ATK = CPA then $\mathcal{O}_1(.) = \varepsilon$ and $\mathcal{O}_2(.) = \varepsilon$;*
- *If ATK = CCA1 then $\mathcal{O}_1(.) = \mathcal{D}_{sk}(.)$ and $\mathcal{O}_2(.) = \varepsilon$;*
- *If ATK = CCA2 then $\mathcal{O}_1(.) = \mathcal{D}_{sk}(.)$ and $\mathcal{O}_2(.) = \mathcal{D}_{sk}(.)$.*

Here a limitation is that $\mathcal{A}_2$ is not allowed to make access to decryption oracle with the challenge c itself as a query. We say that Π is (t, q_D, ϵ)-secure if for every adversary $\mathcal{A}$ that runs at most in time t, achieving $\mathbf{Succ}^{\mathsf{OW\text{-}ATK}}_{\mathcal{A},\Pi}(k) < \epsilon$, where q_D is the query times on decryption oracle $\mathcal{D}_{sk}(.)$.

Definition 5 (IND-ATK). *Let $\Pi = (\mathbf{KGen}, \mathbf{Enc}, \mathbf{Dec})$ be a public key encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be any probabilistic polynomial time algorithm. For ATK $\in$ {CPA, CCA1, CCA2}, under sufficiently large k, let define*

$$\mathbf{Adv}^{\mathsf{IND\text{-}ATK}}_{\mathcal{A},\Pi} := 2 \times \Pr\left[\begin{array}{c}(pk, sk) \leftarrow \mathbf{KGen}(1^k);\\ (m_0, m_1, s) \rightarrow \mathcal{A}_1(pk, \mathcal{O}_1)\\ b \xleftarrow{R} \{0, 1\}; c = \mathbf{Enc}(pk, m_b):\\ \mathcal{A}_2(m_0, m_1, s, c, \mathcal{O}_1) = b\end{array}\right] - 1$$

where s is $\mathcal{A}$'s inner statement information, $\mathcal{O}_1, \mathcal{O}_2$ are oracles that $\mathcal{A}$ can access. According to each attack, $\mathcal{O}_1, \mathcal{O}_2$ are defined as follows:

- *If* $\mathsf{ATK} = \mathsf{CPA}$ *then* $\mathcal{O}_1(.) = \varepsilon$ *and* $\mathcal{O}_2(.) = \varepsilon$*;*
- *If* $\mathsf{ATK} = \mathsf{CCA1}$ *then* $\mathcal{O}_1(.) = \mathcal{D}_{sk}(.)$ *and* $\mathcal{O}_2(.) = \varepsilon$*;*
- *If* $\mathsf{ATK} = \mathsf{CCA2}$ *then* $\mathcal{O}_1(.) = \mathcal{D}_{sk}(.)$ *and* $\mathcal{O}_2(.) = \mathcal{D}_{sk}(.)$*.*

Here a limitation is that $\mathcal{A}_2$ *is not allowed to make access to decryption oracle with the challenge* c *itself as a query. We say that* Π *is* (t, q_D, ϵ)*-secure if for every adversary* $\mathcal{A}$ *that runs at most in time* t*, achieving* $\mathbf{Adv}^{\mathsf{IND\text{-}ATK}}_{\mathcal{A},\Pi}(k) < \epsilon$*, where* q_D *is the query times on decryption oracle* $\mathcal{D}_{sk}(.)$*.*

Remark 1. The above security notion is defined in the standard model. In the random oracle model [5], one should think $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is also allowed to make access to random oracle $\mathcal{O}_H$. To date, the strongest security notion for public key encryption is IND-CCA2[1]. In the standard model, the typical IND-CCA2 public key encryption scheme is Crame-Shoup scheme [18]; and the typical IND-CCA2 public key encryption schemes in the random oracle model include OAEP [6] and others [11,24,48]. Identity-based public key cryptography is a paradigm introduced by Shamir to simplify key management and remove the necessity of public key certificates [55]. To achieve this, the user's public key should be an information which can directly identify him in a non ambiguous way, such as e-mail address, IP address, and so on. The first practical identity based encryption scheme (IBE) was found by Boneh and Franklin in 2001 [12]. Using Fujisaki-Okamoto transformation [24], the IBE can be converted to IND-CCA2 secure under adaptive chosen identity attack.

3 Cryptographic Assumptions on Non-commutative Groups

3.1 Two Well-Known Cryptographic Assumptions

In a non-commutative group G, two elements x, y are *conjugate*, written $x \sim y$, if $y = z^{-1}xz$ for some $z \in G$. Here z or z^{-1} is called a *conjugator*. Over a non-commutative group G [35], we can define the following cryptographic problems which are related to conjugacy [2]:

- Conjugator Search Problem (CSP): Given $(x, y) \in G \times G$, find $z \in G$ such that $y = z^{-1}xz$.
- Decomposition Problem (DP): Given $(x, y) \in G \times G$ and $S \subseteq G$, find $z_1, z_2 \in S$ such that $y = z_1 x z_2$.

At present, we believe that for general non-commutative group G, both of the above problems are difficult enough to be cryptographic assumptions. That is, the CSP (DP, respectively) assumption says that CSP (DP, respectively) is intractable. More precisely, the CSP (DP, respectively) assumption states that there does not exist probabilistic polynomial time algorithm which can solve CSP (DP, respectively) with non-negligible accuracy with respect to problem scale, i.e., the number of input bits of the problem.

[1] Non-malleability against adaptive chosen-ciphertext attacks (NM-CCA2) is another strongest security notions, which has been proved to be equivalent to IND-CCA2 in [4].

[2] Maybe, in theoretical, these problems are not solvable for arbitrary instance. But in practice of the cryptographic applications, we usually start from some solvable instances to construct desired schemes.

3.2 Symmetrical Decomposition and Computational Diffie-Hellman Assumptions over Non-commutative Groups

Enlightened by the above problems, we would like to define the following cryptographic problems over a non-commutative group G:

- **Symmetrical Decomposition Problem (SDP)**: Given $(x, y) \in G \times G$ and $m, n \in \mathbb{Z}$, find $z \in G$ such that $y = z^m x z^n$.
- **Generalized Symmetrical Decomposition Problem (GSDP)**: Given $(x, y) \in G \times G$, $S \subseteq G$ and $m, n \in \mathbb{Z}$, find $z \in S$ such that $y = z^m x z^n$.

Clearly, GSDP can be looked as a type of constrained SDP. In general, if the size of S is large enough and its membership information *does not* help one to extract z from $z^m x z^n$, then we believe that GSDP is at least as hard as SDP. So, in the subsequent presentation, we always address GSDP unless specific indication. Then, the GSD assumption says that GSDP is intractable, i.e., there is no probabilistic polynomial time algorithm which can solve GSDP with non-negligible accuracy with respect to problem scale.

In the above definition of GSDP, if we fix the parameters m, n, then we can define a new function on $G \times S$ as follows:

$$\begin{aligned} G \times S &\to G, \\ (x, z) &\mapsto z^m x z^n. \end{aligned}$$

Further, if we denoted $z^m x z^n$ as a new form x^z, then the above function can be looked as a newly introduce exponential operation on G with respect to its subset S [3]. Similarly, if $y = z^m x z^n$, then z can be looked as the discrete logarithm of y with respect to the base x, i.e. z can be denoted by $\log_x y$.

Now, we can regard GSDP as the discrete logarithm (DL) problem over G. Then, we can introduce the computational Diffie-Hellman (CDH) problem over G by a similar way:

- **Computational Diffie-Hellman (CDH) Problem over Non-commutative Group G (with respect to its subset S)**: Compute $x^{z_1 z_2}$ (or $x^{z_2 z_1}$) for given x, x^{z_1} and x^{z_2}, where $x \in G, z_1, z_2 \in S$.

Note that if $z_1 \in C(z_2)$, i.e., z_1 is commutative with z_2, then $x^{z_1 z_2} = x^{z_2 z_1}$ holds. It is clear that if GSDP, i.e. DL problem over G is tractable, so is CDH problem over G. But the inverse maybe is not true. At present, we have no clue to solve this kind of CDH problem without extracting z_1 (or z_2) from x and x^{z_1} (or x^{z_2}). Then, the CDH assumption over G says that CDH problem over G is intractable, i.e., there is no probabilistic polynomial time algorithm which can solve CDH problem over G with non-negligible accuracy with respect to problem scale.

The same definition can also be considered for the case when G is a non-commutative semi-group and $m, n \in \mathbb{Z}_{>0}$. One thus arrives at the concept of the GSD (also DL) and CDH assumptions over a non-commutative semi-group.

[3] We omit the clause of "with respect to its subset S" for visual comfort, unless the set S should be explicit specified.

3.3 Sampling and Disguising

Just as addressed in [35], we have to be careful when we mention instances in an infinite group G. In the current information theory, it is hard to discuss a uniform distribution in G of elements described by randomly chosen information [35]. To avoid any potential controversy, we always assume that instances to a problem are randomly chosen in a finite subset of an infinite group G restricted by system parameters [35].

Disguising is another issue we have to address here. In abstract groups, the result of multiplication is simply concatenation: $a \cdot b = ab$, thus an extra effort is always required to disguise factors in a product [59]. The importance of this is rather obvious [59]: if, for example, one transmits a conjugate $x^{-1}ax$ of a public element a "as is", i.e., without disguising, then the opponent can determine the private element x just by inspection. Choosing good disguising technique is non-trivial problem outside the scope of this paper, please refer [59] and [19] for more materials.

4 Public Key Cryptosystems Using Non-commutative Rings

4.1 Integral Coefficient Ring Polynomials

Suppose that R is a ring with $(R, +, \mathbf{0})$ and $(R, \cdot, \mathbf{1})$ as its additive abelian group and multiple non-abelian semi-group, respectively. Let us consider integral coefficient polynomials with ring assignment.

At first, the notion of scale multiplication over R is already on hand. For $k \in \mathbb{Z}_{>0}$ and $r \in R$,

$$(k)r \triangleq \underbrace{r + \cdots + r}_{k\ times}. \tag{1}$$

When $k \in \mathbb{Z}_{<0}$, we can define

$$(k)r \triangleq (-k)(-r) = \underbrace{(-r) + \cdots + (-r)}_{-k\ times}. \tag{2}$$

For $k = 0$, it is natural to define $(k)r = \mathbf{0}$.

Property 1. $(a)r^m \cdot (b)r^n = (ab)r^{m+n} = (b)r^n \cdot (a)r^m, \forall a, b, m, n \in \mathbb{Z}$ and $\forall r \in R$.

Proof. According to the definition of scale multiplication, the distributivity of multiplication with respect to addition, and commutativity of addition, this statement is concluded immediately. □

Remark 2. Note that in general, $(a)r \cdot (b)s \neq (b)s \cdot (a)r$ when $r \neq s$, since multiplication in R is non-commutative.

Now, let us proceed to define positive integral coefficient ring polynomials. Suppose that $f(x) = a_0 + a_1x + \cdots + a_nx^n \in \mathbb{Z}_{>0}[x]$ is a given positive integral coefficient polynomial. We can assign this polynomial by using an element r in R and finally obtain

$$f(r) = \sum_{i=0}^{n} (a_i)r^i = (a_0)\mathbf{1} + (a_1)r + \cdots + (a_n)r^n, \tag{3}$$

which is an element in R, of course. Further, if we regard r as a variable in R, then $f(r)$ can be looked as a polynomial about variable r. The set of all this kind of polynomials, taking over all $f(x) \in \mathbb{Z}_{>0}[x]$, can be looked the extension of $\mathbb{Z}_{>0}$ with r, denoted by $\mathbb{Z}_{>0}[r]$. For convenience, we call it the set of 1-ary *positive integral coefficient R-polynomials.*

Suppose that $f(r) = \sum_{i=0}^{n}(a_i)r^i \in \mathbb{Z}_{>0}[r]$, $h(r) = \sum_{j=0}^{m}(b_j)r^j \in \mathbb{Z}_{>0}[r]$ and $n \geq m$, then

$$\left(\sum_{i=0}^{n}(a_i)r^i\right) + \left(\sum_{j=0}^{m}(b_j)r^j\right) = \left(\sum_{i=0}^{m}(a_i + b_i)r^i\right) + \left(\sum_{i=m+1}^{n}(a_i)r^i\right), \tag{4}$$

and according to Property 1 as well as the distributivity, we have

$$\left(\sum_{i=0}^{n}(a_i)r^i\right) \cdot \left(\sum_{j=0}^{m}(b_j)r^j\right) = \sum_{i=0}^{n+m}(p_i)r^i, \tag{5}$$

where $p_i = \sum_{j=0}^{i} a_j b_{i-j} = \sum_{j+k=i} a_j b_k$. And then, we can conclude immediately the following theorem according to Property 1.

Theorem 1. $f(r) \cdot h(r) = h(r) \cdot f(r), \forall f(r), h(r) \in \mathbb{Z}_{>0}[r]$.

Remark 3. If r and s are two different variable, then $f(r) \cdot h(s) \neq h(s) \cdot f(r)$ in general.

4.2 Further Assumptions on Non-commutative Rings

Suppose that $(R, +, \cdot)$ is a non-commutative ring. For any randomly picked element $a \in R$, we define a set $P_a \subseteq R$ by

$$P_a \triangleq \{f(a) : f(x) \in \mathbb{Z}_{>0}[x]\}.$$

Then, let us consider the new versions of GSD and CDH problems over $(R, \cdot)$ with respect to its subset P_a, and name them as polynomial symmetric decomposition (PSD) problem and polynomial Diffie-Hellman (PDG) problem respectively:

- **Polynomial Symmetrical Decomposition (PSD) Problem over Non-commutative Ring** R: Given $(a, x, y) \in R^3$ and $m, n \in \mathbb{Z}$, find $z \in P_a$ such that $y = z^m x z^n$.
- **Polynomial Diffie-Hellman (PDH) Problem over Non-commutative Ring** R: Compute $x^{z_1 z_2}$ (or $x^{z_2 z_1}$) for given a, x, x^{z_1} and x^{z_2}, where $a, x \in R, z_1, z_2 \in P_a$.

Accordingly, the PSD (PDH, respectively) cryptographic assumption says that PSD (PDH, respectively) problem over $(R, \cdot)$ is intractable, i.e., there does not exist probabilistic polynomial time algorithm which can solve PSD (PDH, respectively) problem over $(R, \cdot)$ with non-negligible accuracy with respect to problem scale.

4.3 Diffie-Hellman-Like Key Agreement Protocol from Non-commutative Rings

Now, let us take R as the underlying work fundamental infrastructure and design a Diffie-Hellman-like key exchange protocol, by which two entities, say Alice and Bob, can reach an agreement on a shared, secret session key via a public, insecure network.

The protocol is described as follows:

(0) One of the entities (say, Alice) sends two random small, positive integers (say, less than 10) $m, n \in \mathbb{Z}_{>0}$ and two random[4] elements $a, b \in R$ to another entity (say, Bob) as the signal of launching the protocol.
(1) Alice chooses a random polynomial $f(x) \in \mathbb{Z}_{>0}[x]$ such that $f(a) \neq \mathbf{0}$ and then takes $f(a)$ as her private key.
(2) Bob chooses a random polynomial $h(x) \in \mathbb{Z}_{>0}[x]$ such that $h(a) \neq \mathbf{0}$ and then takes $h(a)$ as his private key.
(3) Alice computes $r_A = f(a)^m \cdot b \cdot f(a)^n$ and sends[5] r_A to Bob.
(4) Bob computes $r_B = h(a)^m \cdot b \cdot h(a)^n$ and sends r_B to Alice.
(5) Alice computes $K_A = f(a)^m \cdot r_B \cdot f(a)^n$ as the shared session key.
(6) Bob computes $K_B = h(a)^m \cdot r_A \cdot h(a)^n$ as the shared session key.

In practice, the steps (0), (1) and (3) can be finished simultaneously and require only one pass communication from Alice to Bob. After that, the steps (2) and (4) can be finished in one pass communication from Bob to Alice. Finally, Alice and Bob can execute the steps (5) and (6) respectively, needless further communication. A high-level depiction of the protocol is given in Figure 1.

Pass	Alice	Bob
	Chooses $m, n \in \mathbb{Z}_{>0}$ at random Chooses $a, b \in R$ at random Chooses $f(x) \in \mathbb{Z}_{>0}[x]$ at random	
1	$\xrightarrow{m,n,a,b,f(a)^m bf(a)^n}$	
		Chooses $h(x) \in \mathbb{Z}_{>0}[x]$ at random
2		$\xleftarrow{h(a)^m bh(a)^n}$
	$K_A = f(a)^m h(a)^m bh(a)^n f(a)^n$ $=$	$K_B = h(a)^m f(a)^m bf(a)^n h(a)^n$

Fig. 1. Diffie-Hellman-Like Key Agreement Based on Non-commutative Ring

It is trivial to prove that the above key agreement protocol can resist passive adversary under the PDH assumption over the non-commutative monoid $(R, \cdot)$. Obviously, similar to the standard Diffie-Hellman protocol [20], the protocol depicted in Figure 1 cannot resist the man-in-the-middle (MIM) attack. The revising work is a meaningful but little tough task which is left for interested readers.

4.4 ElGamal-Like Encryption Scheme From Non-commutative Rings

Based on the above key agreement, it is straightforward to describe an ElGamal-like encryption scheme as follows.

[Basic Scheme]

[4] See Section 3.3 for the sampling issue.

[5] In practice, the element has to be disguised by certain canonical form before it is transmitted via the public channel. Please see Section 3.3 for the disguising issue.

- **Initial setup:** Suppose that the non-commutative ring $(R, +, \cdot)$ is the underlying work fundamental infrastructure and SDP is intractable on the monoid $(R, \cdot)$. Pick two small positive integers $m, n \in \mathbb{Z}_{>0}$. Let $H : R \to \mathcal{M}$ be a cryptographic hash function which maps R to the message space $\mathcal{M}$. Then, the public parameters of the system would be the tuple $< R, m, n, \mathcal{M}, H >$.
- **Key generation:** Each user chooses two random elements $p, q \in R$ and a random polynomial $f(x) \in \mathbb{Z}_{>0}[x]$ such that $f(p) \neq \mathbf{0}$ and then takes $f(p)$ as his private key, computes $y = f(p)^m \cdot q \cdot f(p)^n$ and publishes his public key $(p, q, y) \in R^3$.
- **Encryption:** Given a message $M \in \mathcal{M}$ and receiver's key $(p, q, y) \in R^3$, the sender chooses a random polynomial $h(x) \in \mathbb{Z}_{>0}[x]$ such that $h(p) \neq \mathbf{0}$ and then takes $h(p)$ as salt, computes

  $$c = h(p)^m \cdot q \cdot h(p)^n, \ d = H(h(p)^m \cdot y \cdot h(p)^n) \oplus M,$$

  and finally outputs the ciphertext $(c, d) \in R \times \mathcal{M}$.
- **Decryption:** Upon receiving a ciphertext $(c, d) \in R \times \mathcal{M}$, the receiver, by using his private key $f(p)$, computes the plaintext

  $$M = H(f(p)^m \cdot c \cdot f(p)^n) \oplus d$$

First, we present an "all or nothing" security result for the above basic encryption scheme. The statement of the result as well as the proof technique are very similar to Theorem 8.3 of [44], except an additional random oracle assumption on H.

Theorem 2. *For a plaintext message uniformly distributed in the plaintext message space, the above cryptosystem is "all-or-nothing" secure against CPA under the PDH assumption over the non-commutative ring $(R, +, \cdot)$ provided that H is a random oracle.*

Proof. On the one hand, if PDH problem is tractable, for any given ciphertext pair (c, d) and the corresponding public key (p, q, y), it is easy to compute $k = q^{(\log_q c)(\log_q y)}$ from the triple (q, c, y) and then extract the plaintext $M = d \oplus H(k)$.

On the other hand, suppose on the contrary there exists an efficient adversary $\mathcal{A}$, with access to the random oracle H, against the above cryptosystem, that is, given any public key $(p, q, y = f(p)^m q f(p)^n)$ and ciphertext (c, d), $\mathcal{A}$ outputs

$$M \leftarrow \mathcal{A}^H(p, q, y, c, d)$$

with a non-negligible advantage ϵ such that M satisfies

$$M = d \oplus H(y^{\log_q c}) = d \oplus H(q^{(\log_q y)(\log_q c)}), \text{ i.e. } M = d \oplus H(h^m y h^n) \text{ and } c = h^m q h^n$$

for some $h \in P_p$. Then, for an arbitrary PDH instance (a, x, x^{z_1}, x^{z_2}). We set (a, x, x^{z_1}) as public key and set (x^{z_2}, d) as ciphertext pair for a random $d \in \mathcal{M}$. Then, with the advantage ϵ, $\mathcal{A}$ outputs

$$M \leftarrow \mathcal{A}^H(a, x, x^{z_1}, x^{z_2}, d)$$

with M satisfying

$$M = d \oplus H(x^{z_1 z_2}), \text{ i.e. } M = d \oplus H(z_2^m z_1^m x z_1^n z_2^n)$$

for some $z_2 \in P_a$. Recall that $z_1 \in P_a$, thus $z_2 z_1 = z_1 z_2$ according to Theorem 1. Then,

$$M = d \oplus H(z_2^m z_1^m x z_1^n z_2^n) = d \oplus H(x^{z_1 z_2}) = d \oplus H(x^{z_2 z_1}).$$

Clearly, if the adversary $\mathcal{A}$'s advantage ϵ is non-negligible, then $\mathcal{A}$ must make corresponding H-query on $x^{z_1 z_2}$; Otherwise, since H is modeled as a cryptographic hash, $\mathcal{A}$'s advantage should be *negligible* no matter what he can compute before making such a query.

With the random oracle assumption on H, we can maintain a H-list which contains two fields (r_i, h_i) and is initialized with empty. Whenever the adversary $\mathcal{A}$ makes a H-query with input r, we examine whether there exists the pair (r, h) in H-list. If so, return h as the answer to $\mathcal{A}$; Otherwise, randomly pick $h \in \mathcal{M}$, add the pair (r, h) into H-list and return h as the answer to $\mathcal{A}$. Clearly, the simulation on H is perfect. Finally, when $\mathcal{A}$ outputs M, we can retrieval the correct item $x^{z_1 z_2} = r_i$ by checking the equality $M = d \oplus h_i$. Thus, we can solve PDH problem with the non-negligible probability ϵ. This contradicts the holding of the PDH assumption. □

The above theorem just say that the basic scheme reaches the weakest security, i.e. the OW-CPA security. Although we can use a technique due to Fjisaki-Okamoto (at CRYPTO'99) [25] to convert the above basic scheme into a chosen ciphertext secure system in the random oracle, we would like to adopt another technique also due to Fjisaki-Okamoto (at PKC'99) [24] to reach the same goal, since the latter is more direct than the former. Before to do this, we have at first prove that the above basic scheme reaches the IND-CPA security.

Theorem 3 (IND-CPA of Basic Scheme). *Let H be a random oracle from R to $\mathcal{M}$. Let $\mathcal{A}$ be an IND-CPA adversary that has advantage ϵ against the above basic scheme within t steps. Suppose $\mathcal{A}$ makes a total of $q_H > 0$ queries to H. Then there is an algorithm $\mathcal{B}$ that solves PDH problem over the non-commutative ring R with advantage at least ϵ' within t' steps, where*

$$\epsilon' = \frac{2\epsilon}{q_H}, \textit{ and } t' = \mathcal{O}(t).$$

Proof. Algorithm $\mathcal{B}$ is given as input a 4-tuple (a, x, y_1, y_2) with $y_i = x^{z_i} = z_i^m x z_i^n$ for *unknown* $z_i \in P_a$, $i = 1, 2$, i.e., an instance of PDH problem. Let $y = x^{z_1 z_2}$ denote the solution to PDH problem on this instance.

- **Setup.** At first, the algorithm B sets the system parameters to be $< R, m, n, \mathcal{M}, H >$ and creates a public key (a, x, y_1). Both the system parameters and the public key should be available to the adversary $\mathcal{A}$.
- **H-queries.** Then, B maintains a H-list which contains two fields (r_j, h_j) and is initialized with empty. Whenever the adversary $\mathcal{A}$ makes a H-query with input r, B examines whether there exists the pair (r, h) in H-list. If so, returns h as the answer to $\mathcal{A}$; Otherwise, randomly picks $h \in \mathcal{M}$, adds the pair (r, h) into H-list and returns h as the answer to $\mathcal{A}$. Clearly, the simulation on H is perfect.
- **Challenge.** When A outputs two messages M_0 and M_1 on which it wished to be challenged. B picks randomly a string $d \in \mathcal{M}$ and defines C to be the ciphertext pair $C = (y_2, d)$. It then gives C to A as the challenge. Notice that, by definition, the decryption of C is $d \oplus H(x^{(\log_x y_1)(\log_x y_2})= d \oplus H(x^{z_1 z_2}) = d \oplus H(y)$. (Recall that z_1, z_2 and y are all unknown and y is just the solution to the above instance of PDH.)
- **Guess.** A outputs its guess $b' \in \{0, 1\}$. At this point, B picks a random tuple (r_j, h_j) from the H-list and outputs r_j as the solution to the given instance of PDH.

It is easy to see that $\mathcal{A}$'s view in $\mathcal{B}$'s simulation is the same as in a real attack, in other words, the simulation is perfect. So $\mathcal{A}$'s advantage in this simulation will be ϵ. We let $\mathcal{H}$ be the event that y is queried to H oracle during $\mathcal{B}$'s simulation.

Notice that $H(y)$ is independent of $\mathcal{A}$'s view, so if $\mathcal{A}$ never queries y to the H oracle in the above simulation, then the decryption of C is also independent of its view. Therefore, in the simulation we have $\Pr[b = b'|\neg\mathcal{H}] = 1/2$. By the definition of $\mathcal{A}$, we know that in the real attack (and also in the simulation) $\mid \Pr[b = b'] - 1/2 \mid \geq \epsilon$. We have the following bounds on $\Pr[b = b']$:

$$\begin{aligned}
\Pr[b = b'] &= \Pr[b = b'|\neg\mathcal{H}]\Pr[\neg\mathcal{H}] + \Pr[b = b'|\mathcal{H}]\Pr[\mathcal{H}] \\
&\leq \Pr[b = b'|\neg\mathcal{H}]\Pr[\neg\mathcal{H}] + \Pr[\mathcal{H}] \\
&= \frac{1}{2}\Pr[\neg\mathcal{H}] + \Pr[\mathcal{H}] \\
&= \frac{1}{2} + \frac{1}{2}\Pr[\mathcal{H}],
\end{aligned}$$

$$\begin{aligned}
\Pr[b = b'] &\geq \Pr[b = b'|\neg\mathcal{H}]\Pr[\neg\mathcal{H}] \\
&= \frac{1}{2}\Pr[\neg\mathcal{H}] \\
&= \frac{1}{2}(1 - \Pr[\mathcal{H}]) \\
&= \frac{1}{2} - \frac{1}{2}\Pr[\mathcal{H}]).
\end{aligned}$$

Hence we have $\mid \Pr[b = b'] - 1/2 \mid \leq \frac{1}{2}\Pr[\mathcal{H}]$. By $\mid \Pr[b = b'] - 1/2 \mid \geq \epsilon$ we know that $\Pr[\mathcal{H}] \geq 2\epsilon$. Furthermore, by the definition of the event $\mathcal{H}$, we know that y appears in some tuple on the H-list with probability at least 2ϵ. It follows that $\mathcal{B}$ outputs the correct answer to the above instance of PDH with probability at least $2\epsilon/q_H$ as required. □

At PKG'99, Fujisaki and Okamoto [24] introduced a method to convert an IND-CPA encryption scheme into an IND-CCA2 scheme. For self-containing, we rehearse their main idea as follows:

Suppose $\mathit{\Pi} := \{\mathcal{K}, \mathcal{E}, \mathcal{D}\}$ is an IND-CPA secure public-key encryption scheme with key generation algorithm $\mathcal{K}(1^k)$, encryption algorithm $\mathcal{E}_{pk}(X, S)$ and decryption algorithm $\mathcal{D}_{sk}(y)$, where pk and sk are a public key and the corresponding private key, X a message with $k + k_0$ bits, S a random string with l bits and y a ciphertext. The converted public-key encryption scheme $\bar{\mathit{\Pi}} := \{\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}}\}$ is defined by

$$\begin{aligned}
\bar{\mathcal{K}}(1^k) &:= \mathcal{K}(1^{k+k_0}), \\
\bar{\mathcal{E}}_{pk}(x, r) &:= \mathcal{E}_{pk}(x \parallel r, H(x \parallel r)), \\
\bar{\mathcal{D}}_{sk}(y) &:= \begin{cases} [\mathcal{D}_{sk}(y)]^k, & \text{if } y = \mathcal{E}_{pk}(\mathcal{D}_{sk}(y), H(\mathcal{D}_{sk}(y))) \\ \perp, & \text{otherwise} \end{cases}
\end{aligned}$$

where H is a random function of $\{0,1\}^{k+k_0} \to \{0,1\}^l$, x is a message with k bits, r a random string with k_0 bits and $\parallel$ denotes concatenation.

Theorem 4 (Fujisaki-Okamoto Theorem [24]). *Suppose that $\mathit{\Pi}(1^{k+k_0})$ is the original IND-CPA secure scheme and $\bar{\mathit{\Pi}}$ is the converted scheme. If there exists a (t, q_H, q_D, ϵ)-breaker A for $\bar{\mathit{\Pi}}(1^k)$ in the sense of IND-CCA2 in the random oracle model, there exist constant c and a $(t', 0, 0, \epsilon')$-breaker A' for $\mathit{\Pi}(1^{k+k_0})$ where*

$$\begin{aligned}
&\epsilon' = (\epsilon - q_H \cdot 2^{-(k_0-1)}) \cdot (1 - 2^{-l_0})^{q_D} \text{ and} \\
&t' = t + q_H \cdot (T_{\mathcal{E}}(k) + c \cdot k).
\end{aligned}$$

Here, (t, q_H, q_D, ϵ)-breaker A, informally, means that A stops within t steps, succeeds with probability at least ϵ, makes at most q_H queries to random oracle H, and makes at most q_D queries to decryption oracle D_{sk}. $T_{\mathcal{E}}(k)$ denotes the computational time of the encryption algorithm $\mathcal{E}_{pk}(\cdot)$, and $l_0 := \log_2(\min_{x \in \{0,1\}^{k+k_0}}[\#\{E_{pk}(x,r)|r \in \{0,1\}^l\}])$.

Proof. See Theorem 3 of [24].

According to Fujisaki-Okamoto[24], with sacrificing of k_0 bits plaintext, we can convert our basic encryption scheme into an enhanced one, which reaches IND-CCA2 security. In the enhanced scheme, system parameters $< R, m, n, \mathcal{M} >$ are as same as those in the basic scheme. The cryptographic hash function H in basic scheme is replaced with two new cryptographic hash functions $H_1 : \{0,1\}^{k+k_0} \to \mathbb{Z}_{>0}[x]$ and $H_2 : R \to \{0,1\}^{k+k_0}$, where k is the standard length of a message, i.e., $\mathcal{M} = \{0,1\}^k$, while k_0 is the length of random salt that should not be determined by binary search method (for example, we set $k_0 = 128$).

Now, the enhanced encryption scheme which achieves IND-CCA2 security, is described as follows.

[Enhanced Scheme]

- **Initial setup:** System public parameters include $R, m, n, \mathcal{M}$ and k_0, H_1, H_2.
- **Key generation:** Identical to the **Key Generation** step in the basic scheme.
- **Encryption:** Given a message $M \in \mathcal{M}$ and receiver's key $(p, q, y = f(p)^m \cdot q \cdot f(p)^n) \in R^3$, the sender chooses a random salt $r \in \{0,1\}^{k_0}$ and *extracts*[6] a polynomial $h(x) = H_1(M \parallel r) \in \mathbb{Z}_{>0}[x]$ such that $h(p) \neq \mathbf{0}$ and then computes

  $$c = h(p)^m \cdot q \cdot h(p)^n,\ \ d = H_2(h(p)^m \cdot pk \cdot h(p)^n) \oplus (M \parallel r),$$

  and finally outputs the ciphertext $(c, d) \in R \times \{0,1\}^{k+k_0}$.
- **Decryption:** Upon receiving a ciphertext $(c, d) \in R \times \{0,1\}^{k+k_0}$, the receiver, by using his private key $f(p)$, computes

  $$M' = H_2(f(p)^m \cdot c \cdot f(p)^n) \oplus d.$$

  Finally, extracts $g(x) = H_1(M') \in \mathbb{Z}_{>0}[x]$ and checks whether $c = g(p)^m \cdot q \cdot g(p)^n$ holds. If so, outputs the beginning k bits of M'; otherwise, outputs empty string, which means that the given ciphertext is invalid.

Analogously, the enhanced scheme is the result of applying the Fujisaki-Okamoto [24] transformation to the basic scheme. Based on Theorem 3 and Theorem 4, we have

Theorem 5 (IND-CCA2 of Enhanced Scheme). *Let H_1 and H_2 be random oracles. Then the enhanced scheme is an adaptively chosen ciphertext secure encryption (IND-CCA2) assuming PDH over the non-commutative ring R is hard. More specifically, suppose there is an IND-CCA2 adversary $\mathcal{A}$ that has advantage ϵ against the enhanced scheme within t steps. Suppose $\mathcal{A}$ makes at most q_D decryption queries, and at most q_{H_1}, q_{H_2} queries to the hash functions H_1, H_2 respectively. Then there is an algorithm $\mathcal{B}$ which can solve PDH with the probability at least ϵ' within t' steps, where*

$$\epsilon' = \frac{2}{q_{H_1}}\left[\frac{\epsilon}{(1-2^{-l_0})^{q_D}} + q_{H_2} \cdot 2^{-(k_0-1)}\right], \text{ and}$$
$$t' = \mathcal{O}\left(t - q_{H_2} \cdot (T_{\mathcal{E}}(k) + c \cdot k)\right)$$

[6] See Remark 4 for further discussion.

where c is a constant and $T_{\mathcal{E}}(k)$ denotes the computational time of the encryption algorithm $\mathcal{E}_{pk}(\cdot)$ in our basic scheme, and $l_0 := \log_2(\min_{x\in\{0,1\}^{k+k_0}}[\#\{E_{pk}(x,r)|r \in \{0,1\}^l\}])$.

Proof. At first, from Theorem 3 and Theorem 4, it immediately concludes that our enhanced encryption scheme reaches IND-CCA2 security in the random oracle model assuming that PDH is hard. Then, by combining the results of both the IND-CPA theorem and Fujisaki-Okamoto theorem, we obtain the above bounds. □

Remark 4. It is worth noting the elaborations on implementing a cryptographic hash that maps a binary string to a polynomial, such as $H_1 : \{0,1\}^{k+k_0} \to \mathbb{Z}_{>0}[x]$. In particular, the resulting polynomials should satisfy further constraints, such as the condition $h(p) \neq \mathbf{0}$ and so on. We employ the so-called *divide-and-conquer* strategy to solve this problem: At first, we extract a polynomial $h(x) \in \mathbb{Z}_{>0}[x]$ from a binary string in $\{0,1\}^{k+k_0}$; Then, we adopt a unique, deterministic way to rectify $h(x)$ to $\tilde{h}(x)$ such that $\tilde{h}(x)$ satisfies the desired condition C. In other words, we have to consider the following issues in designing the desired hash:

- **Extracting.** In practice, we prefer to choose polynomials with low degrees and large coefficients. Let us assume that the highest degree is d_H and the maximal coefficient is c_M, then $d_H \cdot c_M$ should be large enough to resist brute force attack. Thus, there is a trivial solution to implement H_1: Suppose that we already have a cryptographic hash function H' which maps $\{0,1\}^{k+k_0}$ to $\mathbb{Z}_{c_M}^{d_H+1}$. Then, for any given image of H_1, i.e., a vector $(z_0, z_1, \cdots, z_{d_H}) \in \mathbb{Z}_{c_M}^{d_H+1}$, we can map it to a goal polynomial $h(x)$ by a natural way:
$$h(x) = z_0 + z_1 x + \cdots + z_{d_H} x^{d_H}. \tag{6}$$
- **Rectifying.** Suppose we adopt an additive rectifying strategy. Then, for resulting polynomial $h(x)$, it can be rectified to $\tilde{h}(x) = h(x) + \Delta$ while
$$\Delta = \min\{\delta \in \mathbb{Z}_{\geq 0} : h(x) + \delta \cdot \mathbf{1}_R \in \mathbb{Z}_{>0}[x] \cap C\},$$
where $\mathbb{Z}_{>0}[x] \cap C$ is the set of polynomials in $\mathbb{Z}_{>0}[x]$ satisfying the given condition C.
- **Collision-Resisting.** The above rectifying strategy should not violate the property of collision resistance. In fact, the collision resistance of H_1 is rooted in the one-wayness of H'.

4.5 Concrete Examples: Public Key Cryptosystems Using Matrix Rings

Let us illustrate our method by using a special matrix ring: $M_2(Z_N)$, where $N = p \cdot q$ while p and q are two large secure primes. We have solid reason to believe that SDP over $M_2(Z_N)$ is intractable, since it is infeasible to extract
$$A = \begin{pmatrix} a & 0 \\ 0 & 0 \end{pmatrix} \in M_2(Z_N), a \in Z_N$$
from
$$A^2 = \begin{pmatrix} a^2 \mod N & 0 \\ 0 & 0 \end{pmatrix} \in M_2(Z_N)$$
without knowing the factoring of N.

Example 1. Diffie-Hellman-Like Key Agreement Using Matrix Rings

Let $N = 7 \cdot 11$ for simplicity[7]. Suppose that Alice chooses

$$m = 3, n = 5, A = \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix}, B = \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix}, \text{ and } f(x) = 3x^3 + 4x^2 + 5x + 6.$$

She computes

$$f(A) = 3 \cdot \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix}^3 + 4 \cdot \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix}^2 + 5 \cdot \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix} + 6 \cdot I = \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix},$$

and

$$r_A = \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^3 \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^5 = \begin{pmatrix} 49 & 53 \\ 42 & 31 \end{pmatrix}.$$

Then, she sends m, n, A, B and r_A to Bob.

Now, suppose that Bob, upon receiving m, n, A, B and r_A from Alice, chooses another polynomial $h(x) = x^5 + 5x + 1$ and computes

$$h(A) = \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix}^5 + 5 \cdot \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix} + I = \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix},$$

and

$$r_B = \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^3 \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^5 = \begin{pmatrix} 29 & 40 \\ 52 & 6 \end{pmatrix}.$$

Then, he sends r_B to Alice.

Finally, Alice extracts the session key

$$K_A = \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^3 \begin{pmatrix} 29 & 40 \\ 52 & 6 \end{pmatrix} \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^5 = \begin{pmatrix} 28 & 37 \\ 14 & 40 \end{pmatrix},$$

while Bob extracts the session key

$$K_B = \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^3 \begin{pmatrix} 49 & 53 \\ 42 & 31 \end{pmatrix} \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^5 = \begin{pmatrix} 28 & 37 \\ 14 & 40 \end{pmatrix}.$$

Apparently, $K_A = K_B$ holds, i.e., the key agreement is successful.

Example 2. Encryption/Decryption Using Matrix Rings

At first, we have to define the message space $\mathcal{M}$ as well as cryptographic hash functions H for the basic scheme and H_1, H_2 for the enhanced scheme (note that in this subsection we always define $R \triangleq M_2(Z_N)$). For simplicity, we assume that $\mathcal{M} \triangleq M_2(Z_N)$ for the basic scheme and $\mathcal{M} \triangleq \left\{ \begin{pmatrix} a & b \\ c & 0 \end{pmatrix} : a, b, c \in Z_N \right\}$ for the enhanced scheme, while

$$H : M_2(Z_N) \to \mathcal{M} = M_2(Z_N), m_{ij} \mapsto 2^{m_{ij}} \mod N,$$

[7] Although the modular N in our *toy* examples is too small, it is enough to illustrate our method.

and

$$H_1 : \mathcal{M} \times Z_N \to \mathbb{Z}_{>0}[x], \left(\begin{pmatrix} a & b \\ c & 0 \end{pmatrix}, r \right) \mapsto 2^r + 2^a x + 2^b x^2 + 2^c x^3,$$

where all coefficients $2^r, 2^a, 2^b$ and 2^c should be regarded as elements in Z_N. For more simplicity, we define

$$(M \parallel r) \triangleq \begin{pmatrix} a & b \\ c & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & r \end{pmatrix} = \begin{pmatrix} a & b \\ c & r \end{pmatrix}.$$

Then, $H_2 : M_2(Z_N) \to \mathcal{M} \times Z_N$ can be defined as

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto (M \parallel r), \text{ where } M = \begin{pmatrix} 2^a & 2^b \\ 2^c & 0 \end{pmatrix} \mod N, r = 2^d \mod N.$$

Next, let $N = 7 \cdot 11$ for example. Suppose that the left system parameters are

$$m = 3,\ n = 5,\ p = \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix},\ q = \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix}.$$

[Encryption/Decryption with Basic Scheme]

Suppose that the polynomial $f(x)$ picked by Alice is just that of in example 1. Then, Alice' private key is $f(p) = \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}$, just as $f(A)$ in Example 1. Then, the corresponding public key would be $pk \triangleq f(p)^3 q f(p)^5 = \begin{pmatrix} 49 & 53 \\ 42 & 31 \end{pmatrix}$, just as r_A in Example 1.

Let us pick a message M randomly, say $M = \begin{pmatrix} 27 & 19 \\ 34 & 8 \end{pmatrix}$. Suppose the salt polynomial we picked randomly is coincide to $h(x)$ in Example 1. Then, the salt matrix $h(p) = \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}$, just as $h(A)$ in Example 1. Now, let us compute the ciphertext (c, d) as follows:

$$c = h(p)^3 q h(p)^5 = \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^3 \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^5 = \begin{pmatrix} 29 & 40 \\ 52 & 6 \end{pmatrix},$$

and

$$\begin{aligned} d &= H(h(p)^3 \cdot pk \cdot h(p)^5) \oplus M \\ &= H\left(\begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^3 \begin{pmatrix} 49 & 53 \\ 42 & 31 \end{pmatrix} \begin{pmatrix} 64 & 13 \\ 49 & 23 \end{pmatrix}^5 \right) \oplus \begin{pmatrix} 27 & 19 \\ 34 & 8 \end{pmatrix} \\ &= H\left(\begin{pmatrix} 28 & 37 \\ 14 & 40 \end{pmatrix} \right) \oplus \begin{pmatrix} 27 & 19 \\ 34 & 8 \end{pmatrix} \\ &= \left(\begin{pmatrix} 2^{28} & 2^{37} \\ 2^{14} & 2^{40} \end{pmatrix} \mod N \right) \oplus \begin{pmatrix} 27 & 19 \\ 34 & 8 \end{pmatrix} \\ &= \begin{pmatrix} 58 & 51 \\ 60 & 23 \end{pmatrix} \oplus \begin{pmatrix} 27 & 19 \\ 34 & 8 \end{pmatrix} \\ &= \begin{pmatrix} 33 & 32 \\ 30 & 31 \end{pmatrix}. \end{aligned}$$

Now, let us check the decryption process:

$$\begin{aligned}
M' &= H(f(p)^3 \cdot c \cdot f(p)^5) \oplus d \\
&= H\left(\begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^3 \begin{pmatrix} 29 & 40 \\ 52 & 6 \end{pmatrix} \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^5\right) \oplus \begin{pmatrix} 33 & 32 \\ 30 & 31 \end{pmatrix} \\
&= H\left(\begin{pmatrix} 28 & 37 \\ 14 & 40 \end{pmatrix}\right) \oplus \begin{pmatrix} 33 & 32 \\ 30 & 31 \end{pmatrix} \\
&= \left(\begin{pmatrix} 2^{28} & 2^{37} \\ 2^{14} & 2^{40} \end{pmatrix} \mod N\right) \oplus \begin{pmatrix} 33 & 32 \\ 30 & 31 \end{pmatrix} \\
&= \begin{pmatrix} 58 & 51 \\ 60 & 23 \end{pmatrix} \oplus \begin{pmatrix} 33 & 32 \\ 30 & 31 \end{pmatrix} \\
&= \begin{pmatrix} 27 & 19 \\ 34 & 8 \end{pmatrix} \\
&= M.
\end{aligned}$$

[Encryption/Decryption with Enhanced Scheme]

Suppose that the private/public keys are unchanged. Let us pick a message M randomly, say $M = \begin{pmatrix} 27 & 19 \\ 34 & 0 \end{pmatrix}$. Suppose the salt number we picked randomly is $r = 35$. Then, we extract a polynomial as follows:

$$\begin{aligned}
h(x) &= (2^{35} \mod N) + (2^{27} \mod N)x + (2^{19} \mod N)x^2 + (2^{34} \mod N)x^3 \\
&= 32 + 29x + 72x^2 + 16x^3.
\end{aligned}$$

Thus,

$$h(p) = 32 \cdot I + 29 \cdot \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix} + 72 \cdot \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix}^2 + 16 \cdot \begin{pmatrix} 2 & 5 \\ 7 & 4 \end{pmatrix}^3 = \begin{pmatrix} 37 & 30 \\ 42 & 49 \end{pmatrix} \neq \mathbf{0}.$$

(Note that if $h(x)$ does not satisfy the condition of $h(p) \neq \mathbf{0}$, we should at first rectify $h(x)$ to $\tilde{h}(x) = h(x) + \mathit{\Delta}$, where

$$\mathit{\Delta} = \min\left\{\delta \in \mathbb{Z}_{\geq 0} : h(p) + \delta \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \neq \mathbf{0}\right\}.$$

Fortunately, in this example. The above extracted $h(x)$ meets the requirement of $h(p) \neq \mathbf{0}$, i.e., $\mathit{\Delta} = 0$.)

Then, then cipertext pair is

$$c = h(p)^3 q h(p)^5 = \begin{pmatrix} 37 & 30 \\ 42 & 49 \end{pmatrix}^3 \begin{pmatrix} 1 & 9 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} 37 & 30 \\ 42 & 49 \end{pmatrix}^5 = \begin{pmatrix} 65 & 37 \\ 35 & 7 \end{pmatrix},$$

and

$$\begin{aligned} d &= H(h(p)^3 \cdot pk \cdot h(p)^5) \oplus (M \parallel r) \\ &= H\left(\begin{pmatrix} 37 & 30 \\ 42 & 49 \end{pmatrix}^3 \begin{pmatrix} 49 & 53 \\ 42 & 31 \end{pmatrix} \begin{pmatrix} 37 & 30 \\ 42 & 49 \end{pmatrix}^5\right) \oplus \begin{pmatrix} 27 & 19 \\ 34 & 35 \end{pmatrix} \\ &= H\left(\begin{pmatrix} 21 & 14 \\ 21 & 7 \end{pmatrix}\right) \oplus \begin{pmatrix} 27 & 19 \\ 34 & 35 \end{pmatrix} \\ &= \left(\begin{pmatrix} 2^{21} & 2^{14} \\ 2^{21} & 2^7 \end{pmatrix} \mod N\right) \oplus \begin{pmatrix} 27 & 19 \\ 34 & 35 \end{pmatrix} \\ &= \begin{pmatrix} 57 & 60 \\ 57 & 51 \end{pmatrix} \oplus \begin{pmatrix} 27 & 19 \\ 34 & 35 \end{pmatrix} \\ &= \begin{pmatrix} 34 & 47 \\ 27 & 16 \end{pmatrix}. \end{aligned}$$

Now, let us check the enhanced decryption process:

$$\begin{aligned} M' &= H(f(p)^3 \cdot c \cdot f(p)^5) \oplus d \\ &= H\left(\begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^3 \begin{pmatrix} 65 & 37 \\ 35 & 7 \end{pmatrix} \begin{pmatrix} 35 & 12 \\ 63 & 9 \end{pmatrix}^5\right) \oplus \begin{pmatrix} 34 & 47 \\ 27 & 16 \end{pmatrix} \\ &= H\left(\begin{pmatrix} 21 & 14 \\ 21 & 7 \end{pmatrix}\right) \oplus \begin{pmatrix} 34 & 47 \\ 27 & 16 \end{pmatrix} \\ &= \left(\begin{pmatrix} 2^{21} & 2^{14} \\ 2^{21} & 2^7 \end{pmatrix} \mod N\right) \oplus \begin{pmatrix} 34 & 47 \\ 27 & 16 \end{pmatrix} \\ &= \begin{pmatrix} 57 & 60 \\ 57 & 51 \end{pmatrix} \oplus \begin{pmatrix} 34 & 47 \\ 27 & 16 \end{pmatrix} \\ &= \begin{pmatrix} 27 & 19 \\ 34 & 35 \end{pmatrix} \\ &= \begin{pmatrix} 27 & 19 \\ 34 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 35 \end{pmatrix} \\ &= M \parallel r. \end{aligned}$$

5 Public Key Cryptosystems Using Non-commutative Groups and Semi-groups

The method described in the above section is suite for general non-commutative rings. A natural question is: Can we transfer these results to general non-commutative groups and non-commutative semi-groups by similar ways?

5.1 Extension of Non-commutative Groups

Now, given a non-commutative group $(G, \cdot, 1_G)$. Suppose that there is a ring $(R, +, \cdot, 1_R)$ and a monomorphism $\tau : (G, \cdot, 1_G) \to (R, \cdot, 1_R)$. Then, the inverse map $\tau^{-1} : \tau(G) \to G$ is also a well-defined monomorphism and for $a, b \in G$, if $\tau(a) + \tau(b) \in \tau(G)$, we can assign a new element $c \in G$ as

$$c \triangleq \tau^{-1}(\tau(a) + \tau(b)), \tag{7}$$

and call c as the *quasi-sum* of a and b, denoted by $c = a \boxplus b$. Similarly, for $k \in R$ and $a \in G$, if $k \cdot \tau(a) \in \tau(G)$, then we can assign a new element $d \in G$ as

$$d \triangleq \tau^{-1}(k \cdot \tau(a)), \tag{8}$$

and call d as the k *quasi-multiple* of a, denoted by $d = k \boxtimes a$.

Then, we can see that the monomorphism τ is linear in sense of that the following equalities hold

$$\begin{aligned}
\tau(k \boxtimes a \boxplus b) &= \tau((k \boxtimes a) \boxplus b) \\
&\overset{d \leftarrow k \boxtimes a}{=} \tau(d \boxplus b) \\
&= \tau(\tau^{-1}(\tau(d) + \tau(b))) \\
&= \tau(\tau^{-1}(\tau(\tau^{-1}(k \cdot \tau(a))) + \tau(b))) \\
&= \tau(\tau^{-1}(k \cdot \tau(a) + \tau(b))) \\
&= k \cdot \tau(a) + \tau(b).
\end{aligned}$$

for $a, b \in G$ and $k \cdot \tau(a) + \tau(b) \in \tau(G)$.

Further, for $f(x) = z_0 + z_1 x + \cdots + z_n x^n \in \mathbb{Z}[x]$ and $a \in G$, if $f(\tau(a)) = z_0 \cdot 1_R + z_1 \cdot \tau(a) + \cdots + z_n \cdot \tau(a)^n \in \tau(G)$, then we can assign a new element $e \in G$ as

$$e \triangleq \tau^{-1}(f(\tau(a))) = \tau^{-1}(z_0 \cdot 1_R + z_1 \cdot \tau(a) + \cdots + z_n \cdot \tau(a)^n), \tag{9}$$

and call e as the *quasi-polynomial* of f on a, denoted by $e = f(a)$.

Clearly, for arbitrary $a, b \in G, k \in R$ and $f(x) \in \mathbb{Z}[x]$, $a \boxplus b, k \boxtimes a$ and $f(a)$ are not always well-defined. But, we can prove that the following theorem holds.

Theorem 6. *For some $a \in G$ and some $f(x), h(x) \in \mathbb{Z}[x]$, if $f(a)$ and $h(a)$ are well-defined, then*

(i) $\tau(f(a)) = f(\tau(a))$*;*
(ii) $f(a) \cdot h(a) = h(a) \cdot f(a)$*.*

Proof. At first, (i) is apparent according to the definition of quasi-polynomial. Next, we have,

$$\begin{aligned}
f(a) \cdot h(a) &= \tau(\tau^{-1}(f(a))) \cdot \tau(\tau^{-1}(h(a))) \quad (\because \tau(\tau^{-1}(g)) = g, g \in G.) \\
&= \tau(\tau^{-1}(f(a)) \cdot \tau^{-1}(h(a))) \quad (\because \tau \text{ is monomorphism.}) \\
&= \tau(\tau^{-1}(f(a) \cdot h(a))) \quad (\because \tau^{-1} \text{ is monomorphism.}) \\
&= \tau(\tau^{-1}(h(a) \cdot f(a))) \quad (\because \text{Theorem } 1) \\
&= \tau(\tau^{-1}(h(a)) \cdot \tau^{-1}(f(a))) \\
&= \tau(\tau^{-1}(h(a))) \cdot \tau(\tau^{-1}(f(a))) \\
&= h(a) \cdot f(a).
\end{aligned}$$

□

5.2 Further Assumptions on Non-commutative Groups

Similar to polynomial version assumptions over a non-commutative ring in Section 4.2, we now consider polynomial version assumption over the non-commutative group G. For any randomly picked element $a \in G$, we define a set $P_a \subseteq G$ by

$$P_a \triangleq \{f(a) \in \tau(G) : f(x) \in \mathbb{Z}[x]\}.$$

Then, we can define the PSD and PDH problems over $(G, \cdot)$ by a similar way:

- **Polynomial Symmetrical Decomposition (PSD) Problem over Non-commutative Group** G: Given $(a, x, y) \in G^3$ and $m, n \in \mathbb{Z}$, find $z \in P_a$ such that $y = z^m x z^n$.
- **Polynomial Diffie-Hellman (PDH) Problem over Non-commutative Group** G: Compute $x^{z_1 z_2}$ (or $x^{z_2 z_1}$) for given a, x, x^{z_1} and x^{z_2}, where $a, x \in G, z_1, z_2 \in P_a$.

Accordingly, the PSD (PDH, respectively) cryptographic assumptions over $(G, \cdot)$ says that PSD (PDH, respectively) problems over $(G, \cdot)$ is intractable, i.e., there does not exist probabilistic polynomial time algorithm which can solve PSD (PDH, respectively) problems over $(G, \cdot)$ with non-negligible accuracy with respect to problem scale.

5.3 Public Key Cryptosystems From Non-commutative Groups

Now, let us take a non-commutative group G with intractable SDP as the underlying work fundamental infrastructure and then restate the Diffie-Hellman-like key exchange protocol in Section 4.3, by a very similar way:

(0) One of the entities (say, Alice) sends two random small, positive integers (say, less than 10) $m, n \in \mathbb{Z}$ and two random elements $a, b \in G$ to another entity (say, Bob) as the signal of launching the protocol.
(1) Alice chooses $f(x) \in \mathbb{Z}[x]$ at random such that $f(a)$ is well-defined, i.e., $f(\tau(a)) \in \tau(G)$. Then, Alice takes $f(a)$ as her private key.
(2) Bob chooses $h(x) \in \mathbb{Z}[x]$ at random such that $h(a)$ is well-defined, i.e., $h(\tau(a)) \in \tau(G)$. Then, Bob takes $h(a)$ as his private key.
(3) Alice computes $r_A = f(a)^m \cdot b \cdot f(a)^n$ and sends r_A to Bob.
(4) Bob computes $r_B = h(a)^m \cdot b \cdot h(a)^n$ and sends r_B to Alice.

Pass	Alice	Bob
	Chooses $m, n \in \mathbb{Z}$ at random Chooses $a, b \in G$ at random Chooses $f(x) \in \mathbb{Z}[x]$ randomly s.t. $f(\tau(a)) \in \tau(G)$	
1	$\xrightarrow{m,n,a,b,f(a)^m b f(a)^n}$	
		Chooses $h(g) \in \mathbb{Z}[x]$ randomly s.t. $h(\tau(a)) \in \tau(G)$
2		$\xleftarrow{h(a)^m b h(a)^n}$
	$K_A = f(a)^m h(a)^m b h(a)^n f(a)^n$ $=$	$K_B = h(a)^m f(a)^m b f(a)^n h(a)^n$

Fig. 2. Diffie-Hellman-Like Key Agreement Based on Non-commutative Groups

(5) Alice computes $K_A = f(a)^m \cdot r_B \cdot f(a)^n$ as the shared session key.
(6) Bob computes $K_B = h(a)^m \cdot r_A \cdot h(a)^n$ as the shared session key.

In practice, the steps (0), (1) and (3) can be finished simultaneously and require only one pass communication from Alice to Bob. After that, the steps (2) and (4) can be finished simultaneously and require another pass communication from Bob to Alice. Finally, Alice and Bob can execute the steps (5) and (6) respectively, needless further communication. Thus, we can depict the protocol in Figure 2.

Similarly, it is easy to describe ElGamal-like encryption schemes, including the basic scheme and the enhanced scheme as well, by using non-commutative groups as the underlying algebraic basis.

[**Basic Scheme**]

- **Initial setup:** Given the non-commutative group $(G, \cdot)$, we assume that SDP on G is intractable. Pick two small positive integers $m, n \in \mathbb{Z}$ and two elements $p, q \in G$ at random. Let $H : G \to \mathcal{M}$ be a cryptographic hash function which maps G to the message space $\mathcal{M}$. Then, set the tuple $< G, m, n, p, q, \mathcal{M}, H >$ as the public parameters of the system.
- **Key generation:** Each user chooses a random polynomial $f(x) \in \mathbb{Z}[x]$ such that $f(\tau(p)) \in \tau(G)$ and takes $sk \triangleq f(p)$ as his private key, then computes and publishes his public key $pk \triangleq f(p)^m \cdot q \cdot f(p)^n \in G$.
- **Encryption:** Given a message $M \in \mathcal{M}$ and receiver's key $pk \in G$, the sender chooses a random polynomial $h(x) \in \mathbb{Z}[x]$ such that $h(\tau(p)) \in \tau(G)$ and takes $h(p)$ as salt, then computes
$$c = h(p)^m \cdot q \cdot h(p)^n, \;\; d = H(h(p)^m \cdot pk \cdot h(p)^n) \oplus M,$$
and finally outputs the ciphertext $(c, d) \in G \times \mathcal{M}$.
- **Decryption:** Upon receiving a ciphertext $(c, d) \in R \times \mathcal{M}$, the receiver, by using his private key $f(p)$, computes the plaintext
$$M = H(f(p)^m \cdot c \cdot f(p)^n) \oplus d$$

[**Enhanced Scheme**]

- **Initial setup:** System public parameters include $G, m, n, p, q, \mathcal{M}$ and k_0, H_1, H_2, where G, m, n, p, q and $\mathcal{M}$ are as same as those in the basic scheme, while the cryptographic hash function H in basic scheme is replaced with two new introduced cryptographic hash functions $H_1 : \{0,1\}^{k+k_0} \to \mathbb{Z}_{>0}[x]$ and $H_2 : G \to \{0,1\}^{k+k_0}$, where k is the standard length of a message, i.e., $\mathcal{M} = \{0,1\}^k$, while k_0 is the length of random salt that should not be determined by binary search method (for example, we set $k_0 = 128$).
- **Key generation:** Identical to the **Key Generation** step in the basic scheme.
- **Encryption:** Given a message $M \in \mathcal{M}$ and receiver's key $pk \in G$, the sender chooses a random salt $r \in \{0,1\}^{k_0}$ and *extracts*[8] a polynomial $h(x) \in \mathbb{Z}[x]$ such that $h(\tau(p)) \in \tau(G)$ and then computes
$$c = h(p)^m \cdot q \cdot h(p)^n, \;\; d = H_2(h(p)^m \cdot pk \cdot h(p)^n) \oplus (M \parallel r),$$
and finally outputs the ciphertext $(c, d) \in R \times \{0,1\}^{k+k_0}$.

[8] See Remark 4 for further explanation.

- **Decryption:** Upon receiving a ciphertext $(c, d) \in R \times \{0,1\}^{k+k_0}$, the receiver, by using his private key $f(p)$, computes

  $$M' = H_2(f(p)^m \cdot c \cdot f(p)^n) \oplus d.$$

  Finally, extracts $g(x) = H_1(M') \in \mathbb{Z}[x]$ and checks whether $c = g(p)^m \cdot q \cdot g(p)^n$ holds. If so, outputs the beginning k bits of M'; otherwise, outputs empty string, which means that the given ciphertext is invalid.

The securities and related proofs of the above cryptosystems are very similar to that of those schemes in Section 4, except replacing the PDH assumption over non-commutative ring R with the PDH assumption over non-commutative group G.

5.4 For Non-commutative Semi-groups

The same construction from Section 5.1 to Section 5.3 can also be considered for the case when G is a non-commutative semi-group, except replacing all appearances of $\mathbb{Z}$ and $\mathbb{Z}[x]$ with $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{>0}[x]$ respectively.

The main differences between the protocols and schemes on non-commutative rings and semi-groups and those on non-commutative groups lie in two aspects:

(1) $m, n \in \mathbb{Z}_{>0}$ on rings and semi-groups while $m, n \in \mathbb{Z}$ on groups;
(2) $f, h \in \mathbb{Z}_{>0}[x]$ on rings and semi-groups while $f, h \in \{g \in \mathbb{Z}[x] : g(\tau(a)) \in \tau(G)\} \subseteq \mathbb{Z}[x]$ on groups.

5.5 More Examples: Public Key Cryptosystems Using Symmetric Groups

Let us illustrate our method by using the group S_3, i.e., the minimal non-commutative group. [9]

At first, we should choose a non-commutative ring as the bridge for definition addable relation over S_3. We choose $M_2(Z_2)$ for convenience.

Next, we should find a monomorphism from S_3 to $M_2(Z_2)$. Let us define $\tau : S_3 \to M_2(Z_2)$ as follows:

$$\begin{pmatrix} 1\,2\,3 \\ 1\,2\,3 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1\,2\,3 \\ 1\,3\,2 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1\,2\,3 \\ 2\,1\,3 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$\begin{pmatrix} 1\,2\,3 \\ 2\,3\,1 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1\,2\,3 \\ 3\,2\,1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

It is not difficult to verify that τ is a monomorphism from S_3 to $M_2(Z_2)$.

Example 3. Diffie-Hellman-Like Key Agreement Using S_3

[9] In practical applications, we should use symmetric groups S_n for some larger n such that the complexity level $\mathcal{O}(n!)$ overwhelms any adversary's computational capability. Here, S_3 and the following employed $M_2(Z_2)$ are too simple to be secure. But here it is enough to use these *toy* examples to demonstrate our method.

Suppose that Alice chooses

$$m = 3, n = 5, A = \begin{pmatrix} 1\,2\,3 \\ 2\,3\,1 \end{pmatrix}, B = \begin{pmatrix} 1\,2\,3 \\ 2\,1\,3 \end{pmatrix},$$

and picks a random polynomial $f(x)$ such that $f(A) \in \tau(S_3)$, assuming that one of such polynomial is

$$f(x) = 4x^2 + x + 2.$$

Then, Alice computes

$$\begin{aligned} f(A) &= \tau^{-1}(f(\tau(A))) \\ &= \tau^{-1}\left(4 \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^2 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} + 2 * I\right) \\ &= \tau^{-1}\left(\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}\right) \\ &= \begin{pmatrix} 1\,2\,3 \\ 2\,3\,1 \end{pmatrix} \end{aligned}$$

and

$$r_A = \begin{pmatrix} 1\,2\,3 \\ 2\,3\,1 \end{pmatrix}^3 \circ \begin{pmatrix} 1\,2\,3 \\ 2\,1\,3 \end{pmatrix} \circ \begin{pmatrix} 1\,2\,3 \\ 2\,3\,1 \end{pmatrix}^5 = \begin{pmatrix} 1\,2\,3 \\ 3\,2\,1 \end{pmatrix}.$$

Then, she sends m, n, A, B and r_A to Bob.

Upon receiving m, n, A, B and $f(A)$ from Alice, Bob chooses another random polynomial $h(x)$ such that $h(A) \in \tau(S_3)$, assuming that one of such polynomial is

$$h(x) = 4x^4 + x^3 + 4x^2 + 3x + 4.$$

Then, Bob computes

$$\begin{aligned} h(A) &= \tau^{-1}(h(\tau(A))) \\ &= \tau^{-1}\left(4 \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^4 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^3 + 4 \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^2 + 3 \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} + 4 * I\right) \\ &= \tau^{-1}\left(\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\right) \\ &= \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix} \end{aligned}$$

and

$$r_B = \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix}^3 \circ \begin{pmatrix} 1\,2\,3 \\ 2\,1\,3 \end{pmatrix} \circ \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix}^5 = \begin{pmatrix} 1\,2\,3 \\ 1\,3\,2 \end{pmatrix}.$$

Then, he sends r_B to Alice.

Finally, Alice extracts the session key

$$K_A = \begin{pmatrix} 1\,2\,3 \\ 2\,3\,1 \end{pmatrix}^3 \circ \begin{pmatrix} 1\,2\,3 \\ 1\,3\,2 \end{pmatrix} \circ \begin{pmatrix} 1\,2\,3 \\ 2\,3\,1 \end{pmatrix}^5 = \begin{pmatrix} 1\,2\,3 \\ 2\,1\,3 \end{pmatrix},$$

while Bob extracts the session key

$$K_B = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}^3 \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}^5 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

Apparently, $K_A = K_B$ holds, i.e., the key agreement is successful.

Example 4. Encryption/Decryption Using S_3

We can implement encryption/decryption by ways which are very similar to those of in Example 2, i.e., to encrypt/decrypt step by step according to the basic scheme and the enhanced scheme defined in Section 5.3.

At first, let us choose two prime p and q such that $q|p-1$ and set Z_p as the message space $\mathcal{M}$. Also, we assume that g is a generator of order q. Then, we define

$$H : S_3 \to \mathcal{M},\ \begin{pmatrix} 1 & 2 & 3 \\ \sigma_1 & \sigma_2 & \sigma_3 \end{pmatrix} \mapsto g^{\sigma_1 + 2\cdot\sigma_2 + 2^2\cdot\sigma_3} \mod p$$

and

$$H_1 : \mathcal{M} \times Z_N \to \mathbb{Z}_{>0}[x], (M, r) \mapsto r_0 + r_1 x + r_2 x^2 + \cdots + r_n x^n,$$

where all coefficients $r_i, 0 \le i \le k$, is determined by the following process: [10]:

$$\begin{aligned} M &= rk_0 + r_0,\ 0 < r_0 < r, \\ r &= r_0 k_1 + r_1,\ 0 < r_1 < r_0, \\ r_0 &= r_1 k_2 + r_2,\ 0 < r_2 < r_1, \\ \cdots\ & \quad \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots \\ r_{l-2} &= r_{l-1} k_l + r_l,\ 0 < r_l < r_{l-1}, \\ \cdots\ & \quad \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots \\ r_{n-1} &= r_n k_{n+1} + r_{n+1},\ 0 < r_{n+1} = 0. \end{aligned}$$

For simplicity, we define [11]

$$M \parallel r \triangleq r \cdot p + M \in \mathbb{Z}$$

and can extract M from $M \parallel r$ by

$$M = (M \parallel r) \mod p.$$

Another required hash $H_2 : S_3 \to \mathcal{M} \times Z_N$ can be defined as

$$H_2 : S_3 \to \mathcal{M} \times Z_p,\ \begin{pmatrix} 1 & 2 & 3 \\ \sigma_1 & \sigma_2 & \sigma_3 \end{pmatrix} \mapsto (g^{\sigma_1 + 2\cdot\sigma_2 + 2^2\cdot\sigma_3} \mod p,\ g^{\sigma_3 + 2\cdot\sigma_1 + 2^2\cdot\sigma_2} \mod p).$$

[10] Here, we assume that $\gcd(M, r) \ne r$; Otherwise, we set $r = g^r \mod p$ and then resume the process. Also, we assume that $M > r$; Otherwise, we can swap them in advance.

[11] Of course, this leads to minor expanding of ciphertext.

Next, let $p = 23, q = 11$ and $g = 2$ for simplifying computation and verification. Suppose that the left system parameters are

$$m = 3, n = 5, p = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, q = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

[Encryption/Decryption with Basic Scheme]

Suppose that the polynomial $f(x)$ picked by Alice is just that of in example 3. Then, Alice' private key is $f(p) = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{smallmatrix}\right)$, just as $f(A)$ in Example 3. Then, the corresponding public key would be

$$pk \triangleq f(p)^3 q f(p)^5 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}^3 \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}^5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix},$$

just as r_A in Example 1.

Let us pick a message M randomly, say $M = 17$. Suppose the salt polynomial we picked randomly is coincide to $h(x)$ in Example 3. Then, the salt permutation

$$h(p) = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix},$$

just as $h(A)$ in Example 3. Now, let us compute the ciphertext (c, d) as follows:

$$c = h(p)^3 q h(p)^5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}^3 \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}^5 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix},$$

and

$$\begin{aligned} d &= H(h(p)^3 \cdot pk \cdot h(p)^5) \oplus M \\ &= H\left(\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}^3 \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}^5\right) \oplus 17 \\ &= H\left(\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}\right) \oplus 17 \\ &= \left(2^2 + 2^{2 \cdot 1} + 2^{2^2 \cdot 3} \mod 23\right) \oplus 17 \\ &= 18 \oplus 17 \\ &= 3. \end{aligned}$$

Now, let us check the decryption process:

$$\begin{aligned} M' &= H(f(p)^3 \cdot c \cdot f(p)^5) \oplus d \\ &= H\left(\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}^3 \circ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}^5\right) \oplus 3 \\ &= H\left(\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}\right) \oplus 3 \\ &= \left(2^2 + 2^{2 \cdot 1} + 2^{2^2 \cdot 3} \mod 23\right) \oplus 3 \\ &= 18 \oplus 3 \\ &= 17 \\ &= M. \end{aligned}$$

[Encryption/Decryption with Enhanced Scheme]

Suppose that the private/public keys are unchanged. Let us pick a message M randomly, say $M = 19$. Suppose the salt number we picked randomly is $r = 7$. Then, we extract a polynomial as follows:

$$\begin{aligned}
\because \quad 19 &= 7 \cdot 2 + 5 \\
7 &= 5 \cdot 1 + 2 \\
5 &= 2 \cdot 2 + 1 \\
2 &= 2 \cdot 1 + 0 \\
\therefore h(x) &= 5 + 2x + x^2 + x^3.
\end{aligned}$$

Note that if $h(x)$ does not satisfy the condition of $h(\tau(p)) \in \tau(S_3)$, we should at first rectify $h(x)$ to $\tilde{h}(x) = h(x) + \varDelta$, where

$$\varDelta = \min\left\{\delta \in \mathbb{Z}_{\geq 0} : h(\tau(p)) + \delta \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in \tau(S_3)\right\}.$$

Fortunately, in this example. The above extracted $h(x)$ meets the requirement of $h(\tau(p)) \in \tau(S_3)$, i.e., $\varDelta = 0$. Thus,

$$\begin{aligned}
h(p) &= \tau^{-1}(h(\tau(p))) \\
&= \tau^{-1}\left(5 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 2 \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^2 + \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^3\right) \\
&= \tau^{-1}\left(\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\right) \\
&= \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix}.
\end{aligned}$$

Consequently, the corresponding ciphertext pair (c, d) would be

$$c = h(p)^3 q h(p)^5 = \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix}^3 \circ \begin{pmatrix} 1\,2\,3 \\ 2\,1\,3 \end{pmatrix} \circ \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix}^5 = \begin{pmatrix} 1\,2\,3 \\ 1\,3\,2 \end{pmatrix}$$

and

$$\begin{aligned}
d &= H(h(p)^3 \cdot pk \cdot h(p)^5) \oplus (M \parallel r) \\
&= H\left(\begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix}^3 \circ \begin{pmatrix} 1\,2\,3 \\ 3\,2\,1 \end{pmatrix} \circ \begin{pmatrix} 1\,2\,3 \\ 3\,1\,2 \end{pmatrix}^5\right) \oplus (19 + 5 \cdot 23) \\
&= H\left(\begin{pmatrix} 1\,2\,3 \\ 2\,1\,3 \end{pmatrix}\right) \oplus 134 \\
&= \left(2^2 + 2^{2 \cdot 1} + 2^{2^2 \cdot 3} \mod 23\right) \oplus 134 \\
&= 18 \oplus 134 \\
&= 148.
\end{aligned}$$

Now, let us check the enhanced decryption process:

$$\begin{aligned}
M' &= H(f(p)^3 \cdot c \cdot f(p)^5) \oplus d \\
&= H\left(\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}^3 \circ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}^5\right) \oplus 148 \\
&= H\left(\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}\right) \oplus 3 \\
&= \left(2^2 + 2^{2\cdot 1} + 2^{2^2\cdot 3} \mod 23\right) \oplus 148 \\
&= 18 \oplus 148 \\
&= 134.
\end{aligned}$$

Then, $M = M' \mod 23 = 19$.

6 Conclusions

Recently, some promising build public-key cryptosystems have been constructed on non-commutative groups, such as braid groups, Thompson's groups, etc. In this paper, we described a totally different method for designing PKC based on general non-commutative algebraic systems, including non-commutative rings, non-commutative groups and non-commutative semi-groups as well. The key ideas behind our proposal lies that we take polynomials over the given non-commutative algebraic systems as the the underlying work structure for constructing cryptographic schemes. By doing so, we can efficiently obtain some commutative sub-structures for the given non-commutative algebraic systems. The security assumption is that the proposed polynomial Diffie-Hellman (PDH) problem over the given non-commutative algebraic systems is intractable.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 60225007, 60572155 and 60673079. We would also like to thank Peng Zeng and Haifeng Qian for their valuable discussion and suggestion.

References

1. M. Anshel. Braid Group Cryptography and Quantum Cryptoanalysis. 8th International Wigner Symposium May 27-30, 2003 GSUC-CUNY 365 Fifth Avenue, NY, NY 10016, USA. 1.2
2. I. Anshel, M. Anshel, and D. Goldfeld. An algebraic method for public-key cryptography. Math. Research Letters 6 (1999) 287-291. 1.2
3. I. Anshel, M. Anshel, and D. Goldfeld. A linear time matrix key agreement protocol over small finite fields. AAECC (2006). 1.2
4. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, Relations among notions of security for public-key encryption schemes. In Advances in Cryptology - Crypto '98, LNCS 1462, Springer Verlag, Berlin , 1998, pp.26-45. 1
5. M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols. In First ACM Conference on Computer and Communications Security, pages 62-73, Fairfax, 1993. ACM. 1

6. M. Bellare and P. Rogaway, Optimal Asymmetric Encryption - How to Encrypt with RSA, In Advance in Cryptology - Eurocrypt '94, LNCS 950, Springer-Verlag, Berlin, 1995, pp.92-111. 1
7. J. Birget, S.S. Magliveras and M. Sramka. On public-key cryptosystems based on combinatorial group theory. Cryptology ePrint Archive: Report 2005/070, 2005. 1.2
8. J. Birman, K.H. Ko, and S.J. Lee, A new approach to the word and conjugacy problems in the braid groups, Adv. Math. 139 (1998), 322-353. 1.2
9. J. Birman, K.H. Ko, and S.J. Lee, The inimum, supremum, and geodesic length of a braid conjugacy class, Adv. Math. 164 (2001), 41-56. 1.2
10. J.-M. Bohli, B. Glas and R. Steinwandt. Towards provable secure group key agreement building on group theory. Cryptology ePrint Archive: Report 2006/079, 2006. 1.2
11. D. Boneh, Simplified OAEP for the RSAS and Rabin Functions, In Advances in Cryptology - Crypto '01, LNCS 2139, Springer-Verlag, Berlin, 2001, pp.275-291. 1
12. D. Boneh, and M. Franklin, Identity-based Encryption from the Weil pairing, SIAM. Journal of Computing, vol. 32 (3), 2003, 586-615. Extended abstract in Advances in Crptology-Crypto'01, LNCS 2139, Berlin, Springer-Verlag, 2001, pp. 213-229. 1
13. Z. Cao. Conic analog of RSA cryptosystem and some improved RSA cryptosystems. Journal of Natrual Science of Heilongjiang University, 16 (4) 1999. 1.1
14. Z. Cao. The multi-dimension RSA and its low exponent security. Science in China Series E, 43 (4): 349-354, 2000. 1.1
15. Z. Cao. A threshold key escrow scheme based on public key cryptosystem. Science in China (E Series), 44 (4): 441-448, 2001. 1.1
16. J.C. Cha, K.H. Ko, S.J. Lee and J.W. Han *et al.* An Efficient Implementation of Braid Groups. C. Boyd (Ed.): ASIACRYPTO 2001, LNCS 2248, pp. 144-156, Springer-Verlag, 2001. 1.2
17. J.H. Cheon and B. Jun. A Polynomial Time Algorithm for the Braid Diffie-Hellman Conjugacy Problem. In D. Boneh (Ed.): CRYPTO 2003, LNCS 2729, pp. 212C225, Springer-Verlag, 2003. 1.2
18. R. Cramer,and V. Shoup, A Practical Public key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attacks, In Advances in Cryptology - Crypto'98, LNCS 1462, Springer-Verlag, 1998, pp.13-25. 1
19. P. Dehornoy. Braid-based cryptography. *Contemporary Mathematics*, 360 (2004) 5-33. 1.2, 3.3
20. W. Diffie and M.E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory 22 (1976), 644-654. 1.1, 4.3
21. B. Eick and D. Kahrobaei. Polycyclic groups: a new platform for cryptography. Preprint arXiv:math.GR/0411077, 2004. 1.2
22. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31 (1985), 469-472. 1.1
23. E.A. El-Rifai, H.R. Morton, Algorithms for positive braids, Quart. J. Math. Oxford Ser. (2) 45 (1994), 479-497. 1.2
24. E. Fujisaki and T. Okamoto, How to Enhance the Security of Public Key Encryption at Minimum Cost, In H. Imai and Y. Zheng (Eds.): PKC'99, LNCS 1560, pp. 53-68, Springer-Verlag, 1999. 1, 4.4, 4.4, 4, 4.4
25. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encrption Schemes, In M. Wiener (Ed.): CRYPTO'99, LNCS 1666, pp. 537-554, Springer-Verlag, 1999. 4.4
26. V. Gebhardt, A new approach to the conjugacy problem in Garside groups, Preprint arxiv.org/math.GT/0306199, 2003. 1.2
27. S. Goldwasser and S. Micali, Probabilistic Encryption, Journal of Computer and System Sciences, 28:270-299, 1984. 2.2
28. J. Gonzales-Meneses, Improving an algorithm to solve the Multiple Simultaneous Conjugacy Problems in braid groups, Preprint arxiv.org/abs/math.GT/0212150, 2002. 1.2
29. D. Grigoriev and I. Ponomarenko. On non-abelian homomorphic public-key cryptosystems. Preprint arXiv:cs.CR/0207079, 2002. 1.2

30. D. Grigoriev and I. Ponomarenko. Homomorphic public-key cryptosystems over groups and rings. Preprint arXiv:cs.CR/0309010, 2003. 1.2
31. J. Hughes, The left SSS attack on Ko-Lee-Cheon-Han-Kang-Park key agreement scheme in B_{45}, Rump session CRYPTO 2000. 1.2
32. J. Hughes, A linear algebraic attack on the AAFG1 braid group cryptosystem, ACISP 2002; LNCS 2384, pp. 176-189, Springer-Verlag, 2002. 1.2
33. J. Hughes and A. Tannenbaum, Length-based attacks for certain group based encryption rewriting systems, Inst. for Mathematics and Its Applic. http://www.ima.umn.edu/preprints/apr2000/1696.pdf, 2000. 1.2
34. A. Kitaev. Quantum measurements and the Abelian Stabilizer Problem. Preprint arXir:cs.CR/quant-ph/9511026, 1995. 1.2
35. K.H. Ko, D.H. Choi, M.S. Cho and J.W. Lee. New signature scheme using conjugacy problem. Cryptology ePrint Archive: Report 2002/168, 2002. 1.2, 3.1, 3.3
36. K.H. Ko, S.J. Lee, J.H. Cheon and J.W. Han *et al.* New Public-Key Cryptosystem Using Braid Groups. M.Bellare (Ed.): CRYPTO 2000, LNCS 1880, pp. 166-183, Springer-Verlag, 2000. 1.2
37. K. Komaya, U. Maurer, T. Okamoto and S. Vanston, Newpublic-key schemes bases on elliptic curves over the ring Zn, Advances in Cryptology, Proceedings of Crypto 91, Lecture Notes in Computer Science 576, ed. J. Feigenbaum, Springer-Verlag (1992), 252C266 1.1
38. E. Lee. Braig groups in cryptography. IEICE Trans. Fundamentals, Vol. E87-A, No. 5, May 2004: 986-992. 1.2
39. S.J. Lee and E.K. Lee. Potential weakness of the commutator key agreement protocol based on braid groups. EUROCRYPT 2002, LNCS 2332, pp. 14-28, Springer-Verlag, 2002. 1.2
40. E. Lee, S.J. Lee and S.G. Hahn. Pseudorandomness from Braid groups. In J. Kilian (Ed.): CRYPTO 2001, LNCS 2139, pp. 486-502, 2001. 1.2
41. S.S. Magliveras, D.R. Stinson, and T. van Trung. New Approaches to Designing Public Key Cryptosystems Using One-Way Functions and Trapdoors in Finite Groups. Technical Report CORR 2000-49, Centre for Applied Cryptographic Research, University of Waterloo. http://www.cacr.math.uwaterloo.ca/ techreports/2000/corr2000-49.ps 1.2
42. S.S. Magliveras, D.R. Stinson and T. van Trung. New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups. J. Cryptography (2002) 15: 285-297. 1.2
43. J. Manuel, G. Nieto, C. Boyd, E. Dawson, A public key cryptosystem based on a subgroup membership problem, Designs, Codes and Cryptography, 36, 301-316, 2005. 2.1
44. W. Mao, Modern Cryptography: Theory and Practice. Publishing House of Electronics Industry, Beijing: 2004. 4.4
45. K. McCurley. A key distribution system equivalent to factoring, Journal of Cryptology 1 (1988), 95-100. 1.1
46. A. Myasnikov, V. Shpilrain, A. Ushakov. A Practical Attack on a Braid Group Based Cryptographic Protocol. In V. Shoup (Ed.): Crypto 2005, LNCS 3621, pp. 86-96, Springer-Verlag, 2005. 1.2
47. M. Naor, and M. Yung, Public-Key Cryptosystem Provably Secure against Chosen Ciphertext Attacks, In Proc. of the 22nd STOC, ACM Press, New York, 1990, pp.427-437. 2.2
48. T. Okamoto and D. Pointcheval, REACT: Rapid Enhanced-security Asymetric Cryptosystem Transform, IN CT-RSA '2001, LNCS 2020, Springer-Verlag, Berlin, 2001, pp.159-175. 1
49. S.-H. Paeng, K.-C. Ha, J.-H. Kim, S. Chee and C. Park. New public key cryptosystem using finite Non Abelian Groups. In J. Kilian (Ed.): CRYPTO 2001, LNCS 2139, pp. 470-485, Springer-Verlag, 2001. 1.2
50. S.-H. Paeng, D. Kwon, K.-C. Ha, and J. H. Kim. Improved public key cryptosystem using finite non abelian groups. Cryptology ePrint Archive: Report 2001/066, 2001. 1.2
51. J. Proos and C. Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. Quantum Information and Computation 3 (2003): 317-344. 1.2
52. M.O. Rabin, Digitized signatures and public-key functions as intractible as factorization, MIT Laboratory for Computer Science Technical Report, LCS/TR-212 (1979). 1.1

53. C. Rackoff, and D. Simon, Non-Interactive Zero-Knowledge Proof of KNowledge and Chosen Ciphertext Attack. IN Advances in Cryptology - CRYPTO '91, LNCS 576, Springer-Verlag, 1992, pp.433-444. 2.2
54. R.L. Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM 21 (1978), 120-126. 1.1
55. A. Shamir, Identity-based cryptosystems and signature schemes, In Advances in Cryptology - Crypto'84, LNCS 196, Berlin, Springer-Verlag, 1984, pp.47-53. 1
56. P. Shor. Polynomail-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 5 (1997): 1484-1509. 1.2
57. V. Shpilrain and A. Ushakov. Thmposon's group and public key cryptography. Preprint arXir:math.GR/0505487, 2005. 1.2
58. V. Shpilrain and A. Ushakov. A new key exchange protocol based on the decomposition problem. Preprint arXiv:math.GR/0512140, 2005. 1.3
59. V. Shpilrain and G. Zapata. Combinatorial group theory and public key cryptography. Preprint arXir:math.GR/0410068, 2004. 3.3
60. P. Smith and M. Lennon, LUC: A newpublic key system, Proceedings of the IFIP TC11 Ninth International Conference on Information Security, ed. E. Dougall, IFIP/Sec 93, 103-117, North-Holland, 1993. 1.1
61. T. Thomas, A.K. Lal. Undeniable Signature Schemes Using Braid Groups. Preprint arXiv:cs.CR/0601049, 2006. 1.2
62. T. Thomas, A.K. Lal. Group Signature Schemes Using Braid Groups. Preprint arXiv:cs.CR/0602063, 2006. 1.2
63. M. Vasco, C. Martinez and R. Steinwandtd. Towards a uniform description of several group based cryptographic primitives. Cryptology ePrint Archive: Report 2002/048, 2002. 1.2
64. N.R. Wagner, M.R. Magyarik. A public-key cryptosystem based on the word problem. In G.R. Blakley and D.Chaum (Eds): CRYPTO'84, LNCS 196, pp. 19-36, Springer-Verlag, 1985. 1.2
65. H.C. Williams. A Modification of the RSA Public-Key Encryption Procedure. IEEE Transactions on Information Theory, IT No.6 (26):726-729, November 1980. 1.1
66. H.C. Williams. Some public-key crypto-funtions as intractible as factorization. In G.R. Blakley and D.Chaum (Eds): CRYPTO'84, LNCS 196, pp. 66-70, Springer-Verlag, 1985. 1.1