New Public Key Cryptosystem Using Finite Non Abelian Groups

Seong-Hun Paeng, Kil-Chan Ha,
Jae Heon Kim, Seongtaek Chee, and Choonsik Park

National Security Research Institute
161 Kajong-dong, Yusong-gu, Taejon, 305-350, KOREA
{shpaeng,kcha,jaeheon,chee,csp}@etri.re.kr

Abstract. Most public key cryptosystems have been constructed based on abelian groups up to now. We propose a new public key cryptosystem built on finite non abelian groups in this paper. It is convertible to a scheme in which the encryption and decryption are much faster than other well-known public key cryptosystems, even without no message expansion.
Furthermore a signature scheme can be easily derived from it, while it is difficult to find a signature scheme using a non abelian group.

1 Introduction

Most frequently used problems in the public key cryptosystems are the factorization problem [19] and the discrete logarithm problem (DLP). Cryptosystems based on these problems have been built on abelian groups [5,3,8,12,13]. In Crypto 2000, Ko et al. proposed a new public cryptosystem based on Braid groups, which are non abelian groups. To authors' best knowledge, it was the first practical public key cryptosystem based on non abelian groups.

When we use a non abelian group G for a public key cryptosystem, we need to consider the following problems related to the word problem:

- How do we express a message as an element of G?
- Can every element of G be represented in a unique way for a given expression?

If an element of G is not represented in a unique way, then a plaintext and a deciphertext may not be the same. Therefore the second problem is very important when we use a non abelian group for a public key system. Matrix groups and semi-direct product of abelian groups are examples of non abelian groups which have such expressions.

In this paper, we suggest a new cryptosystem based on such a finite non abelian group G. Our PKC is based on DLP in the inner automorphism group

$$Inn(G) = \{Inn(g) \mid g \in G\},$$

where $Inn(g)(x) = gxg^{-1}$. The advantages of our PKC are as follows:

J. Kilian (Ed.): CRYPTO 2001, LNCS 2139, pp. 470–485, 2001.

- We can apply our encryption scheme to G even if DLP and the (special) conjugacy problem in G are not hard problems.
- Parameter selections are much easier than those in ECC [12,13] and XTR [8].
- We can increase the speed of the encryption and decryption. More precisely, when m is a message and g^a is the public key in ElGamal-type encryption [5,12,13], $(g^{ab}m, g^b)$ should be sent to a receiver and it is crucial that different random integers b should be used to encrypt different messages. In our scheme, we can fix b without loss of security so that we can increase the speed of the encryption and decryption. Moreover no message expansion is required.
- It is easy to make a signature scheme with our PKC: In general, it is not easy to find a signature scheme using an infinite non abelian group such as a braid group [11].

If we fix b, our PKC is about 30 times faster than RSA for a 32-bit public exponent in RSA encryption scheme and is about 200 times faster in decryption.

2 Preliminaries

2.1 Semi-direct Product

From some given groups, we can easily make new non abelian groups using semi direct products. Recall the definition of the semi-direct product:

Definition 1. *(Semi-direct product) Let G and H be given groups and $\theta : H \to Aut(G)$ be a homomorphism, where $Aut(G)$ is the automorphism group of G. Then semi-direct product $G \times_\theta H$ is the set*

$$G \times H = \{(g, h) |\ g \in G,\ h \in H\}$$

together with the multiplication map

$$(g_1, h_1) \cdot (g_2, h_2) = (g_1 \theta(h_1)(g_2), h_1 h_2).$$

Since

$$(g, h)^{-1} = (\theta(h^{-1})(g^{-1}), h^{-1}),$$

we have

$$(e_G, h_1)(g_2, e_H)(e_G, h_1)^{-1} = (\theta(h_1)(g_2), e_H), \tag{2.1}$$

where e_G, e_H are the identity elements of G and H, respectively. So G can be considered as a normal subgroup of $G \times_\theta H$. If $\theta(H) \neq Id$, then $G \times_\theta H$ is a non abelian group even if G and H are abelian.

Example 1. (1) Most familiar example of the semi-direct product is the isometry group on Euclidean space $\mathbb{R}^n$. This group is the semi-direct product of the translational isometry group $\mathbb{R}^n$ and the orthogonal group $O(n, \mathbb{R})$.

(2) It is a well known fact that $\mathrm{Aut}(\mathbb{Z}_n) = \mathbb{Z}_n^*$, where $\mathbb{Z}_n^*$ is the multiplicative group of $\mathbb{Z}_n$. Since $\mathbb{Z}_4^* \simeq \mathbb{Z}_2$, there exists a non constant homomorphism, in fact, an isomorphism of $\mathbb{Z}_2$ into $\mathbb{Z}_4^*$. Thus $\mathbb{Z}_4 \times_\theta \mathbb{Z}_2$ is a non abelian group.
(3) If G is a non abelian group, then there exists a natural homomorphism from G to $\mathrm{Aut}(G)$. Precisely,

$$\begin{aligned} Inn : G &\to \mathrm{Aut}(G) \\ g &\mapsto Inn(g),\ Inn(g)(h) = ghg^{-1}. \end{aligned} \tag{2.2}$$

We call $Inn(g)$ an inner automorphism. It is easy to check that $\ker(Inn)$ is the center of G. Recall that the center of G is the set $\{z|\ [z,g] = zgz^{-1}g^{-1} = e_G \text{ for all } g \in G\}$. If G is an abelian group, Inn is a constant map and so $G \times_{Inn} G = G \times G$. But if G is a non abelian group, then $G \times_{Inn} G$ is an interesting extension of G.

If we apply a semi-direct product to p-groups inductively, we can make a nilpotent group [7]. It is a well known fact that nilpotent groups can be expressed in a unique way as a direct product of abelian groups. The above $\mathbb{Z}_4 \times_\theta \mathbb{Z}_2$ in Example 1. (1) is a nilpotent group with order 8.

2.2 Conjugacy Problem

One of the most important characteristics of non abelian groups distinguished from abelian groups is that Inn is not constant, i.e. there exist two distinct elements which are congruent to each other.

Definition 2. (1) *For arbitrary $x, y \in G$, the conjugacy problem is to find $w \in G$ such that $wxw^{-1} = y$.*
(2) *For a given $Inn(g)$, the special conjugacy problem is to find g' satisfying $Inn(g') = Inn(g)$.*

There are many groups where the word problem is known to be solvable in polynomial time while there is no known polynomial time algorithm to solve the conjugacy problem (the braid group is an example) [1]. If G is a non abelian group and its conjugacy problem is hard in G, we can consider the following cryptosystem. Let $\{\delta_i\}$ be a set of generators of G. Let g be an element of G. The public key is $\{\epsilon_i = g\delta_i g^{-1}\}$ and the secret key is g. Mathematically, the public key can be expressed as $Inn(g)$. Then the ciphertext is $E = Inn(g)(m)$ and the deciphertext is $g^{-1}Eg$ ([1] or [20]). In order to use such an encryption scheme, every element of G should be easily expressible as a product of δ_i's. If an element of G is also easily expressible as a product of ϵ_i's, then we also obtain $Inn(g^{-1})$ immediately. Since $g^{-1}Eg = Inn(g^{-1})(E)$, we can decrypt without knowing g. Thus it is essential that elements of G should not be easily expressible as products of ϵ_i's.

This system depends on the difficulty of finding g' satisfying $Inn(g') = Inn(g)$ for a given $Inn(g)$, i.e. the above system is based on the special conjugacy problem. Unfortunately, we know few finite non abelian groups to which

we can apply the above system. For example, the general linear group $\mathrm{GL}(2,\mathbb{Z}_p)$ and the special linear group $\mathrm{SL}(2,\mathbb{Z}_p)$ are non abelian groups on which the (special) conjugacy problem is not difficult (see Appendix A).

Remark 1. If we use DLP in $\mathrm{SL}(2,\mathbb{Z}_p)$, we choose $g \in \mathrm{SL}(2,\mathbb{Z}_p)$ whose order is divided by p. The order of $\mathrm{SL}(2,\mathbb{Z}_p)$ is $|\mathrm{SL}(2,\mathbb{Z}_p)| = p(p-1)(p+1)$. Such elements which we are aware of are the conjugates of $I + c\delta_{12}$ and $I + c\delta_{21}$, where $c \in \mathbb{Z}_p$ and δ_{ij} is a matrix whose entries are all zero except the (i,j)-entry which is 1. Let $g = A(I + \delta_{12})A^{-1}$ for

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}(2,\mathbb{Z}_p).$$

Then we have

$$\begin{aligned} g^m &= A(I + m\delta_{12})A^{-1} \\ &= \begin{pmatrix} ad - bc - mac & ma^2 \\ -mc^2 & ad - bc + mac \end{pmatrix} = \begin{pmatrix} 1 - mac & ma^2 \\ -mc^2 & 1 + mac \end{pmatrix}. \end{aligned} \tag{2.3}$$

Consider DLP in the cyclic group $\langle g^m \rangle$. Since $(1,2)$-component of g^m and g^{ml} are ma^2 and mla^2, respectively, we can obtain $l \mod p$. Hence DLP in $\langle g^m \rangle$ is not a hard problem. The (special) conjugacy problem and DLP are not hard problems in $\mathrm{SL}(2,\mathbb{Z}_p)$.

3 New Cryptosystem

In this section, we suggest a new encryption scheme which are based on DLP in the inner automorphism group.

Let G be a non abelian group with non trivial center $Z(G)$. We assume that $Z(G)$ is not small. Let g be an element of G.

Proposed encryption scheme. Let $\{\gamma_i\}$ be a set of generators of G. Since $Inn(g)$ is a homomorphism, $Inn(g)$ is obtained if we know $Inn(g)(\gamma_i)$, i.e. if we express m as $\gamma_{j_1} \cdots \gamma_{j_n}$, then $Inn(g)(m) = Inn(g)(\gamma_{j_1}) \cdots Inn(g)(\gamma_{j_n})$. Therefore we can represent $Inn(g)$ by $\{Inn(g)(\gamma_i)\}$. The basic scheme is the following:

- public key : $Inn(g)$, $Inn(g^a)$
- secret key : a

Encryption

1. Alice expresses a plaintext $m \in G$ as a product of γ_i's.
2. Alice chooses an arbitrary b and computes $(Inn(g^a))^b$, i.e. $\{(Inn(g^a))^b(\gamma_i)\}$.
3. Alice computes $E = Inn(g^{ab})(m) = (Inn(g^a))^b(m)$.
4. Alice computes $\phi = Inn(g)^b$, i.e. $\{Inn(g^b)(\gamma_i)\}$.
5. Alice sends (E, ϕ).

Decryption

1. Bob expresses E as a product of γ_i's.
2. Bob computes ϕ^{-a}, i.e. $\{\phi^{-a}(\gamma_i)\}$.
3. Bob computes $\phi^{-a}(E)$.

To implement our scheme, we should express $Inn(g^a)$ with small bits. Since G is a finitely generated group, $Inn(g^a)$ is expressed by $\{Inn(g^a)(\gamma_i)\}$ for a generator set $\{\gamma_i\}$. If we do not have a fast algorithm to express $\gamma \in G$ by a product of generators, we cannot express $Inn(g^a)$ actually. In the next section, we will introduce a non abelian group to which our scheme can be applied. (Precisely, see 4.3.)

Although our scheme looks like an ElGamal-type, we may not change b for each encryption. In ElGamal-type encryption based on abelian groups (e.g. ECC), we must change b for each encryption. (If a fixed b is used, we can obtain $m_1^{-1}m_2 = (m_1 g^{ab})^{-1} m_2 g^{ab}$).) But in our scheme, it is impossible to obtain $m_1^{-1}m_2$ from $Inn(g^b)(m_1)$ and $Inn(g^b)(m_2)$. Thus we may fix b. As we see in section 4.3, this fact will be very useful for fast encryption and decryption scheme.

Due to the non commutativity of braid groups, the cryptosystem using braid groups has a difficulty in making a signature scheme. However, our scheme enables us to make a signature scheme easily (e.g. Nyberg-Rueppel type signature) even if G is non abelian.

Now we consider the method to find a from the given $Inn(g)$ and $Inn(g^a)$. First, we solve DLP in $\langle Inn(g)\rangle$ directly. The index calculus is the most efficient known method to solve DLP [4]. But its application is too restrictive to be applied to general cyclic groups. It seems that the index calculus cannot be applied to the group $\langle Inn(g)\rangle$. In general cases, expected run times for solving DLP are $O(\sqrt{p})$, where p is the order of a cyclic subgroup.

Secondly, we solve DLP in $\langle g\rangle$. If we assume that the special conjugacy problem is not a hard problem, we can find g_0 satisfying $Inn(g_0) = Inn(g^a)$. We can easily verify that $g_0 = g^a z$ for some $z \in Z(G)$. If $|Z(G)|$ is large enough, then it is almost impossible to determine whether $g^a z$ is an element of $\langle g\rangle$. Then even if DLP in G may be easy, we cannot apply any algorithm to solve DLP in G.

We should be careful in the choice of a plaintext and g. If $[m, g] = e_G$, then $E = g^{ab} m g^{-ab} = m$. In particular, if m is a central element, then $E = m$ so m should not be chosen in the center. Also if g is a central element, then $Inn(g)$ is the identity map and so $E = m$. We should select a non central element g.

We should note that there may be other attacks depending on G as we see in section 5.

Remark 2. In the above scheme, $E = Inn(g^{ab})(m)$, E and m are contained in the same conjugacy class. Assume that E is a ciphertext of either m_0 or m_1, which are not contained in the same conjugacy class. Then an adversary can find the right plaintext by examining the conjugacy class of E. To avoid this attack, we can use a padding method in the encryption (see Remark 4 and [16]). It also makes fast encryption and decryption scheme (which fixes b) non deterministic.

4 Construction of a Non Abelian Group

4.1 An Example of Non Abelian Group $\mathbf{SL(2,\mathbb{Z}_p)\times_\theta \mathbb{Z}_p}$

If we use the semi-direct product, we can construct many non abelian groups with non trivial center as in section 5. But it is not easy to construct a non abelian group on which our system is secure. We modify $\mathrm{SL}(2,\mathbb{Z}_p)$ by a semi-direct product as follows. There exists a cyclic subgroup $\langle\alpha\rangle$ with order p of $\mathrm{SL}(2,\mathbb{Z}_p)$, e.g. $I+\delta_{12}$. Let

$$G = \mathrm{SL}(2,\mathbb{Z}_p)\times_\theta \mathbb{Z}_p,$$

where

$$\theta = Inn\circ\theta_1 : \mathbb{Z}_p \to \mathrm{Aut}(\mathrm{SL}(2,\mathbb{Z}_p))$$

and θ_1 is an isomorphism from $\mathbb{Z}_p$ to $\langle\alpha\rangle$. Then $\theta(y)(x) = \theta_1(y)x\theta_1(y)^{-1}$. Now we solve the conjugacy equations in G. Let $g=(x,y)$. Computing the conjugate of (a,b), we obtain that

$$(x,y)(a,b)(x,y)^{-1} = (x\theta(y)(a)\theta(b)(x^{-1}),b). \tag{4.4}$$

If $b=0$, we have

$$(x,y)(a,0)(x,y)^{-1} = (x\theta(y)(a)x^{-1},0) = ((x\theta_1(y))a(x\theta_1(y))^{-1},0). \tag{4.5}$$

If we solve the special conjugacy problem in $\mathrm{SL}(2,\mathbb{Z}_p)$ as we see in Appendix A, we can obtain $x\theta_1(y)$. Let $(x_1,y_1)\in G$ such that $x_1\theta_1(y_1) = x\theta_1(y)$. For $b\neq 0$, if we use the fact that $\mathbb{Z}_p$ is an abelian group and θ_1 is a homomorphism, we can easily verify that

$$\begin{aligned}
x_1\theta(y_1)(a)\theta(b)(x_1^{-1}) &= x_1\theta_1(y_1)a\theta_1(y_1)^{-1}\theta_1(b)x_1^{-1}\theta_1(b)^{-1}\\
&= (x_1\theta_1(y_1))a\theta_1(y_1)^{-1}\theta_1(b)\theta_1(y_1)\theta_1(y)^{-1}x^{-1}\theta_1(b)^{-1}\\
&= (x\theta_1(y))a\theta_1(-y_1+b+y_1)\theta_1(y)^{-1}x^{-1}\theta_1(b)^{-1}\\
&= x\theta_1(y)a\theta_1(b)\theta_1(y)^{-1}x^{-1}\theta_1(b)^{-1}\\
&= x\theta_1(y)a\theta_1(y)^{-1}\theta_1(b)x^{-1}\theta_1(b)^{-1}\\
&= x\theta(y)(a)\theta(b)(x^{-1}).
\end{aligned} \tag{4.6}$$

It can be easily verified that if $x_1\theta_1(y_1) = -x\theta_1(y)$, then the above equation also holds. Also note that the center of $Z(\mathrm{SL}(2,\mathbb{Z}_p)) = \pm I$. Hence the set of solutions for the special conjugacy problem is

$$S = Inn^{-1}(Inn(g)) = \{(x_1,y_1)|\ y_1\in\mathbb{Z}_p,\ x_1 = \pm x\theta_1(y-y_1)\}. \tag{4.7}$$

The cardinality of S, $|S|$ is $2p$. Note that if $Inn(g) = Inn(g_1)$, then $Inn(g^{-1}g_1) = Id$. It means that $g^{-1}g_1$ is an element of the center of G. Also for any central element z, $Inn(gz) = Inn(g)$. So we know that $S = Inn^{-1}(Inn(g)) = gZ(G)$ and

$$Z(G) = \{(x_1,y_1)|\ y_1\in\mathbb{Z}_p,\ x_1 = \pm\theta_1(-y_1)\}. \tag{4.8}$$

The cardinality of the center of G is $2p$. Note that the probability to choose m and g in the center is smaller than $2p/p^3 = 2/p^2 \approx 0$ and $2p/p^4 = 2/p^3 \approx 0$, respectively.

For a given $Inn(g)$, m satisfying $[g, m] = e_G$ is a fixed point, i.e. $Inn(g)^{ab}(m) = m$. The cardinality of $Z[g] = \{m \mid [g, m] = e_G\}$ is $2p^2$ if we choose g of order p [16] and thus the probability to choose m in $Z[g]$ is smaller than $2p^2/p^3 = 2/p \approx 0$.

Remark 3. From Theorem 2 in section 4.3, DLP in G is reduced to a linear equation $ny = Y$ for given $y \neq 0, Y$, and so it is an easy problem.

4.2 Parameter Selections

We will apply the above scheme to $G = \mathrm{SL}(2, \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$. Since the last component is invariant under the conjugation, we must take the message in $\mathrm{SL}(2, \mathbb{Z}_p)$ (see (4.4)).

In [20], we see

$$\{T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},\ S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\}$$

is a generator set of $\mathrm{SL}(2, \mathbb{Z})$ and hence it is also a generator set of $\mathrm{SL}(2, \mathbb{Z}_p)$. Moreover there exists an algorithm which finds a decomposition of each $g \in \mathrm{SL}(2, \mathbb{Z}_p)$ as a product of T, S [2],[20], i.e.

$$g = S^{i_0} T^{j_1} S T^{j_2} \cdots S T^{j_n} S^{i_{n+1}},$$

where i_0, i_{n+1} is either 0 or 1 and $j_k = \pm 1, \pm 2 \cdots$.

Theorem 1. *If $g \in SL(2, \mathbb{Z}_p)$ with non zero $(2,1)$-entry,*

$$g = T^{j_1} S T^{j_2} S T^{j_3}.$$

Proof. By computing $T^{j_1} S T^{j_2} S T^{j_3}$, we obtain

$$\begin{pmatrix} 1 & j_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & j_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & j_3 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} j_1 j_2 - 1 & j_1 j_2 j_3 - j_3 - j_1 \\ j_2 & j_2 j_3 - 1 \end{pmatrix}.$$

From this equation and the fact that $\mathbb{Z}_p$ is a field, we can find j_1, j_2, j_3 such that $g = T^{j_1} S T^{j_2} S T^{j_3}$ for any $g \in \mathrm{SL}(2, \mathbb{Z}_p)$. (Since every element of $\mathrm{SL}(2, \mathbb{Z}_p)$ is determined when three entries are determined, we only need to consider 3 entries.)

Note that since $\mathbb{Z}$ is not a field, the above theorem does not hold in $\mathrm{SL}(2, \mathbb{Z})$.

Since $\{(T, 0), (S, 0), (I, 1)\}$ is a set of generators of G, we can obtain $Inn(g)$ if we know gTg^{-1}, gSg^{-1} and $g(I,1)g^{-1}$. Since $m \in \mathrm{SL}(2, \mathbb{Z}_p)$ and $\mathrm{SL}(2, \mathbb{Z}_p)$ is a normal subgroup of G, the restriction of $Inn(g)$ to $\mathrm{SL}(2, \mathbb{Z}_p)$, $Inn(g)|_{\mathrm{SL}(2,\mathbb{Z}_p)}$ can be considered as an automorphism of $\mathrm{SL}(2, \mathbb{Z}_p)$. Hence the

public key is $Inn(g)|_{\mathrm{SL}(2,\mathbb{Z}_p)}$ and $Inn(g^a)|_{\mathrm{SL}(2,\mathbb{Z}_p)}$, precisely. In order to express $Inn(g)|_{\mathrm{SL}(2,\mathbb{Z}_p)}$, we only need to know $\{gTg^{-1}, gSg^{-1}\}$.

We choose $\theta_1(1)$ among elements of $\mathrm{SL}(2,\mathbb{Z}_p)$ whose order is p, e.g. $I+\delta_{12}$.

We compute the order of $g=(x,y)\in G$. If $y\neq 0$, then the order of g is a multiple of p.

Theorem 2. *For* $(x,y)\in G$,

$$(x,y)^n = ((x\theta_1(y))^n\theta_1(y)^{-n}, ny).$$

Proof. We prove this using induction. For $n=1$, it is clear. We assume that Theorem 2 holds for $n=k$. Then we obtain that

$$\begin{aligned}(x,y)^{k+1} &= (x,y)^k(x,y) = ((x\theta_1(y))^k\theta_1(b)^{-k}, ky)(x,y)\\ &= ((x\theta_1(y))^k\theta_1(y)^{-k}\theta(y)^k(x), (k+1)y)\\ &= ((x\theta_1(y))^k\theta_1(y)^{-k}\theta_1(y)^k x\theta_1(y)^{-k}, (k+1)y) \qquad (4.9)\\ &= ((x\theta_1(y))^k(x\theta_1(y))\theta_1(y)^{-(k+1)}, (k+1)y)\\ &= ((x\theta_1(y))^{k+1}\theta_1(y)^{-(k+1)}, (k+1)y),\end{aligned}$$

which completes the proof.

We may choose $g=(x,y)$ satisfying $x\theta_1(y)\in A(I+c\delta_{12})A^{-1}$ for some fixed $c\in\mathbb{Z}_p$ and $A\in\mathrm{SL}(2,\mathbb{Z}_p)$. Then we obtain that the order of $Inn(g)$ is p by Theorem 2. If we choose g arbitrarily and the order of g is not fixed, then the security may be increased since we should know the order of a given cyclic group to apply a known algorithm for DLP, i.e. we should solve DLP under the assumption that the order of g is pd for each $d|(p+1)(p-1)$.

4.3 Security and Efficiency

Security of the system. We check the security of our system against solving DLP in $\langle Inn(g)\rangle$ directly. From the public data, $Inn(g)$ and $Inn(g^a)$, we solve DLP to obtain the secret key a. In this case, it seems that the fastest algorithm (index calculus) to solve DLP cannot be applied since $\langle Inn(g)\rangle$ is contained in $\mathrm{Aut}(G)\subset\mathrm{End}(G)\subset G^G$, where $\mathrm{End}(G)$ is the endomorphism group of G and G^G is the set of all function from G to G. We cannot apply the index calculus to any of them since they are not even expressed as matrix groups.

So an expected run time for solving DLP is $O(\sqrt{p})$-group operations if the order of g is p. (In order to increase the security of the system, we can choose g with an order which is a multiple of p. If $p(p+1)=p_1^{e_1}\cdots p_n^{e_n}$, then the total number of divisors of $p+1$ is $(e_1+1)\cdots(e_n+1)$. To find the order of g, we need $(e_1+1)\cdots(e_n+1)$-trials, and it takes $(e_1+1)\cdots(e_n+1)O(\sqrt{p})$-group operations [17].)

Now we check the security of our system against the second method in section 3. As we see in Appendix A, the special conjugacy problem in G is not a hard

problem. Let $S = \{g_1 |\ Inn(g_1) = Inn(g^a)\}$. We can immediately obtain a from $g = (x, y)$ and $g^a = (X, Y)$ since if

$$(x, y)^a = ((x\theta_1(y))^n \theta_1(y)^{-n}, ay) = (X, Y), \tag{4.10}$$

we only need to solve $ay = Y$ for solving DLP for g and g^a. But since $|S| = 2p$, we need $O(p)$-trials to find g^a in S. So it is less efficient than finding a from $Inn(g)$ and $Inn(g^a)$ directly.

For DLP to be a hard problem in $\langle Inn(g)\rangle$, we choose 160 bit prime p. Then the security of our system is comparable to 1024-bit RSA. (An expected run time for solving DLP in $\langle Inn(g)\rangle$ and for factorization in 1024-bit RSA is about 2^{87} and 2^{80}, respectively.)

If we compare our system with RSA and XTR, our system has the following advantage. In RSA and XTR, an expected run time to find the private key from the public key is subexponential, $L[n, 1/3, 1.923]$. In our system, it takes an exponential run time $O(\sqrt{p})$ as ECC.

Number of multiplications in $\mathbb{Z}_p$. Now we consider the number of multiplications in $\mathbb{Z}_p$ required for computing $Inn(g^b)$ from $Inn(g)$. We can express $Inn(g)(S)$ and $Inn(g)(T)$ as $T^{j_1}ST^{j_2}ST^{j_3}$ and $T^{l'_1}ST^{l'_2}ST^{l'_3}$, respectively. Each of them takes 2-multiplications by Theorem 1. Then

$$\begin{aligned} Inn(g^2)(S) &= Inn(g)(T^{j_1}ST^{j_2}ST^{j_3}) \\ &= (Inn(g)(T))^{j_1}(Inn(g)(S))(Inn(g)(T))^{j_2}(Inn(g)(S))(Inn(g)(T))^{j_3} \end{aligned}$$

and

$$\begin{aligned} Inn(g^2)(T) &= Inn(g)(T^{l_1}ST^{l_2}ST^{l_3}) \\ &= (Inn(g)(T))^{l_1}(Inn(g)(S))(Inn(g)(T))^{l_2}(Inn(g)(S))(Inn(g)(T))^{l_3}. \end{aligned}$$

From (2.3) in Remark 1, we can obtain $(Inn(g)(T))^j$ from $Inn(g)(T)$ with 4-multiplications. More precisely, if

$$Inn(g)(T) = \begin{pmatrix} x & y \\ z & w \end{pmatrix},$$

then

$$(Inn(g)(T))^j = \begin{pmatrix} 1 - j(1-x) & jy \\ jz & 1 + j(w-1) \end{pmatrix}.$$

It takes 92 multiplications for computing $Inn(g^2)(S)$ and $Inn(g^2)(T)$. So it takes about $92\log_2 p$ multiplications for computing $Inn(g^b)$ from $Inn(g)$. Also $92\log_2 p$ multiplications are needed to compute $Inn(g^{ab})$ from $Inn(g^a)$. So number of multiplications for encryption is $184\log_2 p$. Since one multiplication needs $O((\log_2 p)^2)$-bit operations [9], the encryption needs about $184(\log_2 p)^3 C \approx 8 \times 10^8 C$-bit operations for some constant C. In 1024-bit RSA, it takes $(\log_2 n)^3 C \approx (1024)^3 C \approx 10^9 C$-bit operations. If the public exponent in RSA encryption scheme is 32-bit number, then it takes $3.2 \times 10^7 C$-bit operations.

Fast encryption and decryption. We can reduce the number of bit operations as follows. Assume that Bob wants to send an encrypted message to Alice. Then Bob computes $Inn(g^a)^b$ and $Inn(g^b)$ for a fixed b and send $Inn(g^b)$ to Alice. As we see in section 3, we may fix b, i.e. contrary to ElGamal encryption, we cannot obtain $m_1^{-1}m_2$ from $Inn(g^b)(m_1)$ and $Inn(g^b)(m_2)$ in our scheme . Alice computes $Inn(g^b)^{-a}$. Bob will encrypt a message m as $E = Inn(g^a)^b(m)$ and send E to Alice. Alice will decrypt E by computing $Inn(g^b)^{-a}(E)$.

In order to compute $Inn(g^a)^b(m)$ from given $Inn(g^a)^b$ and m, it takes 46 multiplications, and so it takes about $1.2 \times 10^6 C$-bit operations in encryption. Even if 32-bit public exponent is used in RSA, $3.2 \times 10^7 C$-bit operations are needed in encryption. Encryption of our system is about 30 times faster than 1024-bit RSA.

In decryption of our system, we need the same number of multiplications as the encryption. In decryption of RSA, it takes about $2.5 \times 10^8 C$-bit operations even if we use the Chinese Remainder Theorem. Thus decryption of our system is 200 times faster than that of RSA.

If we compare our system with ECC, our system has an advantage in the decryption too. In ECC, since b is not fixed, precomputations of g^b is impossible. Then the number of multiplications for decryption in 170-bit ECC are 1900, respectively. Then it is about 40 times faster than ECC.

In ECC, it needs $O(\log_2 p)$ multiplications in decryption, and thus the number of multiplications will increase linearly with respect to the number of bits $\log_2 p$. The decryption of our system always needs 46 multiplications which are independent of the size of p. In Table 1, we roughly compare the number of multiplications for decryption in our system with ECC. Note that the cryptosystems in the same row have the same securities roughly.

This fast scheme can be useful in many applications.

Table 1. Comparison of run times for decryption with ECC(multiplications)

	our PKC(r-bit)	r-bit ECC
$r = 170$	46	1900
$r = 240$	46	2700
$r = 310$	46	3500

Remark 4. We can encrypt a message with a padding as follows (see also [16]). Let $M \neq 0$ be a message and r_1, r_2 be random numbers in $\mathbb{Z}_p$. We encrypt $m = \begin{pmatrix} M & r_1 \\ r_2 & \frac{1+r_1 r_2}{M} \end{pmatrix} \in \mathrm{SL}(2, \mathbb{Z}_p)$. Then m can be an element of any conjugacy class by varying r_1, r_2. It prevents an adversary from determining the right plaintext among two given plaintexts by examining the conjugacy class of E. Furthermore, since b is fixed, the encryption and the decryption is also fast but the encryption scheme is not deterministic.

Expression and key size. Since $Inn(g^a)(T)$ and $Inn(g^a)(S)$ can be considered as elements of $\mathrm{SL}(2, \mathbb{Z}_p)$, we can express them by three entries. Since $Inn(g^a)(T)$ can be expressed by $3\log_2 p$-bit, $6\log_2 p$-bit are needed to express $Inn(g^a)$. If p is a 160-bit prime number, then it takes 960-bit to express $Inn(g^a)$. So we can express the public key with smaller size than RSA.

The secret key size is $\log_2 p \approx 160$-bit, and so it is much smaller than 1024-bit RSA.

5 Other Examples

5.1 The General Linear Group $\mathrm{GL}(k, \mathbb{Z}_p)$

One of the most familiar non abelian group is the general linear group $\mathrm{GL}(k, \mathbb{Z}_p)$. Since cI is a central element for any $c \neq 0$, the center of $\mathrm{GL}(k, \mathbb{Z}_p)$ is sufficiently large, i.e. $|Z(\mathrm{GL}(k, \mathbb{Z}_p)| \geq p/2$. We know that $Inn(g)$ can be represented by a linear map on the $k \times k$-matrix ring [15]. So we can represent $Inn(g)$ by a $k^2 \times k^2$-matrix, $R(g)$. So the DLP on $\langle Inn(g) \rangle$ is convertible to the DLP on the $k^2 \times k^2$-matrix ring.

We must be careful in the choice of g. Considering an attack using the determinant [15], we choose g whose order is much larger than p (e.g. $p(p-1)$). It would be better to choose g satisfying that $\det(R(g)) = 1$. Also the characteristic polynomial of $R(g)$ should be irreducible.

5.2 Other Constructions

We introduce some methods to obtain non abelian groups. For a given non abelian group G, we can obtain a new non abelian group $Inn(G)$ as we see in previous sections. Also $Inn(Inn(G))$ can be obtained from $Inn(G)$. Inductively we can make many non abelian groups from a given non abelian group. Since $Inn(G) = G/Z(G)$, this method reduces the size of a given group.

Extensions of non abelian groups is obtained as follows. First, Let θ_1 be a homomorphism on G. (It may be the identity map.) We define θ as follows:

$$\begin{aligned} \theta = Inn \circ \theta_1 : G &\to Aut(G) \\ g &\mapsto Inn(\theta_1(g)) \end{aligned}$$

Then we construct an extension of G, $\bar{G} = G \times_\theta G$. We can easily obtain $Z(\bar{G}) = \{(X, Y) \in G \times_\theta G \mid x, y \in Z(G)\}$. If we use the group G in section 4, $|Z(\bar{G})| = 4p^2$.

Secondly, Let G be a non abelian group and H be a subgroup of automorphism group $Aut(G)$. We construct a non abelian group naturally. Let $\theta = \mathrm{Id}$. Then we can easily obtain $G \times_\theta H$. For example, we can always obtain $Inn(G) = G/Z(G)$ and $G \times_\theta Inn(G)$. If we know other subgroups of $Aut(G)$, we can construct many useful non abelian groups.

Nilpotent group $G = (\mathbb{Z}_p \times \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$. Since $\mathrm{Aut}(\mathbb{Z}_p \times \mathbb{Z}_p) = \mathrm{GL}(2, \mathbb{Z}_p)$, we can make the following non abelian group. There exists an injective homomorphism

$$\theta : \mathbb{Z}_p \to \mathrm{SL}(2, \mathbb{Z}_p).$$

Then we can construct $G = (\mathbb{Z}_p \times \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$. Since G is a p-group, it is a nilpotent group. Hence G has a non trivial center and its cardinality is at least p.

In this case, a generator set of G is $\{e_1 = (1,0,0), e_2 = (0,1,0), e_3 = (0,0,1)\}$ and we can easily express any elements of G as a product of e_i's. Then

$$Inn((X,y))(e_1) = (\theta(y)((1,0)),0)$$

and

$$Inn((X,y))(e_2) = (\theta(y)((0,1)),0).$$

So $\theta(y) \in \mathrm{SL}(2, \mathbb{Z}_p)$ can be easily obtained. If $g^a = (X', y^a)$ and $g = (X, y)$, then we can obtain $\theta(y)^a$ and $\theta(y)$. We can solve DLP in $\mathrm{SL}(2, \mathbb{Z}_p)$ as in the Remark 1. Hence the cryptosystem in section 3 is not secure in $G = (\mathbb{Z}_p \times \mathbb{Z}_p) \times_\theta \mathbb{Z}_p$.

Since the variables X, y are separated, this phenomenon occurs. We note here that X, y are separated since the subgroup $\mathbb{Z}_p \times \mathbb{Z}_p$ is abelian. To prevent the separation of variables, we suggest the following non abelian group.

Semi-direct product $G = (\mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q) \times_{\theta_2} \mathbb{Z}_q$. We replace the abelian group $\mathbb{Z}_p \times \mathbb{Z}_p$ by $\mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q$, where q is a prime satisfying $q|(p-1)$. Then we can prevent the separation of variables. Since $\mathrm{Aut}(\mathbb{Z}_p) \cong \mathbb{Z}_p^* \cong \mathbb{Z}_{p-1}$, we can make $\mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q$, where θ_1 is an injective homomorphism from $\mathbb{Z}_q$ to $\mathbb{Z}_p^*$. We denote $\mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q$ by H. Then H is not abelian.

We will apply the same method as in section 4.2. We can consider $\mathbb{Z}_q$ as a subgroup of H. Its conjugate is also a cyclic subgroup of order q. Let K be one of the conjugates of $\mathbb{Z}_q$ in H. Then there exists an isomorphism θ' from $\mathbb{Z}_q$ to K, and $\theta_2 = Inn \circ \theta'$.

Equations (4.5),(4.6), (4.7) also hold in $G = (\mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q) \times_{\theta_2} \mathbb{Z}_q$. Then we can find the center of G of order q as in 4.2.

In this case, we denote a generator set of $\mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q$ by $e_1 = (1,0,0)$ and $e_2 = (0,1,0)$. Since $\mathbb{Z}_p$ is a normal subgroup of G, we assume that $Inn(g)(e_1) = (a_1, 0, 0)$ and $Inn(g)(e_2) = (b_1, b_2, 0)$. We can prove that

$$(z,w)^k = \left(\frac{\theta_1(w)^k - 1}{\theta_1(w) - 1} z, kw\right) = \left(\frac{\theta_1(1)^{wk} - 1}{\theta_1(1)^w - 1} z, kw\right)$$

for $(z,w) \in \mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q$ by induction. Then we have for $g = (x_1, x_2, y)$,

$$Inn(g^2)(e_1, 0) = Inn(g)(a_1, 0, 0) = (Inn(g)(e_1))^{a_1} = (a_1, 0, 0)^{a_1} = (a_1^2, 0, 0)$$

and

$$\begin{aligned} Inn(g^2)(e_2,0) &= Inn(g)(b_1,b_2,0) = (Inn(g)(e_1))^{b_1}(Inn(g)(e_2))^{b_2} \\ &= (a_1,0,0)^{b_1}(b_1,b_2,0)^{b_2} = (a_1b_1,0,0)(\frac{\theta_1(1)^{b_2^2}-1}{\theta_1(1)^{b_2}-1}b_1, b_2^2, 0) \\ &= (a_1b_1 + \frac{\theta_1(1)^{b_2^2}-1}{\theta_1(1)^{b_2}-1}b_1, b_2^2, 0). \end{aligned}$$

From this, we obtain that $Inn(g^k)(e_1) = a_1^k \in \mathbb{Z}_p$. Since $H = \mathbb{Z}_p \times_{\theta_1} \mathbb{Z}_q$ is not an abelian group, the order of $\theta_1(1)$ is q. Thus DLP in $\langle Inn(g)\rangle$ can be reduced to DLP in $\mathbb{Z}_p$, and so the cryptosystem in section 4 is not secure in $G = (\mathbb{Z}_p \times_{\theta_1} Z_q) \times_{\theta_2} \mathbb{Z}_q$.

The reason of this phenomenon is $\mathbb{Z}_p$ is an abelian normal subgroup. If α is a generator of an abelian normal subgroup, then $Inn(g)(\alpha) = \alpha^s$ for some s and $Inn(g^k)(\alpha) = \alpha^{s^k}$. So we can reduce DLP in $\langle Inn(g)\rangle$ to DLP in $\langle\alpha\rangle \subset \mathbb{Z}_p$. If we use $Inn(Inn(G))$ instead of $Inn(G)$, we can avoid such an attack.

6 Concluding Remarks

We have presented a novel public key cryptosystem (based on a finite non abelian groups) and suggested some examples of finite non abelian groups. There may be other non abelian groups to be used in our system. However we must be careful in applying a non abelian group to our cryptosystem in order that the cryptosystem is secure. As we see in section 5, we should check the following:

- The existence of abelian normal subgroup reduces the security of the cryptosystems. So any abelian normal subgroup must be of small order.
- The algorithm to express an element of G as a product of generators must be efficient.
- Since $Inn(g)$ is expressed as $\{Inn(g)(\gamma_i) \in G \mid \gamma_i \text{ is a generator}\}$, both the number of generators and bits needed to express an element of G must be of small order.

We may use other homomorphisms from G to $\mathrm{Aut}(G)$ instead of the inner automorphism (if exists). Also we can consider the DLP in the endomorphism group $End(G)$.

If we know any representation of G, G can be considered as a subset of a large matrix group up to the kernel (we call a homomorphism from G to a matrix group a representation of G). Hence the representation of G is very useful for cryptosystem as in section 3. If we use DLP in a subgroup $\langle g\rangle$ of a non abelian group and a representation R of G, it would be better to choose $\det(R(g)) = 1$ [15].

Acknowledgment. We would like to thank to our colleagues in NSRI and Dr. Bae Eun Jung for their useful comments. Also we would like to express our gratitude to professor Hong-Jong Kim and Professor Ki-Suk Lee for their kind advice.

References

1. I. Anshel, M. Anshel, D. Goldfeld *An algebraic method for public-key cryptography*, Mathematical Research Letters 6 (1999), 1–5
2. S. Blackburn, S. Galbraith *Cryptanalysis of two cryptsystems based on group actions*, Proc. ASIACRYPT' 99 (2000), 52–61
3. A. E. Brower, R. Pellikaan, E. R. Verheul *Doing more with fewer bits*, Proc. ASIACRYPT' 99 (2000), 321–332
4. D. Coopersmith, A. M. Odlzyko, R. Schroeppel *Discrete logarithms in GF(p)* , Algorithmica, 1 (1986), 1–15
5. T. ElGamal *A public key cryptosystem and a signature scheme based on discrete logarithms* , IEEE Transactions andInformation Theory, 31 (1985), 469–472
6. S. Flannery *Cryptography:An investigation of a new algorithm vs. the RSA*, http://cryptome.org/flannery-cp.pdf, 1999
7. T. W. Hungerford *Algebra*, Springer-Verlag
8. A. K. Lenstra, E. R. Verheul. *The XTR public key system*, Proc. Crypto 2000 (2000), 1–20
9. A. J. Menezes, P. C. Van Oorshot, S. A. Vanstone *Handbook of applied cryptography*, CRC press, 1997
10. R. Lidl, H. Niederreiter *Introduction to finite fields and their application*, Cambridge University press, 1986
11. K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. -S. Kang, C. Park *New public-key cryptosystem using braid groups*, Proc. Crypto 2000 (2000), 166–184
12. N. Koblitz *Elliptic curve cryptosystems*, Mathematics of Computation, 48 (1987), 203–209
13. V. Miller *Use of elliptic curves in cryptography*, Proc. Crypto 85 (1986), 417–426
14. K. Nyberg, R. Rueppel *A new signature scheme based on DSA giving message recovery*, 1st ACM Conference on Computer and Communications Security, (1993), 58–61
15. S.-H. Paeng, J.-W. Han, B. E. Jung *"The security of XTR in view of the determinant"*, preprint, 2001
16. S.-H. Paeng *" A provably secure public key cryptosystem using finite non abelian groups"*, preprint, 2001
17. S. C. Pohlig, M. E. Hellman *An improved algorithm for computing logarithms over GF(p) and its cryptographic significance*, IEEE Transactions on Information Theory, 23 (1978), 106–110
18. J. M. Pollard *Monte Carlo methods for index computation (mod p)*, Mathematics of computation, 32 (1978), 918–924
19. R. L. Rivest, A. Shamir, L. M. Adleman *A method for digital signature and public-key cryptosystems*, Communications of the ACM, 21 (1978), 120–126
20. A. Yamamura *Public key cryptsystems using the modular group*, PKC'98, 203–216

Appendix A : Special Conjugacy Problem in Matrix Groups

Let G be a matrix group, for example $\mathrm{GL}(2, R)$ or $\mathrm{SL}(2, R)$, where $R = \mathbb{Z}$ or $\mathbb{Z}_p$ for a prime number p. We will solve the special conjugacy problem in G. Let

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \ X = \begin{pmatrix} x & y \\ z & w \end{pmatrix}.$$

We will find X from XAX^{-1} for $A \in G$. Let

$$A = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix} \text{ and } XAX^{-1} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}.$$

From the above equation, we obtain the following linear equations,

$$\begin{aligned} ax &= \alpha x + \beta z \\ az &= \gamma x + \delta z \\ bx + dy &= \alpha y + \beta w \\ bz + dw &= \gamma y + \delta w. \end{aligned}$$

From the first equation, we can easily obtain the ratio of x to z, i.e. $(a-\alpha)x = \beta z$. (Note that we cannot obtain other ratios as we see in Example 2.)

Similarly, if we solve the conjugacy equation for $XA'X^{-1}$ and

$$A' = \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix}, c' \neq 0$$

we can also get another linear system. If we replace βz by $(a-\alpha)x$, we can obtain remaining ratios between x, y, z and w. So we can solve the special conjugacy problem in G easily. By Example 2, we can easily understand the procedure.

Note that the conjugacy problems in $\mathrm{SL}(2, R)$ or $\mathrm{GL}(2, R)$ are not difficult since we can obtain at most two linear equations by the conjugacy equation.

Example 2. In [20], the author suggested a public key system using $\mathrm{SL}(2, \mathbb{Z})$. It was shown that this system is not secure in [2]. For the point based scheme in [2], we can find the secret key if we solve the conjugacy equations directly as above. Let

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Then $\{A, B\}$ is a generator set of $\mathrm{SL}(2, \mathbb{Z})$. Furthermore, $A^3 = B^2 = -I$, and so every element in $\mathrm{SL}(2, \mathbb{Z})$ can be expressed as the normal form $\pm A^{i_1} B \cdots A^{i_{n-1}} B A^{i_n}$, where $i_j = 0, 1$ or 2. In the public key system suggested in [2], they use a semi-group generated by $\{V_1 = (BA)^i,\ V_2 = (BA^2)^j\}$ for given $i, j \geq 2$. The public key is $\{MV_1M^{-1}, MV_2M^{-1}\}$ and the secret key is M. In order to find the secret key from the public key, we must solve the conjugacy equations. For example, let

$$V_1 = (BA)^2 = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}, V_2 = (BA^2)^2 = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} \text{ and } M = \begin{pmatrix} 3 & 1 \\ 5 & 2 \end{pmatrix}$$

Then the public key is

$$MV_1M^{-1} = \begin{pmatrix} -3 & 2 \\ -8 & 5 \end{pmatrix}, MV_1M^{-1} = \begin{pmatrix} 31 & -18 \\ 50 & -29 \end{pmatrix}.$$

Put

$$M = \begin{pmatrix} x & y \\ z & w \end{pmatrix}$$

and find M by solving the conjugacy equation for V_1 and MV_1M^{-1}. We obtain the following linear equations:

$$4x - 2y - 2z = 0$$

$$4y - 2w = 0$$

$$8x - 4z - 2w = 0$$

$$8y - 4w = 0.$$

Then we have $2y = w$. (Check that we cannot obtain other ratios from these equations.)

If we solve the conjugacy equation for V_2 and MV_2M^{-1}, we obtain that $5x = 3z$ and $x + 15y - 9w = 0$. Replacing w by $2y$, we have $x = 3y$ so $2x = 3w$. Hence we obtain the secret key

$$M = C \begin{pmatrix} 3 & 1 \\ 5 & 2 \end{pmatrix}$$

for some C. Since $\det(M) = 1$, we obtain that $C = 1$.

We should note that the dimension of solutions in $\mathrm{GL}(2, R)$ is always larger than 1. From one conjugacy equation, we can obtain at most two linearly independent equations. Combining two conjugacy equations, we obtain three linearly independent equations and one dimensional solutions. In $\mathrm{SL}(2, \mathbb{Z}_p)$, we obtain only one solution. We can apply the same method to any other V_1, V_2 which are suggested in [2].