3-2 A Functional Cryptosystem Using a Group Action

YAMAMURA Akihiro

The main purpose of this paper is to examine applications of group theoretical concepts to cryptography. We construct a backward deterministic system employing the action of the modular group on the upper half plane and the amalgamated free product structure of the group. We invent a geometrical algorithm that finds the normal form of an element of the modular group effectively. This algorithm makes our backward deterministic system tractable. Using the backward deterministic system, we invent a public-key cryptosystem in terms of a functional cryptosystem.

Keywords
Public-key cryptosystem, Functional cryptosystem, Backward deterministic system, Modular group, Amalgamated free product

1 Introduction

Many public-key cryptosystems depend on the difficulty of solving a few specific problems such as finding the prime factorization of a composite number and the discrete logarithm problem. While the existing systems depending on the hardness of these problems are considered secure, there is still deep concern about the security of these systems. Shor [8] invented a fast algorithm for prime factorization and the discrete logarithm problem based on quantum computing. Adleman[1] also reported that a DNA computer solves a 7 vertex and 14 edge instance of the Hamiltonian path problem. An attempt to construct a hardware specialized for factorization problem is implementing. Therefore we should avoid the situation that all the cryptosystems in hand depend on a few principles. Our intention is to provide backup cryptosystems for the currently working cryptosystems depending on difficulties of solving a few specific problems. We propose a public-key cryptosystem as a first step toward inventing a scheme of cryptography using new technologies from mathematics other than number theory. We employ the modular group and import several ideas from combinatorial group theory. The encryption and decryption of our cryptosystem are based on the uniqueness of a certain expression of an element of the modular group and its action on the upper half plane.

First, we briefly review a functional cryptosystem which is the basic scheme of ours. We give the definitions of a backward deterministic system and a morphism between two backward deterministic systems. Then we demonstrate how to construct a backward deterministic system using a group action on a certain space.

Secondly, we recall basic results on combinatorial group theory. An amalgamated free product of groups is introduced and explained. The modular group is the group of 2×2 matrices over rational integers with determinant one. It is known that the modular group is an amalgamated free product of finite cyclic groups. We give a geometrical algorithm that finds the normal form of a matrix in the modu-

lar group using the action of the modular group on the upper half plane. The algorithm is very efficient because of its geometrical nature.

Thirdly, we provide a public-key cryptosystem in terms of a backward deterministic system using the action of the modular group on the upper half plane. A similar cryptosystem using the modular group was introduced in[14]. Our approach is different from them in that ours is based on functional cryptosystem and also our decryption algorithm is faster. We explain the public key, the private key, the encryption and decryption methods. We discuss security issues of the system.

2 Functional cryptosystems

The concept of a functional cryptosystem was introduced to build a public-key cryptosystem using grammar theoretical concepts (see[4][5][10][11][12]). In this section we review several concepts and terminologies. Let χ be a set and f_i a function of χ into χ where I is a finite set. We suppose that there is an element $x \in \chi$ such that if we have

$$f_{i_1} \circ f_{i_2} \circ \ldots \circ f_{i_n}(x) = f_{j_1} \circ f_{j_2} \circ \ldots \circ f_{j_m}(x)$$

where $i_1, i_2, \ldots, i_{n_1}, j_1, j_2 \ldots, j_m \in I$, $k = 1, 2, \ldots, n$ then $n = m$ and $i_k = j_k$. The triple $(\{f_i(i \in I)\}, x, \chi)$ is called a backward deterministic system. Now let $(\{f_i(i \in I)\}, x, \chi)$ and $(\{g_i(i \in I)\}, y, \gamma)$ be backward deterministic systems. The morphism ϕ of $(\{f_i(i \in I)\}, x, \chi)$ to $(\{g_i(i \in I)\}, y, \gamma)$ is a mapping $\phi : \chi \to \gamma$ satisfying $\phi(x) = y$ and also $\phi \circ f_i = g_i \circ \phi$ for each $i \in I$. Assume that $p = f_{i_1} \circ f_{i_2} \circ \ldots \circ f_{i_n}(x)$. Let $q = \phi(p)$.

Then we have

$$\begin{aligned} q = \phi(p) &= \phi(f_{i_1} \circ f_{i_2} \circ f_{i_3} \circ \ldots \circ f_{i_n}(x)) \\ &= g_{i_1}(\phi(f_{i_2} \circ f_{i_3} \circ \ldots \circ f_{i_n}(x)) \\ &= g_{i_1} \circ g_{i_2}(\phi(f_{i_3}(\ldots \circ f_{i_n}(x)\ldots) \\ &= \ldots \\ &= g_{i_1} \circ g_{i_2} \circ \ldots \circ g_{i_n}(\phi(x)) \\ &= g_{i_1} \circ g_{i_2} \circ \ldots \circ g_{i_n}(y) \end{aligned}$$

Note that the morphism ϕ preserves information on the sequence $i_1, i_2, \ldots, i_n$. We employ backward deterministic systems to construct a public-key cryptosystem. The most significant point in making up a public-key cryptosystem is to supply a trapdoor. In the case of a functional cryptosystem, the idea is to find two backward deterministic systems with distinct complexities and an effectively computable morphism between them. We require that one of the backward deterministic systems $(\{f_i(i \in I)\}, x, \chi)$ to be harder than the other in the following sense: Let $p = f_{i_1} \circ f_{i_2} \circ \ldots \circ f_{i_n}(x)$. If we are given the point p on χ, we have no efficient way to find how we apply f_i's on x to get the point p. We remark that there is a unique way to obtain p by applying f_i's on x, since $(\{f_i(i \in I)\}, x, \chi)$ is backward deterministic. On the other hand, the other backward deterministic system $(\{g_i(i \in I)\}, y, \gamma)$ is feasible, that is, if we have $q = g_{i_1} \circ g_{i_2} \circ \ldots \circ g_{i_n}(y)$, there is an efficient algorithm that finds how to apply g_i's on y to get q. A morphism ϕ of $(\{f_i(i \in I)\}, x, \chi)$ into $(\{g_i(i \in I)\}, y, \gamma)$ is a part of the trapdoor of the cryptosystem. We publicize the backward deterministic system $(\{f_i(i \in I)\}, x, \chi)$ and keep $(\{g_i(i \in I)\}, y, \gamma)$ and ϕ secret. A message sender encrypts a message $i_1 i_2 \ldots i_n$ into the composition $f_{i_1} \circ f_{i_2} \circ \ldots \circ f_{i_n}$ of the mappings, computes the point $p = f_{i_1} \circ f_{i_2} \circ \ldots \circ f_{i_n}(x)$ on χ and then sends p to a legal receiver. The legal receiver operates the trapdoor ϕ to the encrypted text p and get $q = \phi(p)$. Since ϕ is a morphism of the backward deterministic systems, we have $q = g_{i_1} \circ g_{i_2} \circ \ldots \circ g_{i_n}(y)$. Then the legal receiver can obtain the sequence of the mappings $g_{i_1} \circ g_{i_2} \circ \ldots \circ g_{i_n}$ using the efficient algorithm for $(\{g_i(i \in I)\}, y, \gamma)$. Hence, the original message $i_1 i_2 \ldots i_n$ can be obtained by the legal receiver. On the other hand, an eavesdropper may be able to get a message p and $(\{f_i(i \in I)\}, x, \chi)$ is public information.

However, the eavesdropper cannot obtain the sequence of mappings $f_{i_1} \circ f_{i_2} \circ \ldots \circ f_{i_n}$ from the information p and the backward deterministic system $(\{f_i(i \in I)\}, x, \chi)$, since the system $(\{f_i(i \in I)\}, x, \chi)$ is intractable. Therefore, the cryptosystem is secure in principle. If we can find a pair of backward deter-

ministic systems and a morphism satisfying the computational complexity requirements, we can employ them to build a public-key cryptosystem. This type of a cryptosystem is called a functional cryptosystem.

We now propose a functional cryptosystem using a group action on a certain object in mathematics. Let G be a group, χ a non-empty set (or some other mathematical object). We say that G acts on χ if there is a mapping ρ of $G \times \chi$ into χ(we usually denote the image $\rho(g, x)$ of (g, x) under ρ by gx) satisfying the followings:

(i) For $a, b \in G$, and $x \in \chi$, we have $(ab)x = a(bx)$.

(ii) For $x \in \chi$, we have $1x = x$ where 1 is the identity element of G.

Suppose that a group G acts on a set χ. Then each element g of G can be regarded as a one-to-one function of χ onto χ under the rule $x \to gx$. Now we consider a homomorphism ϕ of a group G acting on a set χ to a group H acting on a set γ. Assume that a mapping f of χ into γ satisfies $f, (gx) = \phi(g)f(x)$ for each $g \in G$ and $x \in \chi$. Let $g_1 \in G$ and $x \in \chi$. Suppose that $(\{\phi(g_i)(i \in I)\}f(x), \gamma)$ is a backward deterministic system. Then clearly $(\{g_i(i \in I)\}, x, \chi)$ is also a backward deterministic system. The mapping f is a morphism between two systems. We offer a concrete example of such a functional cryptosystem using the modular group in Section **5**.

3 Amalgamated free products

Combinatorial group theory is the research of presentations of groups by generators and relators. Many results concerning algorithms on words or sequences on an alphabet have been obtained in this area of mathematics. This simply implies that concepts in combinatorial group theory meshes the theory of algorithms on words or sequences. In fact the modular group is employed in[14] to construct a cryptosystem. In this section we introduce several concepts from combinatorial group theory for our later use. For more details we refer the reader to[2][7][9]. Let χ be a non-empty set and R a set of words on $X \cup X^{-1}$. A group G is said to have a presentation $Gp(X|R)$ if G is a quotient group of a free group $F(X)$ on the set X by the normal subgroup N generated by the set R , that is, $G = F(X)|N$. An amalgamated free product is one of the most important constructions in combinatorial group theory. Intuitively, a free product of groups G_1 and G_2 amalgamating a subgroup H is a group containing groups G_1 and G_2 such that the intersection of G_1 and G_2 is exactly H. We now give the formal definition of an amalgamated free product of groups. Let G_1, G_2 be groups. Suppose that H_1(resp. H_2) is subgroup of G_1 (or G_2). We also assume that $\phi : H_1 \to H_2$ is an isomorphism. Then the free product of G_1 and G_2 amalgamating H_1 and H_2 is the group presented by

$$Gp(G_1, G_2 | \phi(H_1) = H_2)$$

where this presentation is an abbreviated form of the following presentation

$$Gp(X_1, X_2 | R_1, R_2, h^{-1}\phi(h) \forall h \in H_1)$$

provided that the groups G_1 and G_2 have the presentations

$$G_1 = Gp(X_1|R_1) \quad \text{and} \quad G_2 = Gp(X_2|R_2).$$

The amalgamated free product is usually denoted by $G_{1*H_1=H_2}G_2$. We usually identify the subgroups in the group $G_{1*H_1=H_2}G_2$. The most important aspect of an amalgamated free product is that every element in an amalgamated free product is expressed uniquely in a certain fashion. We introduce the concept of the normal form of an element of an amalgamated free product of groups as follows: Let G be the free product of groups G_1 and G_2 amalgamating H_1 and H_2, that is,

$$G = Gp(G_1, G_2 | H_1 = H_2).$$

We consider coset decompositions of G_1 by H_1 and G_2 by H_2, respectively. Choose a set of coset representatives for each decomposition. Suppose that $\{a_i | i \in I\}$ is the set of coset representatives of G_1 by H_1 and that $\{b_i | j \in J\}$ is the set of coset representatives of G_2 by H_2. Therefore we have the coset decompo-

sitions

$$G_1 = \bigcup_{i \in I} a_i H_1 \quad \text{and} \quad G_2 = \bigcup_{j \in J} b_j H_2 .$$

We suppose that an element g of the group G is written as $s_1 s_2 s_3 \ldots s_{n-1} s_n$ where s_n is in $H = H_1 = H_2$, each $s_k (k = 1, 2, \ldots, n\text{-}1)$ in not in H, but belongs to either $\{a_i | i \in I\}$ or $\{b_j | j \in J\}$ such that if s_k is in the former set of coset representatives then s_{k+1} is in the second set or vice versa. Then we say that g has the normal form $s_1 s_2 s_3 \ldots s_{n-1} s_n$, and that the expression $s_1 s_2 s_3 \ldots s_{n-1} s_n$ is of the normal form.

Proposition 1

Every element of $G_{1*_{H_1=H_2}} G_2$ can be written uniquely as a normal form, that is, if an element g in $G_{1*_{H_1=H_2}} G_2$ has two normal forms $s_1 s_2 s_3 \ldots s_{n-1} s_n$ and $t_1 t_2 t_3 \ldots t_{m-1} t_m$, then we have $n = m$ and $s_j = t_j$ for each $j = 1, 2, \ldots, n$.

For the proof, the reader is referred to [2][7][9]. We now suppose that we are given a free product G of finite groups G_1 and G_2 amalgamating a subgroup H. We choose sets of coset representatives of G_1 and G_2 by H. Suppose that $g = u_1 u_2 \ldots u_n$, is a product of alternate elements from G_1 and G_2 , that is, if $u_i \in G_1$ then $u_{i+1} \in G_2$ and vice versa. We give an algorithm that finds the normal form $s_1 s_2 \ldots s_n$ of g as follows:

Algorithm 1

INPUT: A decomposition $u_1 u_2 \ldots u_n$ of an element g in $G_{1*_{H_1=H_2}} G_2$ as a product of alternate sequence of elements from G_1 and G_2.

OUTPUT: The normal form $s_1 s_2 \ldots s_n$ of g.

Step 0)

We note that $u_1 \in G_1$ or $u_2 \in G_2$. We now assume that $u_1 \in G_1$. Then we have $u_1 = s_1 v_1$ where s_1 is a representative of H in G_1 and $v_1 \in H$. We rewrite g as $g = s_1 v_1 u_2 u_3 \ldots u_n$. We note that $s_1 \in G_1$ and $u_2 \in G_2$. In the case that $u_1 \in G_2$, we do the similar process.

Step 1)

We suppose that we have $g = s_1 s_2 \ldots s_m v_m u_t u_{t+1} \ldots u_n$ where $v_m \in H$ and s_1 is a representative of G_1 or G_2 , such that if $s_1 \in G_1$ then $s_{i+1} \in G_2$ or vice versa and also if $s_m \in G_1$ then $u_t \in G_2$ or vice versa. If there is no u_j in the sequence, we have a sequence of the form $g = s_1 s_2 \ldots s_m v_m$ where v_m is in H. Set $s_{m+1} \leftarrow v_m$. Then we return the normal form $g = s_1 s_2 \ldots s_m s_{m+1}$ and the algorithm terminates.

Now we assume that s_m is a representative of G_1. Then u_t is in G_2 and we can write $v_m u_t = s_m + v_{m+1}$ Where s_{m+1} is a representative of G_2 and $v_{m+1} \in H$.

Step 2)

If $s_{m+1} \notin H$, then we have $g = s_1 s_2 \ldots s_m s_{m+1} v_{m+1} u_{t+1} \ldots u_n$. We should note that $s_{m+1} \in G_2$ and $u_{t+1} \in G_2$. Then go to Step 1).

If $s_{m+1} \in H$, then we have $s_m s_{m+1} v_{m+1} u_{t+1} \in G_1$ since s_m, $u_{t+1} \in G_1$ and s_{m+1}, $v_{m+1} \in H \subset G_1$. Then we have $s_m s_{m+1} v_m v_{m+1} u_{t+1} = s'_m v'_m$ where s'_m is a representative of G_1 and $v'_m \in H$. Then set $s_m \leftarrow s'_m$. and $v_m \leftarrow v'_m$. . Then we have $g = s_1 s_2 \ldots s_m v_m u_{t+2} \ldots u_n$. We should note that $s_m \in G_1$ and $u_{t+2} \in G_2$ if it exists (as $u_t \in G_2$).

In the case that s_m is a representative of G_2 and u_t is in G_1, we do the dual procedure. Then go to Step 1).

At each stage of Step 2), the number of u_k's is reduced. Hence, the algorithm ends within at most $2n+1$ steps if the length of the input is n. Therefore, Algorithm 1 takes only linear time.

4 The modular group

The group of 2×2 matrices over rational integers with determinant 1 is called the modular group and denoted by $SL(2, Z)$, that is,

$$SL(2,Z) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \middle| a,b,c,d \in Z,\ ad - bc = 1 \right\}.$$

This group appears often in the literature of number theory, complex analysis, hyperbolic geometry, discrete group theory and combinatorial group theory. The modular group has been studied profoundly, and hence, we have a lot of technology provided in those areas of mathematics toward creating cryptosystems using it. For more information on the modular group, we refer the reader to [6] and[13].

Let A and B be the matrices in $SL(2, Z)$ given by

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

Furthermore, it is known that A and B generate $SL(2, Z)$. As a matter of fact,$SL(2, Z)$ has the presentation

$$Gp(A,B\,|\,A^6=B^4=1, A^3=B^2).$$

This simply implies that $SL(2, Z)$ is the free product of the cyclic group $<A>$ of order 6 and the cyclic group $<B>$ of order 4 amalgamating the cyclic group $H = <A^3> = <B^2> = \{I, -I\}$, of order 2. Therefore, every element of $SL(2, Z)$ is uniquely written as a normal form. We choose

$$\{I, A, A^2\}$$

as the set of coset representatives of H in $<A>$. We choose

$$\{I, B\}$$

as the set of coset representatives of H in $<B>$. Then every element in $SL(2, Z)$ is uniquely written as

$$s_1 s_2 \ldots s_n$$

where s_n is in H and each $s_k (k = 1, 2, ..., n\text{-}1)$ is A, A^2 or B such that if s_k is in $\{A, A^2\}$, then s_{k+1} is in $\{B\}$ and vice versa. We note that $s_n = \pm I$ since $s_n \in H = \{I, -I\}$.

We now note that there are infinitely many choices for matrices A and B. We show how to find matrices A_1 and B_1 that generate $SL(2, Z)$ subject to the relations $A_1^6 = B_1^4 = 1$ and $A_1^3 = B_1^2$. The followings are proved in[14].

Proposition 2

For a matrix $M \in SL(2, Z)$, the matrices $A_1 = M^{-1}AM$ and $B_1 = M^{-1}BM$ generate $SL(2, Z)$ and satisfy the relations $A_1^6 = B_1^4 = 1$ and $A_1^3 = B_1^2$.

Proposition 3

There are infinitely many distinct conjugates of A and B

By the previous two propositions, there are infinitely many choices for the matrices A_1 and B_1 generating $SL(2, Z)$ and subject to the relations $A_1^6 = B_1^4 = 1$and $A_1^3 = B_1^2$. We now review the action of the modular group on the upper half plane of the Gaussian plane. We denote the upper half plane by H, that is,

$$\mathrm{H} = \{z \in C\,|\,\mathrm{Im}(z) > 0\}$$

where C is the field of all complex numbers and Im(z) is the imaginary part of the complex number z. Let M be a matrix in $SL(2, Z)$. A fractional linear (*Möbius*) transformation f_M determined by the matrix M is given by:

For $z \in C$ \$z,

$$f_M(z)=\frac{az+b}{cz+d}$$

where

$$M=\begin{pmatrix}a & b\\ c & d\end{pmatrix}$$

It is easy to see that for $z \in H$ we have $f_M(z) \in H$. A group action of $SL(2, Z)$ on $SL(2, Z)$ is naturally induced as follows:

For M in $SL(2, Z)$ and $z \in H$,

$$Mz = f_M(z).$$

Obviously $SL(2, Z)$ acts on H in terms of fractional linear transformation. The equivalence relation on H is induced by the group action as follows: For $z_1, z_2 \in C$, $z_1 \sim z_2$ if there is $M \in SL(2, Z)$ such that $Mz_1 = z_2$. We refer the interested reader to[6] and[13] for the details of the action of the modular group on the upper half plane H. We now give a geometrical algorithm that finds the normal form (up to $\pm I$) for a given matrix $M \in SL(2, Z)$ with respect to the matrices A and B. We define several regions on H(see Fig.1).

Let O be the region

$$\{z \in C\,|\,\mathrm{Re}(z)| \le 1/2, 1 \le |z|\}.$$

Let P be the region

$$\{z \in C\,|\,\mathrm{Re}(z)| \ge 1/2, 1 \le |z|\}.$$

Let Q be the region

$$\{z \in C\,|\,1 \ge |z|, 1 \ge |z-1|\}.$$

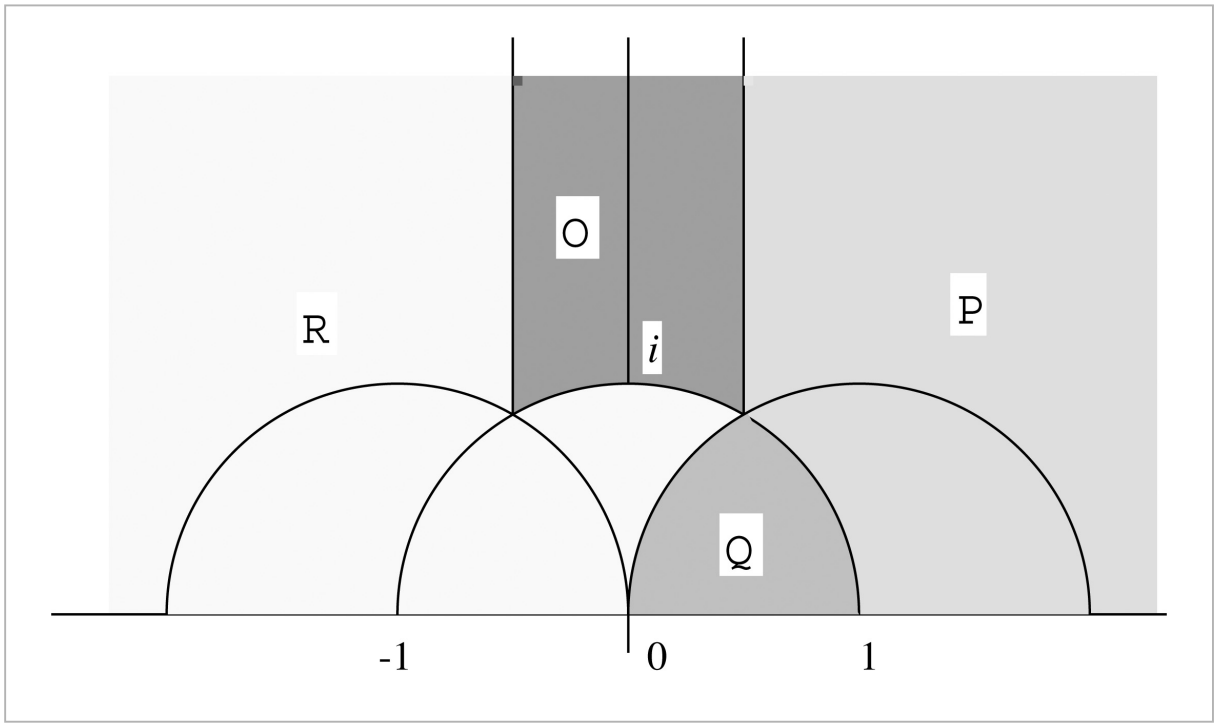

Fig.1 *Upper Half Plane*

Let R be the region

$$\{z \in C | 1 \geq |z|, 1 \leq |z-1|\} \cup \{z \in C | \mathrm{Re}(z) \leq -1/2\}.$$

We note that O is the fundamental domain. (See[6] or[13]) for more details of the fundamental domain.

We now describe the algorithm that for a given point z∈H which is equivalent to $y \in$O finds the matrix N such that $Nz = y$ and its normal form using geometry on the upper half plane.

Algorithm 2

INPUT: A point $z \in$H which is equivalent to the point y in the interior of O.

OUTPUT: The matrix N such that $Nz = y$ and its normal form with respect to A and B.

Step 0)

Let z be the given point. Let L be the empty list ( ).

Step 1)

If z is in O, then return L and the algorithm ends. Otherwise go to Step 2).

Step 2)

If z is in P, then set

$$z \Leftarrow A^{-1}z$$

and push A into L from the right hand side, that is,

$$L \Leftarrow (X_1, X_x, ..., X_n, A)$$

if $L = (X_1, X_x, ..., X_n)$ where X_i is A, A^2 or B.

If z is in Q, then set

$$z \Leftarrow A^{-1}z$$

and push A^2 into L from the right hand side, that is,

$$L \Leftarrow (X_1, X_x, ..., X_n, A^2)$$

if $L = (X_1, X_x, ..., X_n)$.

If z is in R, then set

$$z \Leftarrow B^{-1}z$$

and push B into L from the right hand side, that is,

$$L \Leftarrow (X_1, X_x, ..., X_n, B)$$

if $L = (X_1, X_x, ..., X_n)$.

Then go to Step 1).

Proposition 4

The algorithm above stops within $2n + 1$ steps if the length of the normal form for N is n. Moreover, if $L = (X_1, X_x, ..., X_n)$ where X_k is A, A^2 or B, then the normal form for N with respect to A and B is $X_1, X_x, ..., X_n$ up to $\pm I$.

Proof: We note that A and B generate $SL(2, Z)$ and that O is a fundamental domain of H. It follows that every point p on the upper half plane can be written as

$$p = Mq$$

where q is in O and $M \in SL(2, Z)$.

Furthermore, it is easy to verify that

$$AO \subset P \qquad AR \subset P \qquad AP \subset Q \qquad AQ \subset R \cup O$$

and

$$BO \subset R \qquad BP \subset R \qquad BQ \subset R \qquad BR \subset O \cup P \cup Q.$$

Suppose that N is in $SL(2, Z)$ and that its normal form is $X_1, X_x, ..., X_n$ where X_k is A, A^2 or B for each $k = 1, 2, ..., n$ up to $\pm I$. Take an arbitrary point y from O. We can obtain information of the first letter of the normal form by the position of the point Ny on the upper half plane. If X_1 is A, then Ny must lie in P. If X_2 is A^2, then Ny must lie in Q. If X_1 is B , then Ny must lie in R. For instance, if $X_1X_2 = AB$, then Ny must be in P and we obtain $X_1 = A$ and $X_2 = B$. Similarly we can deduce in other cases. We should note that the algorithm ends exactly in n steps if the length of the normal form is n.

To find the matrix N and its normal form with respect to A and B, one can employ the standard reduction algorithm (Algorithm 7.4.2. in[2]) to find a decomposition of the matrix into some matrices and Algorithm 1, however, Algorithm 2 seems much faster than the combination of the reduction algorithm and Algorithm 1. We should also remark that since we can find the normal form for a matrix $M \in SL(2, Z)$ with respect to the matrices A and B within liner time using Algorithm 2, we can also find the normal form for M with respect to the other generators A_1 and B_1 of $SL(2, Z)$ satisfying the relations $A_1^6 = 1 = B_1^4$ and $A_1^3 = B_1^2$ by using Algorithm 1 and Algorithm 2 consecutively within linear time.

5 A functional cryptosystem using the modular group

Let us define two backward deterministic systems using the action of $SL(2, Z)$ on the upper half plane and apply the scheme of functional cryptosystems in Section **2**. Let A_1 and B_1 be generators of $SL(2, Z)$ subject to $A_1^6 = B_1^4 = 1$ and $A_1^3 = B_1^2$. We have seen that there are infinitely many choices for A_1 and B_1. We choose a word V_1, V_2 on letters A_1 and B_1 such that V_1 and V_2 generate a free subsemigroup of $SL(2, Z)$, that is, if two words X_1 and X_2 on V_1 coincides with V_2 as elements of $SL(2, Z)$, then X_1 and X_2 are identical as words on V_1 and V_2. Furthermore, we require that every concatenation of V_1 and V_2 is in the normal form with respect to A_1 and B_1, that V_1 is not an initial segment of V_2 and that V_2 is not an initial segment of V_1. For example, the matrices $(B_1 A_1)^i$ and $(B_1 A_1^2)^j$ form a freesubsemigroup of $SL(2, Z)$ for all positive integers i and j and satisfy our requirements. It is easy to find such a pair of matrices in general using the combinatorics on words. We choose a matrix M arbitrarily from $GL(2, C)$ and set

$$W_1 = M^{-1}V_1M \qquad W_2 = M^{-1}V_2M$$

Recall that $GL(2, C)$ is the group of all 2×2 invertible matrices on the complex number field C. We note that W_1 and W_2 are $SL(2, C)$ since for each $i = 1, 2$ we have

$$\det(W_i) = \det(M^{-1}V_iM) = \det(M^{-1})\det(V_i)\det(M) \$$
$$= \frac{1}{\det(M)}\det(V_i)\det(M) = \det(V_i) = 1$$

We should note that $SL(2, C)$ acts on the upper half plane H in the same way as $SL(2, Z)$ acts on H in terms of fractional linear transformations. Let $\chi = M^{-1}H = \{M^{-1}q | q \in H\}$. Let p be a point on χ such that the point Mp is in the interior of the fundamental domain O.

Therefore $SL(2, Z)$ acts faithfully on Mp up to $\pm I$, that is, if $LMp = NMp$ for $L, N \in SL(2, Z)$ then we have $L = \pm N$. Let $f_M : \chi \rightarrow H$ be the fractional linear mapping defined by

$f_M(q) = Mq$. Let $G = M^{-1}SL(2, Z)M$. The homomorphism $\phi : G \to SL(2, Z)$ is given by $\phi(N) = MNM^{-1}$. Then it is easy to see that $f_M(Nx) = \phi(N)f_M(x)$ for each $N \in G$ and $x \in \chi$. We can easily verify that $(\{W_1, W_2\}, p, \chi)$ and $(\{V_1, V_2\}, f_M(p), \mathrm{H})$ are backward deterministic using the uniqueness of normal forms of a matrix in the modular group. Obviously f_M is a morphism between them. We follow the scheme described in Section **2** to build a functional cryptosystem using these backward deterministic systems.

Public-key:

The public-key is the backward deterministic system $(\{W_1, W_2\}, p, \chi)$.

Private-key:

The private-key is the backward deterministic system $(\{V_1, V_2\}, f_M(p), \mathrm{H})$.

We suppose that the plaintext to be sent is the sequence $i_1\ i_2\ ...\ i_n$ where $i_k \in \{1, 2\}$ for $k = 1, 2, ..., n$.

Encryption method:

Compute the matrix $Wi_1, Wi_2, ..., Wi_n$ and call this matrix E. We note that

$$E = M^{-1}V_{i_1}MM^{-1}V_{i_2}M...M^{-1}V_{i_n}M = M^{-1}V_{i_1}V_{i_2}...V_{i_n}M$$

Then, let E act the point p on χ by the fractional linear mapping determined by the matrix E. Compute the point $f_E(p) = Eq$ and call it q, that is, $q = Ep$. Since G acts on χ, the point q is on χ. Now the point q is sent to a legal receiver. Therefore q is the cryptotext for the original message $i_1\ i_2\ ...\ i_n$.

Decryption method:

Employing Algorithm 2, the legal receiver finds the normal form $X_1, X_x, ..., X_l$ where X_k is A or A^2 or B for $k = 1, 2, ..., l$ such that $Mq = X_1\ X_2\ ...\ X_l(MP)$. We denote the matrix $X_1, X_x, ..., X_l$ by N. Hence, $Mq = N(MP)$. Since $SL(2, Z)$ is generated by A and B, both A and B are written as products of matrices A_1 and B_1. We suppose that $A = Z_1(A_1, B_1)$ and $B = Z_2(A_1, B_1)$ where $Z_1(A_1, B_1)$ and $Z_2(A_1, B_1)$ are words on A_1 and B_1. By substituting $Z_1(A_1, B_1)$ for A and $Z_2(A_1, B_1)$ for B, respectively, the legal receiver gets

$$N = Z_{j_1}(A_1, B_1)Z_{j_2}(A_1, B_1)...Z_{j_l}(A_1, B_1)$$

where j_k is 1, if X_k is A and j_k is 2, if X_k is B. Employing Algorithm 1, the legal receiver obtains the normal form of N with respect to A_1 and B_1. By the uniqueness of expression of the normal form and our requirements on V_1 and V_2, the legal receiver obtains the sequence $Vi_1, Vi_2\ ...\ Vi_n$, and hence, the original plaintext $i_1\ i_2\ ...\ i_n$.

6 Security issues

We briefly discuss security issues in this section. Since the encryption and decryption depend on the free semigroup structures of subsemigroups of corresponding groups and the conjugation by the elements of $GL(2, C)$ preserves the freeness of subsemigroups, an eavesdropper may want to find a matrix N such that NW_1N^{-1}, NW_2N^{-1} are in $SL(2, Z)$. If the eavesdropper may be able to use Algorithm 1 and Algorithm 2 to break the cryptosystem. To find such a matrix N it is necessary to solve a system of matrix equations

$$NW_1N^{-1} = U \qquad NW_2N^{-1} = V$$

where U, V, N are unknown such that $U, V \in SL(2, Z)$ and $N \in GL(2, C)$. This system consists of 11 equations of 12 variables over the field of complex numbers. We note that if N is found then V U, are automatically derived. There are infinitely many solutions for this system of equations in principle, because the number of the variables is larger than the number of the equations. We know a solution, that is, the matrices M, V_1 and V_2 form one of the solutions. There is no known algorithm to solve the system of equations of this type as far as the author knows. Numerical analysis method may be able to work to solve the system of equation, however, it gives just an approximation of the solution N. Hence, we do not know whether or not numerical analysis method really works. Moreover, we can possibly avoid such an attack by restrict the

field of complex numbers to a finite extension field of the field of rational numbers. It is possible to realize the field operation of a splitting field of an irreducible polynomial over the field of rational numbers on computers. We should also note that N is not necessarily equal to M and that if N is distinct from M, the eavesdropper still has a problem to decrypt the message because N does not necessarily yield free generators of a free subsemigroup of $SL(2, Z)$ satisfying our requirements. For, even if matrices U_1 and U_2, words on generators A_2 and B_2 of $SL(2, Z)$ subject to the relations $A_2^6 = 1 = B_2^4$ and $A_2^3 = B_2^2$, form a set of free generators of a free subsemigroup, a concatenation of them is not necessarily in the normal form with respect to A_2 and B_2, and hence, there is still a trouble to retrieve the plain text.

Another possible attack is to find the matrix E and decompose it directly to the product of W_1 and W_2. There might be a smart way to find and decompose the matrix E. Of course, if the matrix E is found, then the eavesdropper can decompose E by guessing the decomposition and then checking whether or not it gives the correct answer. However, this is a non-deterministic polynomial time algorithm and so takes exponential time. Hence, it is slow for the breaking the system. Therefore, the backward deterministic system (W_1, W_2, p, χ) is considered intractable. On the other hand, the backward deterministic system $(V_1, V_2, f_M(p), \mathrm{H})$ is tractable because we can employ geometry of the upper half plane. In mathematics, geometry often provides a fast algorithm as Algorithm 2. The first backward system is associated to the space χ which is intractable, on the other hand, the second system is associated to the upper half plane that we have good understanding. The difference between the two systems lies in geometry.

7 Conclusion

We explain attempts in[14][15]. There are several research on attacks on the proposed systems. We would like to take into consideration such attacks and make a progress.

References

1 L.M.Adleman, "Molecular computation of solutions to combinatorial problems", Science, Vol.266, pp.1021-1024, Nov.11, 1994.

2 D.E.Cohen, "Combinatorial Group Theory : A Topological Approach", Cambridge University Press, 1989.

3 H.Cohen, "A Course in Computational Algebraic Number Theory", Springer-Verlag, New York, 1996.

4 J.Kari, "A cryptoanalytic observation concerning systems based on language theory", Discr. Appl. Math., Vol.21, pp.265-268, 1988.

5 J.Kari, "Observations concerning a public-key cryptosystem based on iterated morphisms", Theor. Compt. Sci., 66, pp.45-53, 1989.

6 N.Koblitz, "Introduction to Elliptic Curves and Modular Forms", Springer-Verlag, New York, 1991.

7 R.C.Lyndon and P. E. Schupp, "Combinatorial Group Theory", Springer-Verlag, New York, 1976.

8 P.Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", SIAM J. Comp., Vol.26, pp.1484-1509, 1997.

9 J.J.Rotman, "An Introduction to Theory of Groups", Springer, New York, 1995.

10 A.Salomaa, "A public-key cryptosystem based on language theory", Computers and SecurityVerlag, Vol.7, pp.83-87, 1988.

11 A.Salomaa, "Public-Key Cryptography", Springer-Verlag, Berlin, 1990.

12 A.Salomaa and S. Yu, "On a public-key cryptosystem based on iterated morphisms and substitutions", Theor. Compt. Sci., Vol.48, pp.283-296, 1986.

13 J-P.Serre, “A Course in Arithmetic, Springer-Verlag, New York, 1973.
14 A.Yamamura, “Public-key cryptosystems using the modular group”, International Workshop on Practice and Theory in Public Key Cryptography, LNCS, Vol.1431, Springer-Verlag, pp.203- 216, 1998.
15 A.Yamamura, “A functional cryptosystem using a group action”, Information Security and Privacy (ACISP99), LNCS, Springer-Verlag, Vol.1587, pp.314-325, 1999.

***YAMAMURA Akihiro**, Ph.D.*

Group Leader, Security Fundamentals Group, Information and Networks Systems Department

Information security, Cryptography, Algebraic systems and their algorithms