On: 21 December 2014, At: 11:18
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

Journal of Discrete Mathematical Sciences and Cryptography

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tdmc20

Key-lock pair mechanism for access control using tribes of Farey fractions

H. Chandrashekhar [a] & M. Nagaraj [a]

[a] Department of Mathematics , Bangalore University , Bangalore , 560 001 , India

Published online: 03 Jun 2013.

To cite this article: H. Chandrashekhar & M. Nagaraj (2000) Key-lock pair mechanism for access control using tribes of Farey fractions, Journal of Discrete Mathematical Sciences and Cryptography, 3:1-3, 11-22, DOI: 10.1080/09720529.2000.10697895

To link to this article: http://dx.doi.org/10.1080/09720529.2000.10697895

Key-lock pair mechanism for access control using tribes of Farey fractions

H. Chandrashekhar

M. Nagaraj

Department of Mathematics

Bangalore University

Bangalore-560 001

India

ABSTRACT

We propose a new single key-lock mechanism based on the concept of an access control matrix. In this system, each file is given a lock and each user is given a key and through simple operations on keys and locks the user access privilege can be revealed. We use Chinese Remainder theorem and tribes of Farey fractions in this method instead of the method based on Euler's Theorem of Number Theory used by Chang [2]. An advantage of our method is the ease with which coding can be done for the locking mechanism and for the much larger number of users and files.

0. INTRODUCTION

Recently, time-sharing computer systems, computer networks and distributed network systems have permitted large number of users to share common databases. Because of this people have begun to be concerned with the rising importance of information security.

It is generally agreed that some kind of information protection measure is required to prevent disclosures to unauthorized persons. An access control mechanism grants the privilege to access information resources in the system to a user. For instance, users may be able to access filed via READ, WRITE, EXECUTE, DELETE OR APPEND commands, but different users will be given different access rights to individual files. Traditionally this can be achieved by using an access control matrix which specifies who has what access privileges to system resources (see Chang [2]).

Journal of Discrete Mathematical Sciences & Cryptography
Vol. 3 (2000), Nos. 1-3, pp. 11-22

Sharing segments of data or programs becomes more and more important in today's computing systems. For example, one might selectively let others use a data segment that he has stored or a routine that he has developed. Therefore, as sharing is inevitable, it is important to let the system know which users are allowed to what degree of access on which segments of data or programs, so that data with certain privacy would be at the disposal of right users. An access control matrix can represent privacy decisions on the relationships of users to files. Let us consider the matrix as shown below:

Table 0.1

File

User	1	2	3	4
Rao	Read	Own	Write	Read
Suresh	Write	Read		Own
Ramesh	Read	Write	Own	

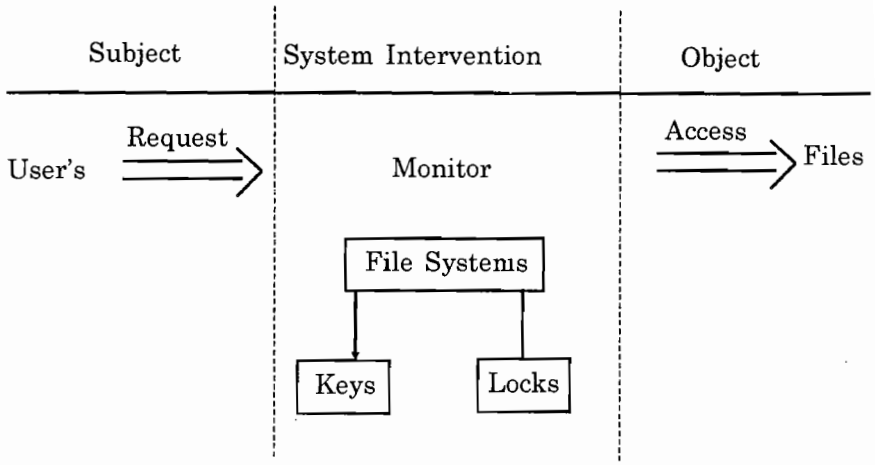
Table 0.1 illustrates the access control matrix of a simple information protection system with three users and four files (i.e., segments of data or programs). In this case we can see that Rao owns file 2 which Suresh can read and Ramesh can write. Ramesh might grant Suresh the right to write on file 3 or Ramesh might request to read file 4 which is owned by Suresh.

The concept and the implementation of access control matrix seem to be simple and easy. But we cannot store the access control matrix because it will tend to be sparse and large when the system grows large.

Now we shall describe the key-lock pair mechanism which is proposed by Wu and Hwang [4].

For a user, every attempt to access a file is intercepted and validated by the file system. We assume that there are a fixed number n of files to be protected. The file system assigns a lock for each file. When a new user joins the system his access rights to these n files are decided and the file system generates a key for each user according to the locks and his access rights.

The organization of the single key-lock system is as depicted in the following figure:



Organisation of the SKL system

Figure 0.2

We now review Wu and Hwang's key-lock pair mechanism that fulfils the requirement of the single key-lock (SKL for short) system [4]. Each user is given a key and each file a lock by the file system. An operation on the key of user i with the lock of file j yields the attribute value in the (i, j)th entry of the invisible access control matrix. The construction procedures of keys and locks and the operations of keys and locks are better illustrated through a simple example:

We assume the state of a simple system having five users and six files as shown below:

Table 0.3

	File j					
User i	1	2	3	4	5	6
1	2	4	1	4	4	0
2	1	2	2	1	2	1
3	2	0	3	3	3	0
4	1	1	2	1	2	3
5	3	0	4	2	2	1

0. No access
1. Execute
2. Read
3. Write
4. Own.

In this case access by user i to file j is allowed only when the request of the privilege matches the attribute value a_{ij}, which is the (i, j)th entry in the access control matrix as shown above. Here, we note that a linear hierarchy of access privileges may optionally be implied, as is the case in our example. That is, the access is allowed only if the request privilege is smaller than or equal to the value of a_{ij}. Therefore, the right to read implies the right to execute, the right to own implies the right for all and so on.

Wu and Hwang [4] devised a method to assign each user U_i a key K_i of n-digit sequence and each file F_j a lock L_j of n-digit sequence too, where n is the total number of files. In this method the attribute value a_{ij} can be evaluated as $a_{ij} = K_i * L_j$ where the operator * means the inner product over Galois field $GF(t)$ and t is chosen as the smallest prime number that is larger than all attribute members of the access control matrix considered. (See C. C. Chang [2])

In the example above, since all a_{ij}'s have values less than 7, we may assume that the access control matrix is over finite field $GF(7)$. Now a 5×5 non-singular matrix with pseudo-random entries over $GF(7)$ is chosen as follows:

$$K = \begin{bmatrix} 4 & 2 & 1 & 3 & 0 \\ 5 & 0 & 1 & 2 & 0 \\ 3 & 4 & 0 & 1 & 3 \\ 2 & 0 & 1 & 3 & 4 \\ 2 & 1 & 1 & 1 & 4 \end{bmatrix}.$$

At this moment we choose the rows of matrix K as the keys for the user. That is

$K_1 = (4, 2, 1, 3, 0)$

$K_2 = (5, 0, 1, 2, 0)$

$K_3 = (3, 4, 0, 1, 3)$

$K_4 = (2, 0, 1, 3, 4)$

$K_5 = (2, 1, 1, 1, 4)$.

For evaluating the lock of F_1, we assume that $L_1 = (x_1, x_2, x_3, x_4, x_5)$. Thus we have the following five equations:

$2 = 4x_1 + 2x_2 + 1x_3 + 3x_4 + 0x_5$

$1 = 5x_1 + 0x_2 + 1x_3 + 2x_4 + 0x_5$

$$2 = 3x_1 + 4x_2 + 0x_3 + 1x_4 + 3x_5$$

$$1 = 2x_1 + 0x_2 + 1x_3 + 3x_4 + 4x_5$$

$$3 = 2x_1 + 1x_2 + 1x_3 + 1x_4 + 4x_5$$

Over $GF(7)$, solving these equations we have $L_1 = (x_1, x_2, x_3, x_4, x_5) = (4, 2, 3, 0, 1)$. Similarly, we determine

$$L_2 = (1, 1, 2, 1, 2)$$

$$L_3 = (6, 1, 1, 3, 2)$$

$$L_4 = (2, 5, 1, 2, 1)$$

$$L_5 = (3, 2, 6, 1, 2)$$

$$L_6 = (2, 0, 3, 1, 0).$$

To check its correctness, let us consider

$$a_{36} = K_3 * L_6$$

$$= (3, 4, 0, 1, 3) * (2, 0, 3, 1, 0)$$

$$= 7$$

$$= 0 \text{ over } GF(7)$$

which is correct.

Three big disadvantages Wu and Hwang's [4] method are 1) the size of required storage due to the keys and locks actually exceeds that of the original access control matrix, 2) the operations of keys and locks are tedious, 3) the constructions of keys and locks are not simple. C. C. Chang [2] has proposed another single key-lock pair mechanism using Euler's theorem of number theory. In this method the construction of keys is not simple. Therefore we intend to develop a new key-lock pair mechanism using tribes of Farey fractions as described in the next section 1.

1. KEY-LOCK PAIR MECHANISM USING TRIBES OF FAREY FRACTIONS

In this section we present a mechanism that fulfils the requirement of the key-lock protection system. We suppose that each user

U_i is given a key K_i, each file F_j a lock L_j and an operation on the L_j with K_i yields the attribute a_{ij} in the (i, j)th entry of the access control matrix.

Let there be n files and m users. Let the access control matrix be as follows:

Table 1.1

Files

Users	1	2	3	...	n
1	a_{11}	a_{12}	a_{13}		a_{1n}
2	a_{21}	a_{22}	a_{23}		a_{2n}
3	a_{31}	a_{32}	a_{33}		a_{3n}
. . .					
m	a_{m1}	a_{m2}	a_{m3}		a_{mn}

where the attribute a_{ij} denotes the access for the user U_i to the file F_j.

We use Chinese Remainder Theorem in our method.

THEOREM 1.1. (Chinese Remainder Theorem) *Let $n_1, n_2, \ldots, n_r$ be positive integers such that $gcd(n_i, n_j) = 1$ for $i \neq j$. Then the system of linear congruences*

$$x \equiv a_1 (mod\ n_1)$$

$$x \equiv a_2 (mod\ n_2)$$

$$\vdots$$

$$x \equiv a_r (mod\ n_r)$$

has a simultaneous solution, which is unique modulo the integer $n_1 n_2 \ldots n_r$.

Example 1.2. Let the system of three congruences be

$$x \equiv 2 (mod\ 3)$$

$$x \equiv 3 (mod\ 5)$$

$$x \equiv 2 (mod\ 7).$$

Here $n_1 = 3$, $n_2 = 5$, $n_3 = 7$ are mutually relatively prime integers.

Let

$$n = n_1 n_2 n_3 = 3 \times 5 \times 7 = 105$$

and

$$N_1 = \frac{n}{n_1} = \frac{n}{3} = 35$$

$$N_2 = \frac{n}{n_2} = \frac{n}{5} = 21$$

$$N_3 = \frac{n}{n_3} = \frac{n}{7} = 15.$$

Now the linear congruences

$$35x \equiv 1 (mod\, 3)$$

$$21x \equiv 1 (mod\, 5)$$

$$15x \equiv 1 (mod 7)$$

are satisfied by

$$x_1 = 2, \quad x_2 = 1, \quad x_3 = 1$$

respectively.

Thus a solution of the system is given by

$$x = 2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1$$

$$= 233.$$

Modulo 105, we get the unique solution

$$x = 233$$

$$= 23 (mod\, 105).$$

Key-Lock Mechanism 1.3

We select n mutually relatively prime positive integers $l_1, l_2, ..., l_n$ different from attribute values a_{ij}. Each file F_j is assigned an integer l_j from the set above, while is treated as its lock.

The attributes $a_{i1}, a_{i2}, a_{i3}, ..., a_{in}$ will determine the key number for the user U_i. for this, we consider the following system of linear congruences:

$$\begin{aligned} x &\equiv a_{i1}(mod\, l_1) \\ x &\equiv a_{i2}(mod\, l_2) \\ &\vdots \\ x &\equiv a_{in}(mod\, l_n). \end{aligned} \tag{1.1}$$

Let the solution of the system (1.1) be λ_i. We call this number the **key number** for the user U_i.

For each user U_i, the unique integer λ_i is determined. Now, we encode this integer λ_i by using tribe of either Gaussian Farey fraction $\frac{\alpha_i}{\beta_i}$ or rational Farey fraction $\frac{h_i}{k_i}$ and the key k_i is determined. We describe the method using the tribe of Gaussian Farey fraction.

We associate the Gaussian Farey fraction $\frac{\alpha_i}{\beta_i}$ to each User U_i. The characteristic equations of $\frac{\alpha_i}{\beta_i}$ are given by

$$\beta_i\varepsilon_i - \alpha_i\eta_i = \pm\mathbf{1}, \quad \pm\mathbf{i}. \tag{1.2}$$

We consider the characteristic equation,

$$\beta_i\varepsilon_i - \alpha_i\eta_i = 1. \tag{1.3}$$

Let $\alpha_i = a_i + \mathbf{i}b_i$ and $\beta_i = c_i + \mathbf{i}d_i$ be two Gaussian integers where $a_id_i - b_ic_i = \mathbf{1}$.

We have the **tribe** of $\frac{\alpha_i}{\beta_i}$ as

$$T_{\frac{\alpha i}{\beta i}} = \left\{ \frac{-b_i + \lambda\alpha_i}{-d_i + \lambda\beta_i}, \lambda \in J[\mathbf{i}] \right\} \tag{1.4}$$

(See Chandrashekhar and Nagaraj [1] and Nagaraj and Srinivasa Murthy [3]).

We assign the key k_i to the user U_i as

$$k_i = \frac{-b_i + \lambda_i\alpha_i}{-d_i + \lambda_i\beta_i} = \frac{\varepsilon_i}{\eta_i} \tag{1.5}$$

by taking $\lambda = \lambda_i$, the key **number**.

Therefore, the key for the user U_i is

$$k_i = (\varepsilon_i, \eta_i). \tag{1.6}$$

We define

$$k_i * l_j = \frac{\beta_i(\varepsilon_i + b_i) + \alpha_i(\eta_i + d_i)}{2\alpha_i\beta_i} \,(mod\, l_j) \tag{1.7}$$

$$= \lambda_i \,(mod\, l_j)$$

$$= a_{ij}.$$

Thus, the attribute a_{ij} can be evaluated with key k_i and lock l_j which is the access for the user U_i to the file F_j.

Similarly we can construct keys using tribes of rational Farey fractions.

This is our new single key-lock pair mechanism.

Remark 1.4. If we use tribes of rational Farey fractions $\frac{h_i}{k_i}$ (See M. Nagaraj and R. Srinivasa Murthy [3]) to encode the key number λ_i then

$$k_i * l_j = \frac{k_i(\varepsilon_i - x_{0i}) + h_i(\eta_i - y_{0i})}{2h_ik_i} \,(mod\, l_j)$$

$$= \lambda_i (mod\, l_j)$$

$$= a_{ij}.$$

Example 1.5. The following table illustrates the state of a simple system having three users and three files.

Table 1.2

	File j		
User i	1	2	3
1	2	4	1
2	1	2	2
3	2	0	3

Let $l_1 = 5$, $l_2 = 7$ and $l_3 = 11$.

By Chinese Remainder Theorem, we calculate the key numbers λ_1, λ_2 and λ_3 for the users U_1, U_2, U_3 respectively.

The solution of the system of linear congruences

$$x \equiv 2(mod\ 5)$$
$$x \equiv 4(mod\ 7)$$
$$x \equiv 1(mod\ 11)$$

is $x = 1992$

$= \lambda_1$.

The solution of the system of linear congruences

$$x \equiv 1(mod\ 5)$$
$$x \equiv 2(mod\ 7)$$
$$x \equiv 2(mod\ 11)$$

is $x = 1311$

$= \lambda_2$.

The solution of the system of linear congruences

$$x = 2(mod\ 5)$$
$$x \equiv 0(mod\ 7)$$
$$x \equiv 3(mod\ 11)$$

is $x = 1092$

$= \lambda_3$.

(i) *Keys using the tribes of Gaussian Farey fractions*

We associate arbitrary Gaussian Farey fractions $\frac{\alpha_i}{\beta_i}$ with users U_i.

Let $\frac{\alpha_1}{\beta_1} = \frac{1+2\mathbf{i}}{2+5\mathbf{i}}$, $\frac{\alpha_2}{\beta_2} = \frac{3+\mathbf{i}}{5+2\mathbf{i}}$ and $\frac{\alpha_3}{\beta_3} = \frac{5+2\mathbf{i}}{7+3\mathbf{i}}$.

Now $k_1 = \frac{-2+\lambda_1\alpha_1}{-5+\lambda_1\beta_1}$

$= \frac{1990+3984\mathbf{i}}{3979+9960\mathbf{i}}$

$= (1990 + 3984\mathbf{i}, 3979 + 9960\mathbf{i})$

$= (\varepsilon_1, \eta_1)$.

Similarly $k_2 = (3932 + 1311\mathbf{i}, 6553 + 2622\mathbf{i})$

$= (\varepsilon_2, \eta_2)$

and $k_3 = (5458 + 2184\mathbf{i}, 7641 + 3276\mathbf{i})$

$= (\varepsilon_3, \eta_3)$.

Verification. From the equation (1.7)

$$k_1 * l_1 = \frac{\beta_1(\varepsilon_1 + b_1) + \alpha_1(\eta_1 + d_1)}{2\alpha_1\beta_1} (mod\ l_1)$$

$$= \frac{(2+5\mathbf{i})(1990+3984\mathbf{i}+2)+(1+2\mathbf{i})(3979+9960\mathbf{i}+5)}{2(1+2\mathbf{i})(2+5\mathbf{i})} (mod\ 5)$$

$$= 1992\ (mod\ 5)$$

$$= 2$$

which is access for file F_1 and for the user U_1. Similarly we can verify key k_1 with l_2 and l_3.

(ii) *Keys using the tribes of rational Farey fractions*

If we associate rational Farey fractions with users U_i then the encoded keys for the above example are as follows:

Let us associate, the Farey fractions $\frac{h_i}{k_i}$ for the users U_i.

Let $\frac{h_1}{k_1} = \frac{2}{5}$, $\frac{h_2}{k_2} = \frac{3}{5}$ and $\frac{h_3}{k_3} = \frac{5}{7}$.

Now $k_1 = \frac{x_{01} + \lambda_1 h_1}{y_{01} + \lambda_1 k_1}$

$= \frac{1 + 1992 \times 2}{2 + 1992 \times 5}$

$= \frac{3985}{9962}$

$= (3985, 9962)$

$= (\varepsilon_1, \eta_1)$.

Similarly $k_2 = 3935, 6558)$

$= (\varepsilon_2, \eta_2)$

and $k_3 = (5463, 7648)$

$= (\varepsilon_3, \eta_3)$.

Concluding Remarks 1.6

In this paper we have presented a new single key-lock pair mechanism. In our method we use Chinese Remainder Theorem for determining the key and the tribe of Gaussian Farey fraction for coding the key. The advantages of this method over the methods described by Wu and Hwang [4] and C. C. Chang [2] are:

1) The operations of keys and locks are not tedious.
2) The constructions of keys and locks are simple.

REFERENCES

1. H. Chandrashekhar and M. Nagaraj, Tribes of Gaussian Farey Fractions, *The Mathematics Student,* Vol. 63, Nos. 1-4 (1994), pp. 196-200.
2. C. C. Chang, An Information Protection Scheme Based upon Number Theory, *The Computer Journal,* Vol. 30, No. 3, 1987, pp. 249-253.
3. M. Nagaraj and R. Srinivasamurthy, Properties of Tribes of Farey Fractions, *J. Ramanujan Math. Soc.*, Vol. 4(i), 1989, pp. 25-31.
4. M. L. Wu and T. Y. Hwang, Access Control with Single Key-Lock, *IEEE Transactions on Software Engineering,* Vol. SE-10(2), pp. 185-191(1984).