Key agreement protocols for distributed secure multicast over the ring $E_p^{(m)}$

J.-J. Climent[1], J. A. López-Ramos[2], P. R. Navarro[3] & L. Tortosa[3]
[1]*Departament d'Estadística i Investigació Operativa,*
Universitat d'Alacant, Spain
[2]*Departamento de Matemáticas, Universidad de Almería, Spain*
[3]*Departament de Ciència de la Computació i Intel·ligència Artificial,*
Universitat d'Alacant, Spain

Abstract

Protocols for authenticated key exchange allow parties within an insecure network to establish a common session key which can then be used to secure their future communication. In this paper we introduce a protocol for distributed key agreement over a noncommutative ring with a large number of noninvertible elements. This protocol uses polynomials with coefficients in the center of the ring. We also present the necessary steps for recalculating the shared secret key when a new user joins the system, or when a user leaves the system.
Keywords: secure communications, key exchange, noncommutative ring, multicast protocol.

1 Introduction

The classical systems of cryptography all suffer about the well-known *key distribution problem*. This is the problem of establishing a private channel by means of which the sender and receiver of messages can exchange the key current in use. Diffie and Hellman [1] addressed this problem in their seminal paper in 1976. The security of this protocol is based on the problem of computing discrete logarithms in the multiplicative group of a finite field.

Most of the public key cryptosystems and public key exchange protocols are based, from the point of view of its security, on the difficulty to solve some number theory problems over finite commutative algebraic structures. Some efficient

WIT Transactions on Information and Communication Technologies, Vol 45, © 2013 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA130021

attacks have been proposed for many of the well-known protocols in the last decades. For example, Odoni *et al.* [2] propose the group of matrices over a finite field as base group for Diffie–Hellman key exchange; however, this model is cryptanalyzed by Menezes and Wu [3] using eigenvalues theoretic properties.

With the main objective to avoid the attacks over commutative structures, different models have been proposed in recent years. In 2007, Cao *et al.* [4] present a general key exchange protocol, whose security is based on the difficulty to solve the Symmetric Decomposition Problem over a noncommutative ring, using polynomials.

Traditional communication modes have been one-to-one or unicast, and one-to-all or broadcast. Among these two extremes we find multicast, the targeting of a single data stream to a select set of receivers, which may or may not include the sender. Therefore, we can say that multicasting is the ability to transmit a single stream to multiple subscribers at the same time.

There are three fundamental types of IPv4 addresses: unicast, broadcast, and multicast. A unicast address is designed to transmit a packet to a single destination. A broadcast address is used to send a datagram to an entire subnetwork. A multicast address is designed to enable the delivery of datagrams to a set of hosts that have been configured as members of a multicast group in various scattered subnetworks. Network-level IP multicast was proposed over a decade ago (see, for example, [5, 6]).

With the main objective to avoid the attacks over commutative structures, Climent *et al.* [7] study the ring $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$, for a prime p, and prove that it is isomorph to the ring E_p whose elements are 2×2 matrices, with entries in the first row belonging to $\mathbb{Z}_p$ and the entries in the second row belonging to $\mathbb{Z}_{p^2}$. Using this ring and the arithmetic implemented over it, some key exchange protocols are presented in [8, 9], using polynomials which coefficients are elements of the center of the ring. One of these protocols based on the noncommutative ring E_p was cryptanalyzed by Kamal and Youssef [10]. This cryptanalysis is based on the existence of a large number of invertible elements in E_p, when p is large enough.

To avoid this weakness Climent *et al.* [11] introduce an extension $E_p^{(m)}$ of E_p that maintains the main properties of E_p. In particular, it can not be embedded in a ring of matrices over a commutative ring. And most importantly, the number of noninvertible elements in $E_p^{(m)}$ is very large when m is large compared with p.

The rest of the paper is organized as follows. In Section 2 we recall some properties of the ring $E_p^{(m)}$. In Section 3 we introduce a multicast communication protocol initially defined over any noncommutative ring R, based on key exchanges developed by Climent *et al.* [8]. The implementation of the multicast protocol is performed over the ring $E_p^{(m)}$, due to its characteristics that make it safe to known attacks. In Section 4 we describe the process to join or leave the group and in Section 5 we give an example for a system with 4 users. Finally, we present our main conclusions in Section 6.

2 The ring $E_p^{(m)}$

Climent *et al.* [11] proved that the set

$$E_p^{(m)} = \Big\{ [a_{ij}] \in \mathrm{Mat}_{m\times m}(\mathbb{Z}) \ \mid \ a_{ij} \in \mathbb{Z}_{p^i} \text{ if } i \leq j, \\ \text{and } a_{ij} \in p^{i-j}\mathbb{Z}_{p^j} \text{ if } i > j \Big\}$$

is a noncommutative unitary ring with addition and multiplication given by

$$[a_{ij}] + [b_{ij}] = \left[(a_{ij} + b_{ij}) \bmod p^i\right],$$

$$[a_{ij}] \cdot [b_{ij}] = \left[\left(\sum_{k=1}^{m} a_{ik} b_{kj}\right) \bmod p^i\right],$$

respectively. That is, the addition and multiplication of the elements of $E_p^{(m)}$ is analogous to the addition and multiplication of $m \times m$ matrices with entries in $\mathbb{Z}$, with the particularity that the entries in the ith row are reduced modulo p^i, for $i = 1, 2, \ldots, m$. So, the mull matrix and the identity matrix are the additive and multiplicative identities of $E_p^{(m)}$. As we mentioned earlier in Section 1, the number of noninvertible elements in $E_p^{(m)}$ is very large when m is large compared with p. For example, for $p = 7$ and $m = 2, 4, 8, 16, 32$, the number of noninvertible elements is about $26.53\,\%$, $46.02\,\%$, $70.86\,\%$, $91.51\,\%$ and $99.28\,\%$ respectively (see [11]).

Moreover, this ring is not an integral domain. So it is not a left nor a right Euclidean ring and, consequently, the ring of polynomials with coefficients in $E_p^{(m)}$ is not Euclidean.

The center of this ring plays an important role in the protocol that we will introduce in Section 3 and it is characterized as

$$Z\left(E_p^{(m)}\right) = \Big\{ [a_{ij}] \in E_p^{(m)} \ \mid \ a_{ij} = 0 \text{ if } i \neq j, \\ \text{and } a_{ii} = \sum_{r=1}^{i} p^{i-r} u_{i-r} \text{ with } u_{i-r} \in \mathbb{Z}_p \Big\}.$$

So, $\left|Z(E_p^{(m)})\right| = p^m$. Furthermore, $Z\left(E_p^{(m)}\right)$ is not an Euclidean ring.

3 A multicast protocol over $E_p^{(m)}$

The communication model we propose in this multicast protocol over $E_p^{(m)}$ is based on the IP multicast framework. We need to perform communications in a restricted group, where all the components (*members* or *users*) of this restricted group will manage all rekeying operations by themselves.

Steiner *et al.* [12] introduce two new protocol that enhances the extension of the Diffie–Hellman key exchange for rekeying. Moreover, one of them, widely known as CLIQUES, is used for the same authors [13] for rekeying in Dynamic Peer Groups.

Before starting to describe the details of the protocol, we need to introduce the following notation for some sums:

$$\sigma(j,1,i)=\sum_{j=1}^{i-1} j \qquad \text{and} \qquad \delta(j,1,i,l)=\sum_{\substack{j=1\\ j\neq i-l}}^{i-1} j.$$

So let us assume that the set of users is given by $\{U_1, U_2, \ldots, U_h\}$. Then, users agree to use the noncommutative ring $E_p^{(m)}$. Furthermore, note that if we consider $f(\mathsf{x}), g(\mathsf{x}) \in Z\left(E_p^{(m)}\right)[\mathsf{x}]$ and $M \in E_p^{(m)}$, we have that

$$f(M)^r g(M)^s = g(M)^r f(M)^s, \quad \text{for all positive integers } r \text{ and } s, \tag{1}$$

although $E_p^{(m)}$ is not commutative. This property allows us to establish the following protocol.

Protocol 1: Let us assume that $M \in E_p^{(m)}$ and $K_0 = N \in E_p^{(m)} \setminus Z\left(E_p^{(m)}\right)$ are public. Every user U_i, for $i = 1, 2, \ldots, h$ chooses a polynomial $f_i(\mathsf{x}) \in Z\left(E_p^{(m)}\right)[\mathsf{x}]$ and a pair of positive integers r_i and s_i. Then $(r_i, s_i, f_i(\mathsf{x}))$ is the private key for the user U_i.

1. User U_1 computes the element K_1 of $E_p^{(m)}$ given by

$$K_1 = f_1(M)^{r_1} K_0 f_1(M)^{s_1}. \tag{2}$$

User U_1 sends K_1 to user U_2.

2. User U_2 computes the elements K_2 and K_3 of $E_p^{(m)}$ given by

$$\begin{aligned} K_2 &= f_2(M)^{r_2} K_0 f_2(M)^{s_2},\\ K_3 &= f_2(M)^{r_2} K_1 f_2(M)^{s_2}. \end{aligned} \tag{3}$$

User U_2 sends to user U_3 the 3-vector of elements in $E_p^{(m)}$ given by

$$(K_1, K_2, K_3).$$

3. User U_3 computes the elements K_4, K_5 and K_6 of $E_p^{(m)}$ given by

$$\begin{aligned} K_4 &= f_3(M)^{r_3} K_1 f_3(M)^{s_3},\\ K_5 &= f_3(M)^{r_3} K_2 f_3(M)^{s_3},\\ K_6 &= f_3(M)^{r_3} K_3 f_3(M)^{s_3}. \end{aligned} \tag{4}$$

User U_3 sends to user U_4 the 4-vector of elements in $E_p^{(m)}$ given by

$$(K_3, K_4, K_5, K_6)\,.$$

4. In general, for $i = 4, 5, \dots, h-1$, user U_i computes the elements of $E_p^{(m)}$

$$\begin{aligned} K_{i+\delta(j,1,i,l)} &= f_i(M)^{r_i} K_{\delta(j,1,i,l)} f_i(M)^{s_i}, \quad \text{for } l = 1, 2, 3, \dots, i-1, \\ K_{i+\sigma(j,1,i)} &= f_i(M)^{r_i} K_{\sigma(j,1,i)} f_i(M)^{s_i}. \end{aligned} \tag{5}$$

User U_i sends to user U_{i+1} the $(i+1)$-vector of elements in $E_p^{(m)}$ given by

$$\Big(K_{i-1+\delta(j,1,i,1)}, K_{i+\delta(j,1,i,1)}, K_{i+\delta(j,1,i,2)}, \\ \dots, K_{i+\delta(j,1,i,i-1)}, K_{\sigma(j,1,i+1)}\Big).$$

5. When user U_h receives the h-vector

$$\Big(K_{h-2+\delta(j,1,h-1,1)}, K_{h-1+\delta(j,1,h-1,1)}, K_{h-1+\delta(j,1,h-1,2)}, \\ \dots, K_{h-1+\delta(j,1,h-1,h-2)}, K_{\sigma(j,1,h)}\Big). \tag{6}$$

he/she computes the elements of $E_p^{(m)}$ given by

$$L_1^{(h)} = f_h(M)^{r_h} K_{h-2+\delta(j,1,h-1,1)} f_h(M)^{s_h}, \tag{7}$$

$$\begin{aligned} L_l^{(h)} &= f_h(M)^{r_h} K_{h-1+\delta(j,1,h-1,l-1)} f_h(M)^{s_h}, \\ &= f_h(M)^{r_h} K_{\delta(j,1,h,l)} f_h(M)^{s_h}, \quad \text{for } l = 2, 3, \dots, h-1, \end{aligned} \tag{8}$$

$$L_h^{(h)} = f_h(M)^{r_h} K_{\sigma(j,1,h)} f_h(M)^{s_h}. \tag{9}$$

User U_h sends to every user the $(h-1)$-vector $\left(L_1^{(h)}, L_2^{(h)}, \dots, L_{h-1}^{(h)}\right)$ of elements in $E_p^{(m)}$.

6. Finally, when every user U_i, for $i = 1, 2, \dots, h-1$, receives the $(h-1)$-vector $\left(L_1^{(h)}, L_2^{(h)}, \dots, L_{h-1}^{(h)}\right)$, he/she takes the $(h-i)$th entry, $L_{h-i}^{(h)}$, and computes the element

$$S_i = f_i(M)^{r_i} L_{h-i}^{(h)} f_i(M)^{s_i}. \tag{10}$$

User U_h denote by S_h the element $L_h^{(h)}$, i.e., $S_h = L_h^{(h)}$.

Next theorem establishes that the shared secret by all users is $L_h^{(h)}$.

Theorem 1: *With the notation of Protocol 1, it follows that*

$$S_1 = S_2 = \cdots = S_{h-1} = S_h. \tag{11}$$

Proof: Assume that $i = 1, 2, \ldots, h$. From expressions (2)–(5) and (7)–(9), and taking into account expression (1), it follows that

$$L^{(h)}_{h-i} = \left(\prod_{\substack{j=1\\ j\neq i}}^{h} f_j(M)^{r_j}\right) K_0 \left(\prod_{\substack{j=1\\ j\neq i}}^{h} f_j(M)^{s_j}\right). \tag{12}$$

Now, from expressions (10) and (12) and taking into account expression (1) again, it follows that

$$S_i = \left(\prod_{j=1}^{h} f_j(M)^{r_j}\right) K_0 \left(\prod_{j=1}^{h} f_j(M)^{s_j}\right).$$

So, expression (11) holds.

This protocol reduces considerably the number of messages and rounds. These are exactly h in both cases.

Let us remark finally that in the corresponding protocol given in [12] and [13], a simple division on a finite field would yield every power computed for every user and thus, to compromise every user's private key. In the case we are considering, this attack is not possible since the number of noninvertible elements in $E_p^{(m)}$ is very large when m is large compared with p as we stated in Section 2.

4 Join-leave operations

When a new user U_{h+1} joins the group a rekeying is needed in order to preserve backward secrecy.

Let us assume that U_h has stored the h-vector of elements in $E_p^{(m)}$ given by expression (6). Then the join operation will consist of the following steps:

1. User U_h generates a new polynomial $\hat{f}_h(\mathsf{x}) \in Z\left(E_p^{(m)}\right)$ as well as two new positive integers $\hat{r}_h$ and $\hat{s}_h$. Then, he/she computes the elements of $E_p^{(m)}$ given by

   $$K_{h+\delta(j,1,h,l)} = \hat{f}_h(M)^{\hat{r}_h} K_{\delta(j,1,h,l)} \hat{f}_h(M)^{\hat{s}_h}, \quad \text{for } l = 2, 3, \ldots, h-1,$$
   $$K_{h+\sigma(j,1,h)} = \hat{f}_h(M)^{\hat{r}_h} K_{\sigma(j,1,h)} \hat{f}_h(M)^{\hat{s}_h}.$$

   User U_h sends to the new user U_{h+1} the $(h+1)$-vector

   $$\Big(K_{h-1+\delta(j,1,h,1)}, K_{h+\delta(j,1,h,1)}, K_{h+\delta(j,1,h,2)}, \ldots, K_{h+\delta(j,1,h,h-1)}, K_{\sigma(j,1,h+1)}\Big).$$

2. User U_{h+1} acts as previously user U_h did, and computes the elements of $E_p^{(m)}$ given by

   $$L_1^{(h+1)} = f_{h+1}(M)^{r_{h+1}} K_{h-1+\delta(j,1,h,1)} f_{h+1}(M)^{s_{h+1}},$$

$$\begin{aligned} L_l^{(h+1)} &= f_{h+1}(M)^{r_{h+1}} K_{h+\delta(j,1,h,l-1)} f_{h+1}(M)^{s_{h+1}}, \\ &= f_{h+1}(M)^{r_{h+1}} K_{\delta(j,1,h+1,l)} f_{h+1}(M)^{s_{h+1}}, \quad \text{for } l = 2,3,\ldots,h, \\ L_{h+1}^{(h+1)} &= f_{h+1}(M)^{r_{h+1}} K_{\sigma(j,1,h+1)} f_{h+1}(M)^{s_{h+1}}. \end{aligned}$$

3. User U_{h+1} sends to every user the h-vector of elements of $E_p^{(m)}$ given by

$$\left(L_1^{(h+1)}, L_2^{(h+1)}, \ldots, L_h^{(h+1)}\right). \tag{13}$$

4. Finally, when every user U_i, for $i = 1,2,\ldots,h$, receives the h-vector given by expression (13), he/she takes the $(h+1-i)$th entry, $L_{h+1-i}^{(h+1)}$, and computes the new element $\hat{S}_i$ as

$$\hat{S}_i = f_i(M)^{r_i} L_{h+1-i}^{(h+1)} f_i(M)^{s_i}, \quad \text{for } i = 1,2,\ldots,h.$$

User U_{h+1} denote by $\hat{S}_h$ the element $L_{h+1}^{(h+1)}$, i.e., $\hat{S}_{h+1} = L_{h+1}^{(h+1)}$.

Now, Theorem 1 ensures that the shared secret by all users is

$$\hat{S}_1 = \hat{S}_2 = \cdots = \hat{S}_h = \hat{S}_{h+1}.$$

Assume again that the system consists of h users: $U_1, U_2, \ldots, U_h$. if user U_i, for some $i = 1,2,\ldots,h-1$, decides to leave the system, a rekeying is also needed in order to preserve forward secrecy.

Recall that user U_h has the h-vector of elements in $E_p^{(m)}$ given by expression (6). As in the join operation, user U_h generates two new positive integers $\hat{r}_h$ and $\hat{s}_h$, as well as a new polynomial $\hat{f}_h(\mathrm{x}) \in Z\left(E_p^{(m)}\right)$. Then, the elements of $E_p^{(m)}$ are computed by

$$\begin{aligned} L_1^{(h)} &= \hat{f}_h(M)^{\hat{r}_h} K_{h-2+\delta(j,1,h-1,1)} \hat{f}_h(M)^{\hat{s}_h}, \\ L_l^{(h)} &= \hat{f}_h(M)^{\hat{r}_h} K_{h-1+\delta(j,1,h-1,l-1)} \hat{f}_h(M)^{\hat{s}_h}, \\ &= \hat{f}_h(M)^{\hat{r}_h} K_{\delta(j,1,h,l)} \hat{f}_h(M)^{\hat{s}_h}, \quad \text{for } l = 2,3,\ldots,h-1, \\ L_h^{(h)} &= \hat{f}_h(M)^{\hat{r}_h} K_{\sigma(j,1,h)} \hat{f}_h(M)^{\hat{s}_h}. \end{aligned}$$

That is, expressions (7)–(9) are used to compute $L_l^{(h)}$, but replacing $f_l(M)$, r_l and s_l, by $\hat{f}_l(M)$, $\hat{r}_l$ and $\hat{s}_l$, respectively, for $l = 1,2,\ldots,h$. Note that it is not necessary to compute $L_{h-i}^{(h)}$, because user U_i leaves the group.

Finally, user U_h sends to all users, except to user U_i, the $(h-1)$-vector

$$\left(L_1^{(h)}, L_2^{(h)}, \ldots, L_{h-1}^{(h)}\right),$$

and each user computes the new shared secret key as in step 6 of Protocol 1.

In case user U_h decides to leave the system, then user U_{h-1} changes his/her private key and acts as U_h.

5 Example

In this example we show how to share a secret in a system with $h = 4$ uses using Protocol 1.

Assume that $p = 2$ and $m = 5$ and consider the public elements $M \in E_2^{(5)}$ and $N \in E_2^{(5)} \setminus Z\left(E_2^{(5)}\right)$ given by

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 3 & 2 & 7 \\ 8 & 0 & 2 & 14 & 0 \\ 0 & 8 & 4 & 2 & 23 \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 2 & 0 & 0 & 2 & 3 \\ 4 & 0 & 5 & 1 & 2 \\ 0 & 0 & 0 & 7 & 15 \\ 0 & 8 & 0 & 0 & 5 \end{bmatrix}.$$

For some polynomials $f_1(\mathrm{x}), f_2(\mathrm{x}), f_3(\mathrm{x}), f_4(\mathrm{x}) \in Z\left(E_2^{(5)}\right)[\mathrm{x}]$ we have obtained the following elements:

$$f_1(M) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 0 & 4 & 1 & 6 & 5 \\ 0 & 8 & 12 & 7 & 14 \\ 16 & 16 & 16 & 20 & 21 \end{bmatrix}, \quad f_2(M) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 6 & 7 \\ 0 & 0 & 0 & 13 & 6 \\ 16 & 16 & 24 & 8 & 13 \end{bmatrix},$$

$$f_3(M) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 7 & 6 & 4 \\ 8 & 4 & 6 & 4 & 6 \\ 16 & 16 & 20 & 30 & 27 \end{bmatrix}, \quad f_4(M) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 3 & 2 \\ 0 & 2 & 5 & 4 & 6 \\ 8 & 8 & 2 & 4 & 6 \\ 16 & 8 & 28 & 26 & 29 \end{bmatrix}.$$

Assume that $K_0 = N$ and consider

$$(r_1, r_2, r_3, r_4) = (10, 5, 6, 11) \quad \text{and} \quad (s_1, s_2, s_3, s_4) = (5, 14, 16, 12).$$

Recall that the $(r_i, s_i, f_i(M))$ is the private key of user U_i for $i = 1, 2, 3, 4$.

User U_1 computes the element

$$K_1 = f_1(M)^{r_1} K_0 f_1(M)^{s_1} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 & 1 \\ 4 & 4 & 1 & 1 & 7 \\ 0 & 8 & 4 & 13 & 9 \\ 16 & 8 & 0 & 4 & 9 \end{bmatrix}.$$

User U_1 sends K_1 to user U_2.

User U_2 computes the elements

$$K_2 = f_2(M)^{r_2} K_0 f_2(M)^{s_2} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 & 1 \\ 4 & 0 & 5 & 7 & 1 \\ 0 & 0 & 0 & 3 & 5 \\ 16 & 24 & 8 & 0 & 9 \end{bmatrix},$$

$$K_3 = f_2(M)^{r_2} K_1 f_2(M)^{s_2} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 2 & 0 & 0 & 2 & 3 \\ 4 & 4 & 1 & 7 & 6 \\ 0 & 8 & 4 & 9 & 15 \\ 0 & 24 & 8 & 4 & 5 \end{bmatrix}.$$

User U_2 sends to user U_3 the 3-vector (K_1, K_2, K_3).

User U_3 computes

$$K_4 = f_3(M)^{r_3} K_1 f_3(M)^{s_3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 2 & 3 & 0 & 1 \\ 8 & 4 & 6 & 12 & 0 \\ 16 & 0 & 12 & 6 & 27 \end{bmatrix},$$

$$K_5 = f_3(M)^{r_3} K_2 f_3(M)^{s_3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 2 & 3 & 4 & 7 \\ 8 & 4 & 6 & 4 & 12 \\ 16 & 0 & 28 & 6 & 11 \end{bmatrix},$$

$$K_6 = f_3(M)^{r_3} K_3 f_3(M)^{s_3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 3 & 6 & 0 \\ 8 & 4 & 6 & 8 & 6 \\ 0 & 16 & 4 & 14 & 7 \end{bmatrix}.$$

User U_3 sends to user U_4 the 4-vector (K_3, K_4, K_5, K_6).

User U_4 computes the elements

$$L_1^{(4)} = f_4(M)^{r_4} K_3 f_4(M)^{s_4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 3 & 2 & 6 \\ 8 & 4 & 6 & 0 & 2 \\ 0 & 16 & 20 & 14 & 7 \end{bmatrix},$$

$$L_2^{(4)} = f_4(M)^{r_4} K_4 f_4(M)^{s_4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 2 & 3 & 4 & 7 \\ 8 & 4 & 6 & 4 & 12 \\ 16 & 0 & 28 & 6 & 27 \end{bmatrix},$$

$$L_3^{(4)} = f_4(M)^{r_4} K_5 f_4(M)^{s_4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 2 & 3 & 0 & 5 \\ 8 & 4 & 6 & 12 & 8 \\ 16 & 0 & 12 & 6 & 11 \end{bmatrix},$$

$$L_4^{(4)} = f_4(M)^{r_4} K_6 f_4(M)^{s_4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 3 & 2 & 6 \\ 8 & 4 & 6 & 0 & 2 \\ 0 & 16 & 20 & 14 & 23 \end{bmatrix}.$$

Let $S_4 = L_4^{(4)}$.

Now, user U_4 sends to users U_1, U_2 and U_3 the 3-vector $\left(L_1^{(4)}, L_2^{(4)}, L_3^{(4)}\right)$.

User U_1 uses $L_3^{(4)}$ to compute $S_1 = f_1(M)^{r_1} L_3^{(4)} f_1(M)^{s_1}$.

User U_2 uses $L_2^{(4)}$ to compute $S_2 = f_2(M)^{r_2} L_2^{(4)} f_2(M)^{s_2}$.

Finally, user U_3 uses $L_1^{(4)}$ to compute $S_3 = f_3(M)^{r_3} L_1^{(4)} f_3(M)^{s_3}$.

Now, as we established in Theorem 1, $S_1 = S_2 = S_3 = S_4$.

Assume now that a new user, U_5, wants to join the above system.

User U_4 generates a new private key $\left(\hat{r}_4, \hat{s}_4, \hat{f}_4(M)\right)$ with $\hat{r}_4 = 7$, $\hat{s}_4 = 4$ and

$$\hat{f}_4(M) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 3 & 2 & 3 & 0 \\ 0 & 0 & 7 & 4 & 5 \\ 0 & 4 & 12 & 5 & 14 \\ 16 & 8 & 0 & 4 & 23 \end{bmatrix}.$$

User U_4 computes

$$K_7 = \hat{f}_4(M)^{\hat{r}_4} K_3 \hat{f}_4(M)^{\hat{s}_4} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 2 & 3 & 2 \\ 4 & 4 & 7 & 5 & 5 \\ 8 & 8 & 8 & 5 & 13 \\ 0 & 24 & 8 & 8 & 23 \end{bmatrix},$$

$$K_8 = \hat{f}_4(M)^{\hat{r}_4} K_4 \hat{f}_4(M)^{\hat{s}_4} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 6 & 1 & 2 & 4 \\ 8 & 12 & 2 & 8 & 10 \\ 0 & 16 & 12 & 10 & 5 \end{bmatrix},$$

$$K_9 = \hat{f}_4(M)^{\hat{r}_4} K_5 \hat{f}_4(M)^{\hat{s}_4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 6 & 1 & 6 & 6 \\ 8 & 12 & 2 & 0 & 14 \\ 0 & 16 & 28 & 10 & 5 \end{bmatrix},$$

$$K_{10} = \hat{f}_4(M)^{\hat{r}_4} K_6 \hat{f}_4(M)^{\hat{s}_4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 6 & 1 & 4 & 1 \\ 8 & 12 & 2 & 12 & 12 \\ 16 & 0 & 20 & 18 & 1 \end{bmatrix}.$$

User U_4 sends to user U_5 the 5-vector $(K_6, K_7, K_8, K_9, K_{10})$.

User U_5 generates a private key $(r_5, s_5, f_5(M))$ with $r_5 = 4$, $s_5 = 10$ and

$$f_5(M) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 6 & 5 & 6 & 7 \\ 8 & 8 & 6 & 6 & 4 \\ 0 & 8 & 28 & 6 & 17 \end{bmatrix}.$$

User U_5 computes the elements

$$L_1^{(5)} = f_5(M)^{r_5} K_6 f_5(M)^{s_5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 3 & 2 & 6 \\ 8 & 4 & 6 & 0 & 2 \\ 0 & 16 & 20 & 14 & 23 \end{bmatrix},$$

$$L_2^{(5)} = f_5(M)^{r_5} K_7 f_5(M)^{s_5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 6 & 1 & 0 & 3 \\ 8 & 12 & 2 & 4 & 0 \\ 16 & 0 & 4 & 18 & 17 \end{bmatrix},$$

$$L_3^{(5)} = f_5(M)^{r_5} K_8 f_5(M)^{s_5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 6 & 1 & 6 & 6 \\ 8 & 12 & 2 & 0 & 14 \\ 0 & 16 & 28 & 10 & 21 \end{bmatrix},$$

$$L_4^{(5)} = f_5(M)^{r_5} K_9 f_5(M)^{s_5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 6 & 1 & 2 & 0 \\ 8 & 12 & 2 & 8 & 2 \\ 0 & 16 & 12 & 10 & 21 \end{bmatrix},$$

$$L_5^{(5)} = f_5(M)^{r_5} K_{10} f_5(M)^{s_5} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 6 & 1 & 0 & 3 \\ 8 & 12 & 2 & 4 & 0 \\ 16 & 0 & 4 & 18 & 1 \end{bmatrix}.$$

Let $S_5 = L_5^{(5)}$.

User U_5 sends to users U_1, U_2, U_3 and U_4 the 4-vector $\left(L_1^{(5)}, L_2^{(5)}, L_3^{(5)}, L_4^{(5)}\right)$.

User U_1 uses $L_4^{(5)}$ to compute $S_1 = f_1(M)^{r_1} L_4^{(5)} f_1(M)^{s_1}$.

User U_2 uses $L_3^{(5)}$ to compute $S_2 = f_2(M)^{r_2} L_3^{(5)} f_2(M)^{s_2}$.

User U_3 uses $L_2^{(5)}$ to compute $S_3 = f_3(M)^{r_3} L_2^{(5)} f_3(M)^{s_3}$.

Finally, user U_4 uses $L_1^{(5)}$ to compute $S_4 = f_4(M)^{r_4} L_1^{(5)} f_4(M)^{s_4}$.

Now, as we established in Theorem 1, $S_1 = S_2 = S_3 = S_4 = S_5$.

6 Conclusions

We introduce a key agreement protocol for secure communications based over a noncommutative ring with a large number of noninvertible elements. The protocol shows to be efficient for large audiences, making it applicable nowadays for widely extended secure multicast communications and allows users to join or leave the communication group preserving forward and backward secrecy in an efficient way.

Acknowledgements

The work of the first author was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Economía y Competitividad of the Gobierno de España. The work of the second author was partially supported by the grant FQM 0211 of the Junta de Andalucía.

References

[1] Diffie, W.D. & Hellman, M.E., New directions in cryptography. *IEEE Transactions on Information Theory*, **22(6)**, pp. 644–654, 1976.

[2] Odoni, R.W.K., Varadharajan, V. & Sanders, P.W., Public key distribution in matrix rings. *Electronics Letters*, **20**, pp. 386–387, 1984.

[3] Menezes, A.J. & Wu, Y.H., The discrete logarithm problem in $GL(n, q)$. *Ars Combinatoria*, **47**, pp. 23–32, 1997.

[4] Cao, Z., Dong, X. & Wang, L., New public key cryptosystems using polynomials over non-commutative rings. Cryptology ePrint Archive, Report 2007/007, 2007. `http://eprint.iacr.org/`.

[5] Deering, S.E. & Cheriton, D.R., Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems*, **8(2)**, pp. 85–110, 1990.

[6] Deering, S.E., Estrin, D.L., Farinacci, D., Jacobson, V., Liu, C.G. & Wei, L., The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, **4(2)**, pp. 153–162, 1996.

[7] Climent, J.J., Navarro, P.R. & Tortosa, L., On the arithmetic of the endomorphisms ring $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$. *Applicable Algebra in Engineering, Communication and Computing*, **22(2)**, pp. 91–108, 2011.

[8] Climent, J.J., Navarro, P.R. & Tortosa, L., Key exchange protocols over noncommutative rings. The case $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$. *Proceedings of the 11th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE 2011)*, ed. J. Vigo Aguiar, pp. 357–364, 2011.

[9] Climent, J.J., Navarro, P.R. & Tortosa, L., Key exchange protocols over noncommutative rings. The case of $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$. *International Journal of Computer Mathematics*, **89(13–14)**, pp. 1753–1763, 2012.

[10] Kamal, A.A. & Youssef, A.M., Cryptanalysis of a key exchange protocol based on the endomorphisms ring $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_p^2)$. *Applicable Algebra in Engineering, Communication and Computing*, **23(3–4)**, pp. 143–149, 2012.

[11] Climent, J.J., Navarro, P.R. & Tortosa, L., An extension of the noncommutative Bergman's ring with a large number of noninvertible elements. *Submitted*, 2013.

[12] Steiner, M., Tsudik, G. & Waidner, M., Diffie-Hellman key distribution extended to group communication. *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, ACM: New York, NY, pp. 31–37, 1996.

[13] Steiner, M., Tsudik, G. & Waidner, M., Key agreement in dynamic peer groups. *IEEE Transactions of Parallel and Distributed Systems*, **11(8)**, pp. 769–780, 2000.