Project 2: Lattices in quaternion algebras

Computer Algebra for Cryptography (B-KUL-H0E74A)
Clémence Duplat r0977801

May 24, 2024

1 Task 1

To implement the functions of task 1, we simply applied some calculations with the constraint that the quaternions are in H_p, where p is a prime number satisfying $p \equiv 3 mod 4$.

For the function *HpMult* and *HpAdd*, those calculations are really straightforward. To ensure they are correctly implemented in magma, we can verify the basic properties like **asscoiativity**:

```
 HpMult(p, HpMult(p, u, v), w) eq HpMult(p, u, HpMult(p, v, w));
```

commutativity

```
HpAdd(u1, u2) eq HpAdd(u2, u1);
```

distributivity

```
 HpMult(p, u, HpAdd(v, w)) eq HpAdd(HpMult(p, u, v), HpMult(p, u, w));
```

and for the *HpAdd* we can also verify that adding the zero quaternion to a quaternion leaves this quaternion unchanged.

For the last function *HpInv*, we used the fact that every non-zero element $u \in H_p$ (where H_p is a so-called division ring) admits an inverse u^{-1} satisfying $u^{-1}u = uu^{-1} = 1$. We can thus show that $u^{-1} = \frac{\bar{u}}{n(u)}$ because $uu^{-1} = \frac{u\bar{u}}{n(u)} = \frac{n(u)}{n(u)} = 1$. The correctness of this function can be tested by ensuring that the norm of the quaternion times its inverse is truly one

```
HpMult(p, u, HpInv(p, u)) eq [1, 0, 0, 0];
```

2 Task 2

2.1 Task 2a

The output of the function *LatticeSum* is a matrix B whose rows form a basis of $L_1 + L_2$. We know that the rows of the matrices B_1 and B_2 form bases of the respective lattices L_1 and L_2. By stacking vertically together B_1 on top of B_2, we combine all basis vectors from both lattices into a single matrix. The dimensions of those matrices does not match the required output. Therefore, we will compute the Hermite normal form of this matrix, which provides a canonical, simplified form of our matrix. It rearranges the lattice basis into a form where each vector adds a new dimension that is not already spanned by the previous vectors. However, the function *HermitForm* in Magma works on matrices with entries in Z. Therefore, we will have to convert our rational matrix into an integer matrix. This is done by finding the LCM (least common multiple) of the denominators of all entries in B. Each rational entry can be defined as $r_{ij} = \frac{n_{ij}}{d_{ij}}$. By multiplying each entry of B by the LCM of all d_{ij}, each element can be defined as $\frac{n_{ij} * LCM}{d_{ij}}$. Each rational element is thus converted into an integer one, because LCM is a multiple over each denominator d_{ij}. The LCM is used in order to avoid additional and difficult multiplications.
Once we have compute the Hermite normal form, we have to convert our matrix back into rationals. This is done by dividing the entries of the matrix by the least common multiple already computed before.

2.2 Task 2b

- First of all we can show that given $B \in \mathbb{Q}^{n\times n}$ a basis matrix for L, then there is a very explicit description of a basis for L^*: $(B^T)^{-1}$.
  Indeed, let $B = [b_1, b_2, \dots, b_n]$ where $b_i \in \mathbb{Q}^n$. Based on the notation of the dual of the full rank lattice L given in the assignment, any vector $w \in L^*$ has to satisfy $\langle w, b_i\rangle \in \mathbb{Z}$ for all i. Let $w = (B^T)^{-1}v$ for some $v \in \mathbb{Z}^n$. Then we have that for b_i:
  $$\langle w, b_i\rangle = \langle (B^T)^{-1}v, b_i\rangle = v^T(B^T)^{-1}b_i$$
  We know $(B^T)^{-1}b_i = e_i$ and we can thus get:
  $$v^T(B^T)^{-1}b_i = v^Te_i = v_i$$
  We thus have that $v_i \in \mathbb{Z}$. This shows that $w \in L^*$ and thus prove that $(B^T)^{-1}$ was indeed a good choice for the basis of L^*.

- We can also prove the following identity : $L_1^* + L_2^* = (L_1 \cap L_2)^*$, that holds true for any pair of full-rank lattices $L_1, L_2 \subseteq \mathbb{R}^n$.
  First of all lets prove that $(L_1 \cap L_2)^* \subseteq L_1^* + L_2^*$. If we pose that $w \in (L_1 \cap L_2)^*$ we can say that $\langle u, w\rangle \in \mathbb{Z}$ for all $u \in L_1 \cap L_2$ (by definition of dual lattice). The intersection of a lattice contains elements that are in both lattices: $L_1 \cap L_2 \subseteq L_1$ and $L_1 \cap L_2 \subseteq L_2$. Therefore, we have that $\langle u, w\rangle \in \mathbb{Z}$ for all $u \in L_1$ and all $u \in L_2$. Logically, we find that $w \in L_1^* \cap L_2^*$ and thus $(L_1 \cap L_2)^* \subseteq L_1^* + L_2^*$.
  Then lets prove that $L_1^* + L_2^* \subseteq (L_1 \cap L_2)^*$. This time let $w \in L_1^* + L_2^*$ that we can denote as $w = w_1 + w_2$ where $w_1 \in L_1^*$ and $w_2 \in L_2^*$. For any $u \in L_1 \cap L_2$ we have that $\langle u, w\rangle = \langle u, w_1\rangle + \langle u, w_2\rangle$. Because $w_1 \in L_1^*$, we have that $\langle u, w_1\rangle \in \mathbb{Z}$, and because $w_2 \in L_2^*$, we have that $\langle u, w_2\rangle \in \mathbb{Z}$. Therefore, $\langle u, w\rangle \in \mathbb{Z}$ for all $u \in L_1 \cap L_2$, proving that $w \in (L_1 \cap L_2)^*$.
  Combing both we proved that $(L_1 \cap L_2)^* = L_1^* + L_2^*$.

The two proofs are useful to compute the intersection of two lattices. Indeed, it can be computed in three steps : compute the basis of the dual (first proof), compute the sum of the basis of the duals and finally take the transpose of the inverse to obtain the intersection (second proof). The function *LatticeIntersect* is implemented in Magma and well commented.

3 Task 3

In this task, a function verifying if a lattices is a maximal order is done. This is done using *theorem 1* defined in the assignment. Our implementation is used, with $p^{271} - 1$, to determine which of the following lattices are maximal orders:

- i) $\mathbb{Z} + \mathbb{Z}i + \mathbb{Z}j + \mathbb{Z}k$: false, there exist an order (Lattice 4) in H_p such that $Lattice1 \subset Lattice4$. This is in contradiction with the *Definition 2* of an maximal order.
- ii) $\frac{1}{2}\mathbb{Z} + \mathbb{Z}i + \mathbb{Z}j + \mathbb{Z}k$: false, this lattice is not even an order, meaning it will also not be a maximal order. Indeed, we see that *Definition 1* is not fulfilled : the lattice is not closed under multiplication.
- iii) $2\mathbb{Z} + \mathbb{Z}i + \mathbb{Z}j + \mathbb{Z}k$: false, this lattice is not even an order, meaning it will also not be a maximal order. Indeed, we see that *Definition 1* is not fulfilled : $1 \notin Lattice3$
- iv) $\mathbb{Z} + \mathbb{Z}i + \mathbb{Z}\frac{i+j}{2} + \mathbb{Z}\frac{1+k}{2}$: true, it is a maximal order.
- v) $\mathbb{Z} + \mathbb{Z}i$: this lattice is not even an order, meaning it will also not be a maximal order. Indeed, we see that *Definition 1* is not fulfilled : it is not a full-rank lattice.

4 Task 4

4.1 Task 4a

In order to explain why $O_L(I) = \{\alpha \in H_p | \alpha I \subseteq I\}$ is an order, we need to prove that *Definition 2* is satisfied.

- $1 \in O_L(I)$: satisfied because multiplying 1 by any element of I results in that element remaining in I ($1 * x = x$ for any $x \in I$) and thus $1I = I \subseteq I$).

- If α, $\beta \in O_L(I)$then also $\alpha\beta \in O_L(I)$: we will have that $\alpha I \subseteq I$ and $\beta I \subseteq I$. We have to prove that $\alpha\beta I \subseteq I$. If we consider an element $x \in I$, we know that $\beta x \subseteq I$ (because $\beta \in O_L(I)$). It follows that $\alpha(\beta x) \subseteq I$ and because $\alpha(\beta x) = (\alpha\beta)x$, we have that $(\alpha\beta)x \in I$. This argument holds true for any $x \in I$ and therefore we find that $\alpha\beta I \subseteq I$ and therefore that $\alpha\beta \in O_L(I)$.

- Additionly, we can prove that $O_L(I)$ is a full-rank lattice, which is straightforward because it matches the dimensionality of H_p (spanned by the quaternion).

4.2 Task 4b

This taks was implemented in magma and is well commented.

4.3 Task 4c

This task was implemented in magma and is well commented. The function was tested with $p = 2^{127} - 1$ and the Lattice 4 which is the unique maximal order (see task 3). The left maximal order can be computed using a few random ideals, which are just different linear combinations of the Lattice 4. The function is correct because the left maximal orders are still maximal.

5 Task 5

From *Theorem 3*, we can conclude that norm and the trace of any element of O are integers.
Any order $O \subseteq H_p$ is closed under conjugation i.e. for any $u = a + bi + cj + dk \in O$ and $\overline{u} = a - bi - cj - dk \in O$.
We have the trace $t(u) = u + \overline{u} = 2a$, where a is a rational number. We know that u and $\overline{u}$ are both in O, and that O is an order (a lattice over $\mathbb{Z}$). This means O is closed under addition ($u + \overline{u} \in O$) and contains integer linear combinations of its elements. The trace must thus be an integer.
We have the norm $n(u) = u * \overline{u} = a^2 + b^2 + pc^2 + pd^2$, where each term a,b,c,d are rationals. We know that u and $\overline{u}$ are both in O, and that O is an order (a lattice over $\mathbb{Z}$). This means O is closed under multiplication ($u\overline{u} \in O$) and contains integer linear combinations of its elements. The order O ensures thus that that the norm $n(u)$ is an integer.

6 Task 6

6.1 Task 6a

This function project the elements of an order 0 in the quaternion algebra $\mathbb{H}_p$ onto the imaginary components. The notation of O_proj in the assignment will indicate us that the real component will be removed. This code is implemented in magma and well commented.

6.2 Task 6b

This function uses the projected lattice just computed in order to compute a signature based of the norm of the shortest vector in the lattice. To do so, we will use the magma function *ShortestVector* that compute this shortest euclidean norm for the rationals. We thus have to convert the rationals into integers. To handle the fact that the norm n of $\mathbb{H}_|$ does not coincide with the Euclidean norm on $\mathbb{R}^4$, we can scale the coordinates appropriately (scale c and d by $\sqrt{p}$). This will ensure that the norms are comparable. When the rational numbers are very small, this can introduce some precision issues leading to rounding errors. Therefore, we will scale the matrix entries by a very large factor. As said before, to ensure that the imaginary components are scaled correctly relative to each other, we will adjust the scalling factor by multiplying it with $\sqrt{p}$.
We do not have to forget scale back the shortest norm.
For my maximal order Lattice 4 and $p = 2^{217} - 1$, it is working and returns me a signature of 4.

6.3 Task 6c

We can use Minkowski's theorem to find an upper bound on this norm. The theorem states that for any full rank Lattice L and any measurable convex set C symmetric about the origin we have:

$$\text{if } vol(C) > 2^d vol(L) \text{ then } (C \cap L) \setminus 0 \neq \emptyset$$

First, can compute the volume of the lattice $L = L(b_1, ...b_d)$ by using the Gram matrix $G = (\langle u_i, u_j \rangle)_{i,j}$: $\mathrm{vol}(L) = \sqrt{\det(G(b_1, ...b_d)}$

We can use *theorem 1* that states that $\langle u_i, u_j \rangle = t(u_i u_j)$ and thus find that the $vol(L) = p$.

The bounds of the successive minima are given by:

$$\lambda_1(L) \leq 2 \left(\frac{\mathrm{vol}(L)}{v_d} \right)^{1/d} \leq \sqrt{d} \cdot \mathrm{vol}(L)^{1/d}$$

and the volume v_d of the unit ball in d-dimensions is given by:

$$v_d = \frac{\pi^{d/2}}{\Gamma(1 + d/2)}$$

and we thus get for our case where $d = 4$:

$$v_4 = \frac{\pi^2}{2}$$

and thus the following upper bound on the norm of the shortest vector:

$$\lambda_1(L) \leq 2 \left(\frac{p}{\pi^2/2} \right)^{1/4}$$

7 Task 7

The *KeyGen* function in magma generate a secret and a public key. To do so, we will take random values b,c and d in our specific ring in order to compute a^2. We can now use the function *IsSquare* that will have two outputs: the value a and if it is a solution in the ring. If the solution is in the ring, we can now construct our quaternion, corresponding to the secret key. The public key is a matrix whose rows generates the left order $O_L(I)$, where $I = l^e O + uO$ is the ideal that has to have a norm l^e. Once the condition on the ideal is satsified, we can use the function *LeftMaximalOrder* to compute the public key. This function is well implemented and well commented in Magma.

8 Task 8

We have to proof that $O_L([I_B]_* I_A) = O_L(I_A \cap I_B)$ along three steps.

- **If $I \subseteq I'$ are right O-ideals such that $n(I) = n(I')$, then $I = I'$**

  Suppose we have I and I' that are right O-ideals. The norm $n(I)$ of an ideal I is defined by *Definition 3* as $n(I) = \gcd(\{n(u) \mid u \in I\})$. So if $I \subseteq I'$ (every element of I is also an element of I') and $n(I) = n(I')$ (gcd of the norms is the same), then I' (that can be expressed as terms of I) does not contain any more elements than in I. We thus have that, $I = I'$.

- **Use this to show that** $([I_B]_* I_A) \cdot I_B = I_A \cap I_B$

  By the product $[I_B]_* I_A$ of two lattices, we mean the lattice generated by all possible products:$[I_B]_* I_A = \{\sum_i b_i \cdot a_i \mid b_i \in I_B, a_i \in I_A\}$
  Because I_B and I_A are right O-ideals, they are closed under the right multiplication by any element of the order O. We thus have that $b * a \in I_B$ and $a * b \in I_A$. Therefore any product of this form is also a subset of $I_A \cap I_B$.
  Any element in $I_A \cap I_B$ is composed of elements that are both in I_A and I_B, and thus it can be expressed as an element of $([I_B]_* I_A) \cdot I_B$.
  We thus finally have that $([I_B]_* I_A) \cdot I_B = I_A \cap I_B$

- **Explain why $O_L([I_B]_* I_A) \subseteq O_L(([I_B] * I_A) \cdot I_B)$ and use maximality to conclude that equality holds**

  We can take back the definition of a left order. For an $\alpha \in O_L([I_B]_* I_A)$ we have that $\alpha([I_B]_* I_A) \subseteq [I_B]_* I_A$.
  Taking the formula found in step 2 we can find that $([I_B] * I_A) \subseteq ([I_B]_* I_A) \cdot I_B$. Knowing this and the fact that if $\alpha \in O_L([I_B]_* I_A)$ we have that $\alpha([I_B]_* I_A) \subseteq ([I_B]_* I_A) \cdot I_B$. We thus have that $\alpha \in O_L(([I_B]_* I_A) \cdot I_B)$, which implies that $O_L([I_B]_* I_A) \subseteq O_L(([I_B] * I_A) \cdot I_B)$.

  Both part of the last equation are maximal orders and by using the *Definition 2* of a maximal order (*Definition 2*) it automatically follows that:

$$O_L([I_B]_* I_A) = O_L(([I_B]_* I_A) \cdot I_B)$$

9 Task 9

For this task we create a function *ExtractKey* that output the signature (shared key) of $O_L(I \cap J)$. It will scale the input keys based on their respective norms, combine them compute the left maximal order. We can then use *OrderSignature* in order to compute the signature.
When comparing the signatures of Alice and Bob for $p = 2^{127} - 1$, I observe that the shared key match, meaning my code works correctly.