听写：ScaleresTalk成长会听力狂练小组成员。成员名单：April，@蓝梦水冰凌，@小小草—华吕燕，Erin，海绵，Quesera，宁宁，花臂，流水，@译员小将Summers，Ogloo，Confetti，@Arthur-韩亚非，Nini，@Doris\_素丹，Freya，Lily

翻译、压制：@刘巍然-学酥

校对：@Scalers，@MorningW

时间轴：@MorningW

我们回到讲座中来

So…

我们首先来总结一下第一个小时讲解的内容

So, just to summarize what we discussed for an hour.

我们讲了格的基 讲了两个格基的等价条件

Basis of Lattices we saw, when two bases are equivalent.

我们讲的第二个知识是格的基本区域 后面我们应该还会再用到这一概念

And next notion, the second notion we saw is that of fundamental region which we should see again.

基本区域是一个区域

The fundamental region is a region

如果把这个区域放在每一个格点上 所有这些区域连接在一起 可以完整覆盖整个平面

that once you translate it by any lattice point, you get this joint set of sets that with unique unit in the whole plain, cover the whole plane.

正如幻灯片上的这个图 以及图上的这些区域

So this is like this picture here, and like pictures over there.

基本区域还有一个等价定义

And it has an equivalent definition that

在一个基本区域中只存在一个格点 不可能同时存在两个格点

you should never have two points of any translate of lattice inside the body that's equivalent.

所以这个基本区域只包含各个陪集的点集

So such a body, such a fundamental region contains exactly one representative of each coset.

我给大家讲解这个概念时借助了存储函数值的思想

So, the reason I motivated this by saying you want to restore the function.

但在Vadim的讲座中会提到 格中使用这种表示方法的目的不是为了存储函数值

But you see later in Vadim’s talk, actually the motivation is usually slightly different.

我们在空间中取点的时候 不太关心点的实际位置 而是关心点在格中的相对位置

We want to take points, for instance, see the point, the lattice point. We do not really care about the point set. We only care about the position relative to the lattice.

如果我们想存储点对应的函数值 我们只需要把点模平行多面体

And all we have to store the point modulo of parallelepiped,

然后存储这个点模基本区域后的函数值

storing the location in some fundamental region,

一般把这个基本区域称为基本平行多面体 因为这个名字更形象一些

usually for being fundamental parallelepiped, because this is most convenient to work with.

我们开始讲解下一个知识点

So, to bring it to the next notion,

我们再来看一下这个图 我看到有些朋友们开始记笔记了 非常好

let's look at the picture again and someone already doing to break the notes. So it is very nice.

我们来看看左边这个正方形

Let's look at the square here.

现在我们考虑这个正方形的面积 正方形的面积是1 因为1乘以1 很简单

So consider the area of this square, so you look at the square. The area is one, cause one times one.

但右边这个平行四边形的面积是多少？

That is very easy. But what is the area of the parallelogram?

大家知道在代数学里面怎么求这个面积吗？

Anybody remember, somebody remember in algebra?

面积是…

So it’s…

你可以计算一下 这个面积同样也是1

So you compute that. So it is one. It is also one.

如果你认真思考一下 就会发现这不是一个巧合

And it is not a coincident actually if you come to think of it.

还记得基本区域的定义吗？

Remember the fundamental region?

我们定义基本区域的方法是把这个区域放在每一个格点上 用格点平移这个区域

The way I define it is if you repeat this region and put one on each lattice point,you translated by lattice points.

基本区域应该可以覆盖整个空间 幻灯片上两个区域都能覆盖整个空间

you should get nice desolation of the whole plain. It’s similarly there.

如果你稍微想一想 这个定义也意味着这两个区域的面积必须是一样的

And if you think about it for a few seconds, you convince yourself that this implies that these two areas must have the same volume, have the same area.

证明方法是在平面上取一个非常大的区域

And one way to prove it would be take a very big area of the plane,

并且假设在这个区域中 你有上亿个格点

and assuming in that big area, you have billion lattice points.

而且这个区域中还有很多类似这样的小区域

It also expected to have a bit of these little regions inside.

这样你就分别用左右两个区域来覆盖整个大区域 他们覆盖的都是同样的大区域

And so you have this big region that is in one picture and in another picture. They both cover the same area.

它们的面积应该一样 因为上亿个小区域都能覆盖同样的大区域

So they should have the same area, because billion of them can cover the same area of the plane.

取很多小区域之后 面积就和区域的形状没关系了 可以忽略区域的形状

The reason they take so many to avoid the shape, to ignore, to be able to ignore the specific shape of that region.

我们可以取很多很多小区域 用不同的小区域覆盖大区域

You want to take many, many of them, kind of average over a big area of space,

所有这些区域的面积都是一样的

All have the same volume.

有人注意到吗？对于平行多面体来说也是一样的 大家稍微想一下

But you can also… someone noticed here? You can also see directly for parallelepiped. Does we see in a minute?

一般来说 对于任意平行多面体 这个性质都成立

But in general, this is true for any fundamental region, all of the same area,

即使平行多面体是天马的形状 也能直接得到天马的面积 可以直接计算面积值

even though I have this here this strange horse with hawk. I can know what the area is. So just compute the area.

很容易计算平行多面体的体积

You know, a fundamental parallelepiped is something that is easy to compute.

可以直接得到天马形状的面积值 因为它是一个基本区域

And I know what area of this shape, just because it is a fundamental region.

Vadim会提到另一个特性 可以把平行多面体上的点一一映射到另一个平行多面体上

And another feature of these things, you’ll see later in the Vadim’s talk, is that you can also go from one to another, from one region to another.

这形成一个双射 形成一个平行多面体到平行多面体的一对一映射

It is a bijection of… volume preserving bijection from one to another.

模这个平行多面体和模另一个平行多面体本质上一模一样 很容易相互转换

There is no difference if you want to work modulo this is a parallelepiped, modulo the other fundamental parallelepiped, it kind of easily go from one to another.

本质上 这些平行多面体包含的信息是一样的

They all essentially contain the same information for us.

我们后面会精确定义这一点 我很愿意把Vadim要讲内容都讲了 不过不会的

OK, this would make it precise officially later. I’m happy to explain Vadim for everything. I don’t do.

这就是我下面要定义的一个概念了 叫做行列式

So here is the next notion that I want to define. It’s the determinant.

什么是行列式？

So what is the determinant?

本质上说 行列式就是平行多面体的体积 它告诉了我们格点的密度

It’s essentially the volume that I just mentioned. It tells us the density of lattice points.

什么是行列式？行列式是定义为基矩阵的体积 即基向量所构成多面体的体积

So what is determinant? Determinant is defined simply as the absolute value of the determinant of this basis matrix, the matrix with columns are the basis elements.

所以格L(B)的行列式由基B生成的平行多面体的体积所定义 也就是det(B)的绝对值

So again, the determinant of a lattice L(B), it is a lattice defined by basis B that is generated by basis B, it’s simply that the absolute value of det(B).

如果你观察一下这个定义 你可能先会问个问题 为什么要定义这么一个标量呢？

If you actually stare the definition, the first you thing you should ask yourself is why you define this, this value definition?

因为我们实际上在用基的性质来定义一个标量 而我们知道一个格有很多的基

because I’m actually defining something the property of the lattice based on the bases, and we know that lattice has lots of bases.

所以为了让这个定义有意义 我们需要保证对于不同的格基 其标量值都是一样的

So for this to make sense, you really want to make sure that for different bases, you get same quantity.

实际上标量值确实是一样的 我们可以证明 证明思路如下

Indeed you do. We have to prove this and the idea is this.

我已经讲过 两个基生成了同一个格当且仅当一个基可以用另一个基乘以幺模矩阵得到

I told you already that two different bases generate the same Lattice, if and only if there’re related by multiplication by unimodular matrix.

所以我就有了另一个基 用BU表示

So if I have different basis BU,

而BU的行列式等于B和U的行列式相乘

then the determinant of that thing of BU is specifically by the multiplicatively of the determinant.

也就是B的行列式乘以U的行列式 同样要取绝对值

It’s simply the determinant of B times determinant of U. Again, absolute value.

但是根据定义 幺模矩阵的行列式值为正负1

But unimodular matrix by definition has a determinant that is plus or minus 1,

所以可以把这一项拿掉 得到BU的行列式等于B的行列式

so you can just cancel it then you get determinant of B.

这个定义很好 格行列式的定义很准确

So this is well defined. The determinant of lattice is a well-defined notion.

行列式还有很多有趣的意义 它告诉了我们格点的密度值

And also there’re lots of dramatic meanings. It tells us the density of the points in lattices.

如果行列式很大 意味着格点比较稀疏 点与点之间有更多的空隙

So if determinant is big, it means that the lattice is sparser. There’s more space between the points.

具体原因还是要从线性代数角度考虑

So again, the reason this happens because you should recall from linear algebra,

一个格的行列式 等价于列向量构成矩阵所对应的平行多面体体积

whatever that the determinant of a lattice is exactly the volume of the basic parallelepiped generated by the columns of the matrix.

如果有人忘了的话 我在这里给大家复习一下 行列式就是对应的平行多面体的体积

So this is… the formula for the volume parallelepiped is simply the absolute value of the determinant in case you’ve forgot.

而我前面讲到 平行多面体P(B)就是格的一个基本区域

And as I told you before, the parallelepiped P(B) is the fundamental region of lattice.

平行多面体的体积正是B的行列式 所以行列式表示格点的密度

So the volume of the parallelepiped is exactly the determinant of B, hence it’s exactly the determinant of the lattice, hence the determinant is the reciprocal of the density

很容易计算行列式的值 因此很容易计算格点的密度

So, this is one property for lattice that we can easily compute, right? This is something we can compute. The determinant is easy to compute.

如果你有一个格 你很容易指出这个格的格点密度情况

And if you have lattice, you can easily tell what density it has.

我会时刻提醒某个定义计算的难易程度 因为这是格密码学的一个重要内容

I’m keep mentioning computation of things because this is the main point of lattice’s crypto.

这就是行列式的概念

So this is one notion of the determinant.

我要讲的下一个概念是连续极小

So the next notion I want to introduce is that of successive minima.

这就不再是一个很直观的概念了 这是数学家早期研究的一个概念

This is already far less trivial notion. This is something that mathematicians started working early.

Minkowski主要研究这方面的内容 我在这里会给大家介绍Minkowski定理

Minkowski has worked in this. We’ll show Minkowski’s theorem on this.

什么是连续极小？

So what are the successive minima?

我们用λ\_1表示第一个连续极小值 λ\_1是L中最短向量的长度

So λ\_1, this is known as the first successive minimum. λ\_1 is the length of the shortest vector in L.

如果我们有一个格L L的λ\_1表示格的最短向量长度

If you have a lattice L, λ\_1 of L is simply the length of the shortest vector in the lattice.

当然 这种定义方法忽略了连续极小的两个隐含意义

So actually the ways defined here is wrong in two senses and missing in another sense.

第一个忽略的隐含意义是 我问的是最短向量

So, I guess the first thing it’s missing is the first thing wrong is because I ask for the shortest vector.

实际上最短向量永远是0 0永远是格中的一个向量

Actually the shortest vector is always 0. 0 is always the vector in Lattice.

我们当然不能把0考虑进来 所以我们要求是非0向量

Of course I don’t want to take it into account, say, non-zero.

第二个隐含意义是 我们用欧几里得范数来表示距离 也就是l\_2距离

The second thing I should specify is that length is, say, Euclidian length, l\_2 length.

当然了 不一定非要用欧几里得范数 不过一般用欧几里得范数表示

It doesn’t have to be Euclidian. But usually it’s going to be Euclidian.

当用非欧几里得范数时 会得到其他一些有趣的性质

There is also interesting stuff happening for non-Euclidian norms.

不过现在我们只考虑欧几里得范数

But we don’t need it here. So I think we will be happy with l\_2 length.

只用欧几里得范数 最一般的范数表示方法

Just the Euclidian norm of the vector, the usual norm.

我们还忽略了一个隐含意义 使得这个定义并不是很准确 有人能猜到吗？

There’s actually one more thing missing here. One thing it’s not fully precise. Can anyone guess?

完全正确 我不应该用最短向量这个词 我应该用最短向量集合这个词

Exactly, should be. Instead of the shortest vector, it should be a set of the vectors.

因为最短向量肯定不是仅有1个 至少有2个最短向量

because there’s really never one, there’re always at least two.

因为如果你有一个最短向量 这个向量的负向量一定也是最短向量 其长度一样

because if you have the shortest vector, its negation is also the vector of the same length,

而且负向量也一定是在格点中

it’s also the vector in the lattice.

在有些情况下 可能还有很多最短向量

But often you might… sometimes you might have more.

比如Z^n格 在Z^n格中 我们有2n个最短向量 每个轴上都有两个最短向量

You might have a lattice Z^n. In Z^n, you have 2n shortest vectors, two on each axis.

不过人们一般统一称他们为最短向量 所以这么说也没问题

But it’s never there. People often say this is the shortest vector, forget about it.

如果更准确些 我们应该称之为最短向量集合

To be precise, one should say a set of short vector.

这就是λ\_1 这就是第一个连续极小

This is λ\_1. This is the first successive minima.

你估计能猜到 还有其他的连续极小

You can guess there are more.

我们用下述方法定义所有的连续极小

So, more generally the way we define all the successive minima is the following.

λ\_k 其中k的范围是从1到n n表示格的维度

So λ\_k, so k is between 1 and n. And I should say n is always formulated the dimension of the lattice.

n一般来说是格的维度 k的范围是从1到n

So, n is always the dimension of lattice. k is between 1 and n.

连续极小是类似这样的半径值 我们在k维下寻找半径值

It’s kind of this radius. We start seeking k dimension.

我们寻找k格线性无关的向量

We start seeking k linearly independent vectors.

这个图中大家已经能看到最短向量了

So in the picture here, you see the shortest vector.

你可以注意到 图中有两个最短向量

And you notice that there’re twice the shortest vectors here.

但没关系 这两个最短向量都不是λ\_2

But it doesn’t come, it’s not gonna determine λ\_2.

λ\_2得用线性无关的向量表示

For λ\_2 have linearly independent vector.

因此这是第一个向量

Hence, this is the first vector.

这是我们在二维空间中寻找两个线性无关向量时得到的第一个半径

This is the first radius where we start seeking two linearly independent vectors.

这是一个向量 长度就是λ\_1 向量的长度用λ\_1表示

This is the vector. This is λ\_1. The length of this vector is λ\_1.

λ\_2不是这个向量 λ\_2实际在这里

λ\_2 is not this vector. But it’s actually the one here.

这是我们找到两个线性无关向量时得到的格点

The point we start seeking two linearly independent vectors.

我们一般只关心λ\_1也就是最短非0向量

So, and usually what we care about is λ\_1, which is the shortest nonzero vector.

λ\_n是另一边的极限值了 λ\_n是n个线性无关向量的半径

And λ\_n, which can be other extreme, λ\_n is the radius where you’ve n linearly independent vectors.

这时候你就得到了整个线性空间的基

You have essentially bases of the entire linear space,

不是格的基 而是整个线性空间的基 不过这两个是等价的

not the basis of lattice, but the basis of the entire space, sort of the same thing.

接下来

So…

我们会经常用到下一个概念 这是个非常有用的概念

The next notion I need… we use that quite a lot. This is a very useful notion.

实际上这是线性代数中的概念 你估计已经在线性代数课程中学过这个概念了

Actually this is basically linear algebra, this is…Some of you might have seen in basically algebra course.

在格中 这个概念也是一样的 但是这个概念非常有用 与格的联系非常紧密

This has nothing to do with lattice. But it’s very useful and also in relation with lattices.

这是个什么概念呢？这个概念称为Gram-Schmidt正交化 相信很多人都听说过

So what is this process? It’s called Gram-Schmidt Orthogonalization. Sure, most of you hear it.

这里我给大家复习一下 这个正交化过程是如何工作的 这对格非常有用

But just tried to recall exactly how it works. It’s very useful for lattices.

我们有一个有序的向量集合 我们希望让他们正交

So what we have? We have a sequence of vectors we want to make them orthogonal.

我们的方法是对每个向量在前面的向量上做投影

So what we can do is kind of projecting each vector on the orthogonal span of the previous ones.

这是一个有序向量几何v\_1, v\_2, …, v\_n

It's a sequence of vectors v\_1, v\_2, ..., v\_n.

每次操作时 我们把一个向量投影到前面向量的生成空间中

And each time, project each vector on the orthogonal span of the other one.

所以向量的顺序起到了很重要的作用 所以这里是有序向量集合

Since there’s an important order in these vectors. So this is not a set of vectors.

我们来举个例子…

It’s still a little bit.

我们这里有v\_1 我们把v\_2投影到v\_1的正交空间中

So, I have v\_1 there. And then v\_2 is going to be projected on the orthogonal space of v\_1.

所以v\_2会被平移到v\_1的正交空间上 差不多是这个样子

So I think v\_2 is kind of shifted to be orthogonal to v\_1. And it’s going to look like that.

注意到v\_2波浪不再是格点了

Notice that v\_2 tildebar is no longer in the lattice.

所以大家注意 这个处理过程不是一个格上的处理过程 而是一个线性代数处理过程

As I said, this process is not a lattice process. It's in some other process from linear algebra.

而且令人惊讶的是 v\_2波浪不再是一个格点了 我们不能把它看作格点了

It’s also suprising v\_2 tildebar is not a lattice point. So, don't expect it to be.

这个定理非常有用

The theory is very useful.

写出计算v\_2波浪的方法是一个很棒的线性代数习题

If you want to write it in linear algebra, it’s a good exercise to figure out exactly how to compute the v\_2 tildebar.

v\_1波浪就是v\_1

So, v\_1 tildebar is the v\_1.

如果我们把第一个向量投影到前面向量的生成空间上的话

The first vector, if I projected on the orthogonal span of the previous one,

由于前面没有向量的 所以我们直接保留这个向量

so no previous vectors, we just saved in place.

但是v\_2的处理过程就有意思了

But v\_2 tildebar is already more interesting.

v\_2波浪等于v\_2减去v\_2与v\_1波浪的内积 然后再映射到…

It’s v\_2 minus the inner product of v\_2 with v\_1 tildebar. And then I go in direction of...

这等于v\_2减去v\_2在v\_1上的投影值 公式如幻灯片所示

So, v\_2 minus the projection of v\_2 on the span of v\_1, which is already written there.

等于v\_2投影到v\_1波浪的值再乘以v\_1波浪后 除以v\_1波浪范数的平方

It’s v\_2 in the projection of v\_1 tildebar times v\_1 tildebar divided by the norm of v\_1 tildebar squared.

我们很容易看出这个公式的意义 作为一个练习吧

Anyway, it's easy to figure out even if you don't see it clearly. It's an exercise.

我们不用太多关注具体的公式

The exact expression won’t be a matter for us.

我们要关注的是 这个过程输出了一个正交向量集合

What matters is that the procedure creates a set of orthogonal vectors.

这个正交向量集合具有很多重要的性质

And it has lots of nice properties.

这里是两个向量 你可以想象一下三个向量的情况 此时正交投影应该垂直于屏幕

So these are two vectors. You should imagine what happens if there are three vectors, like the third vector is here, projected orthogonal to the board.

这个过程与格的关联性是这样的

One nice thing connection with lattice, is this.

Vadim会用到这个关联性 有些人在预习这个讲座的笔记时估计已经看到了

And this is something Vadim will use, maybe someone basically in this note, in this talk,

这个关联性称为Gram-Schmidt基本区域

what I would call the Gram-Schmidt fundamental region.

我给大家展示了很多基本区域 比如天马什么的 这是另一个基本区域

So, I already showed a lot of fundamental regions, horses and stuff. But this is another one.

这个基本区域很有用 一般用于命题的证明

It’s sometimes useful, sometimes shows up in proofs.

我们在格中选一个基 比如这样的一个基 这个基包含两个向量

So let’s take a basis. Let’s take a lattice in a basis, like a basis over there. I have these two vectors.

这两个向量定义了一个基本平行多面体 也就是这个区域

And they, you know, define the fundamental parallelepiped. The one over there.

我们知道这也是一个基本区域

And we know it’s a fundamental region.

如果我把这个区域放在所有格点上 区域就能覆盖整个平面

If I would repeat it, it would get this nice… continues in all directions.

幻灯片上我只列了这些点 但是这个区域覆盖了整个平面

I just had some points, I had to stop. But, this is really nice, covering over the whole plane.

我现在取这个基的Gram-Schmidt正交基

But you could also now take the Gram-Schmidt Orthogonalization of that basis.

你会最终得到两个向量 v\_1和v\_2波浪 或者说v\_1波浪和v\_2波浪

And you end up with two vectors, v\_1 and v\_2 tildebar. So, v\_1 tildebar and v\_2 tildebar.

其中v\_2波浪不再是一个格点了

v\_2 tildebar is no longer a lattice point.

考虑这个正交基 这也是一个平行多面体 但是是由正交向量生成的 我称之为方盒

But, if you consider the box that to generate. So consider this box. It’s also a parallelepiped, but generated by orthogonal vectors. It’s what I would call it a box.

这也是一个基本区域 产生的区域是这样的 有点像砖块

This is also a fundamental region. This is what it creates. It’s a brick wall region.

这并不是巧合 不难证明Gram-Schmidt正交基可以生成一个基本区域

It’s not coincidence. This is always ones, and not difficult to prove that, also, the Gram-Schmidt generates a fundamental region.

这有时候会很有用 在下一个讲座中我们会讲解这个基本区域的作用

This is sometimes useful. And so we should sort of show up in the next talk.

下一页幻灯片中我会稍微讲解一下它的作用

So maybe I'd say, in the next slide, say a little bit what this happens.

我们从数学的角度看看会发生什么 从数学的角度很容易体现它的作用

I try to show mathematically what it happens. It’s easier to show them mathematically.

接下来…

OK, so…

我们现在已经有了一种方法 可以通过基产生一个正交向量集

So, now that we have a way to generate from any basis of orthogonal set.

下一步我们可以把向量规范化 就能得到规范化正交向量集了

The next logical thing to do will be to normalize vectors, and have orthogonal normal set, right?

我们可以取任意一个格基 v\_1, ..., v\_n

This allows us to take any lattice basis, v\_1, ..., v\_n.

使用Gram-Schmidt正交化 得到正交基 再规范化这个正交基

So, apply Gram-Schmidt orthogonalization in our normalized vectors.

这样我们会得到v\_1波浪除以v\_1波浪的范数 一直到v\_n波浪除以v\_n波浪的范数

And this brings to you the vectors v\_1 tildebar divided by normal v\_1 tildebar, and so on, up to v\_n tildebar divided by normal v\_n tildebar.

这也是一个规范化向量集

This is also normal vectors.

它们也不是格点 我们不需要让这些向量为格点

Again, they’re not lattice points. There’s no need for them to be.

这个基有什么好的性质？通过基思考一些定义和定理会更容易理解

But, what’s really nice about this basis? It is often very useful to think in terms of this basis, is that…

这个基是线性代数层面上的 不是格层面上的 它是R^n的一个基

So now this is a basis in the linear algebra sense instead of in a lattice sense. It’s basis of R^n.

这个基的好处在于

And what’s nice about this basis is

如果你把格基v\_1, v\_2,…, v\_n写成这种基的形式

if you now try to write your lattice basis, v\_1, v\_2,…, v\_n, try to express it in this basis.

则列向量会恢复为格向量 但列向量被R^n上的基调整了一下

Not the columns are going to be now the lattice vectors, but written in this basis so far R^n.

这样表述后 基就不再是一个正交基了 而是把格基用R^n中的基旋转了一下

It’s not in this down basis any more It’s more just kind of... I rotated everything. I rotated them using different basis of R^n.

这样表示后 格基将成为非常好的上三角形式

And it ended up with something very nice. And it ended up with upper triangular form.

这种表述形式非常有用 后面大家会看到

It’s extremely useful, as you see.

我们现在要证明里面的两个性质 这两个性质也体现了这种基表达方式的好处

We’ve now proved two things using it, because this is a very nice way to think of lattice basis.

取(v\_1, v\_2,…, v\_n)这个原始矩阵 然后对这个基做变换

So I should say all of that, I took the matrix (v\_1, v\_2,…, v\_n), the original matrix. I just changed my basis.

本质上我在这个基的左边乘以一个正交矩阵

I essentially multiplied from the left by an orthogonal matrix.

所以实际上我什么都没做 只是把格基用R^n上的正交向量旋转了一下

So essentially, I didn’t do anything. I just view the same vectors in different basis of R^n, different orthonormal basis.

格基的所有性质保持不变 因为正交基表示空间旋转 这样做会发生什么呢？

So, everything stays the same. All the orthonormal bases suggest a rotation of space. And now notice what’s happen?

举例来说 向量v\_1非常简单 向量v\_1在这里…

For instance, the vector v\_1 is very simple. The vector v\_1 in this…

现在 v\_1的第一个坐标是一个非0数 其他坐标均为0

Currently, this basis just looks like some non-zero number in the first coordinate, and then zero is everywhere.

这是因为根据正交变换定义 v\_1波浪就等于v\_1 v\_1就在v\_1波浪的方向上

This is just because the vector v\_1 is, by definition, the same as v\_1 tildebar. It’s in the direction of v\_1 tildebar.

所以我们已经知道v\_1只在第一个轴上有坐标

So, we already know that v\_1 is only in the first axis. OK?

根据定义 第二个向量v\_2是在第一个向量生成空间中的…

Now the second vector v\_2, we can, by definition, is in the span of the first…

我们往回看一下

Let's go back to show this.

在这里 v\_2在v\_1波浪和v\_2波浪的生成空间中

Here. So v\_2 is in the span of v\_1 tildebar and v\_2 tildebar. It is a span.

我们可以这么解释 如果是二维空间 任何一个向量都在这两个向量的生成空间中

If it’s in two dimensions, everything is going to be in the span of these two vectors.

大家可以想象一下 在高维空间中会是什么情况

But you could imagine, maybe in high dimension how this would work.

第二个向量中 第一个坐标是非零向量

So again, the second vector is some non-zero elements in the first coordinate.

第二个坐标就是v\_2波浪 v\_2波浪的范数就是第二个向量的第二个坐标

And then, the second coordinate, as you can see, has v\_2 tildebar. The normal v\_2 tildebar is what it has in the second coordinate.

所以这只是同一个基的另一种表示方法 但这种表示非常方便

So, this is simply a different way to write the same basis. But this is very convenient.

第一点是 不难用这种表示方法来证明Gram-Schmidt方盒是一个基本区域

So one thing, perhaps, to try to prove, using that, is to try to prove that the Gram-Schmidt box is a fundamental region. I want to show this is actually not difficult.

我会给大家两个引理 用这两个引理就可以证明出来

I’ll show you two other lemmas so that you can prove using that.

我要提醒大家的是 这种表示方法有时也被称作QR分解

I’ve just thought I should also mention this is sometimes known as the QR decomposition.

如果你在线性代数课程中听过的话 QR分解就是将一个矩阵正交化

If you’ve seen that in linear algebra, this is kind of the decomposition of a matrix into orthogonal.

这是Q部分 R部分是一个上三角矩阵

This is the Q part, and R is the upper triangular part.

当把QR分解用到格基上时 他就有了更为特殊的意义

But it gets a special meaning once you apply it to lattice basis.

我现在要证明两个结论

The two things I want to prove.

第一个结论很容易证明

The first one is quite easy,

这个结论称 由v\_1, …, v\_n生成格的行列式等于所有v\_i波浪范数的乘积

says that the determinant of the lattice generated by v\_1, …, v\_n is simply the product of the norms of v\_i tildebar.

这个问题你可以这么看

You can essentially see here.

这是一个上三角矩阵

That is an upper triangular matrix.

上三角矩阵的行列式等于对角线上元素的乘积

So, the determinant of upper triangular matrix is same to the product of things in the diagonal.

所以我们就得到所有v\_i波浪的乘积 所有v\_i波浪范数的乘积

So, we get the product of all the v\_i tildebar, the norm of the v\_i tildebar.

这个结论很容易证明 有时候会用得到这个结论

So this is very easy. This is sometimes a useful fact.

第二个结论可能更有用些 用处更大

Maybe more useful fact is the second one. This is much more useful.

这个结论称λ\_1 回忆一下 λ\_1是第一个连续极小 也就是最短非零向量的长度

And this is that the λ\_1, which is the… Recall, this is the first successive minima. It is the length of the shortest nonzero vector.

λ\_1最短也不会小于最小的那个v\_i波浪范数长度

It has to be at least the minimum of the v\_i tildebars, the norm of v\_i tildebars.

我们盯着这个矩阵 看看怎么证明这一结论

So, let’s try to stare this matrix, and prove it.

回忆一下 什么是格？

So recall what is the lattice?

格是这个矩阵列向量的整数组合 所以这个矩阵的列向量生成了格

The lattice is simply all integer combinations of the column of this matrix. So this matrix is the column generated the lattice.

这种表示方法只是把基旋转了一下 本质上没有改变格基

All it did was simply rotating a bit of our basis. But it doesn’t change anything.

想象一下 这就是我们的格 由矩阵列向量的整数组合生成的格

So this is… Imagine this is our lattice generated by the columns of this matrix by all the integer combinations of the columns of the matrix.

首先要注意到 这是一个上三角矩阵

So the first thing to notice is that it is an upper triangular matrix.

只要最右边列向量最下面的值不为0 列向量的长度最短为v\_n波浪的范数

The first thing to notice if you use a nonzero co-efficient on the rightmost column, if you use a nonzero co-efficient in the rightmost column, then surely the length of the lattice vector you’re going to get is at least the norm of v\_n tildebar.

注意到最右边的列向量是最后一个线性无关格向量

And you see that, just because the last coordinate.

只要此列向量最下面的值不为0 则最短向量的的长度至少为v\_n波浪的范数

If I take here, a nonzero co-efficient here, then the shortest vector here of the last point is at least the norm of the v\_n tildebar.

即使前面坐标的系数都能互相消去 向量的范数中一定含有最后这个坐标的范数值

So the norm of the vector is also needs to be this norm of tildebar, even if all the previous ones can be cancelled and become zero.

很容易得到这个结论

This is easy, right?

如果最右边列向量最下面的值不为0 就很容易得到这个结论

If the last column is not zero, the last co-efficient we take is a non-zero, it’s easy.

如果是这种情况 我们知道最短向量长度至少为v\_n波浪

We know that this is at least v\_n tildebar,

所以第一个连续极小值最小也不会小于v\_n波浪

which is at least biggest successive minima between the v\_n tildes.

如果最右边列向量最下面的值为0呢？

So what happens if it is zero, if the last co-efficient is zero?

我们考虑其左侧的列向量

Then consider the next tildebar,

这个列向量最下面的值为0 如果其倒数第二个值不为0

if the next tildebar is non-zero, now the last one is zero, so the next tildebar is non-zero,

则λ\_1至少要大于v\_(n-1)波浪的长度

then the next tildebar’s coordinator here come to be the norm of v\_(n-1) tildebar, at least in absolute value,

结论也成立 以此类推

which is also good, and so on.

我们实际上在证明什么？

Essentially what you’re doing?

我们在对列取任意非0系数的的整数组合

You’re saying take any non-zero combination of the columns, integer combination of the column,

整数最后最后一个列的系数非0

take the last column that has a non-zero co-efficient,

这样最后一个列会为λ\_1提供一个范数 λ\_1要大于v\_i波浪的范数 证明完毕

and that column don’t give you here an entry that’s bigger than the minimal of the norm of the v\_i tildebar. And then you’re done.

总能找到一个非0系数 使得最短非零向量中的某一项不能被消掉

It is always the last non-zero gives you the entry that cannot be cancelled anywhere

因为这是一个上三角矩阵 对角线下方全为0 肯定有一项消不下去

because it’s upper triangular and it never appears. Nothing else can be cancelled.

这个结论非常基础

So, this is a very basic statement.

但这个结论非常好 因为我们通过这种方式得到了λ\_1的一个下界

This is nice because this is one very useful way to get lower bounds on λ\_1,

用其它方法很难从格基中得到λ\_1的下界

which otherwise difficult number to get handed on, to lower bounds λ\_1.

可以通过这种方式得到λ\_1的一个下界 这个结论还和LLL算法有一定的联系

So this is the way you will do it. Useful in the connection with the LLL Algorithm.

这就是Gram-Schmidt正交化

So this is it.

我们还可以证明其他一些有趣的结论 这就是Minkowski定理

Now we can try to prove something more interesting. This is Minkowski’s Theorem.

我们先从一个基本定理讲起 这个定理是由Blichfeld提出的

We start with another theorem, one more basic one by Blichfeld,

Blichfeld定理的内容如下

and it said the following thing.

考虑任意一个格 现在我们用Λ表示格 而不用L

So now, consider any lattice, now use Λ, not L.

但这两个符号都表示一个格 只是表示方法不一样而已

But it’s the same, it’s the same idea, just different notation.

选一个格Λ 令S是任意一个集合 要求S的体积大于格的行列式

So, take a lattice Λ, and not let S be any set whose volume is bigger than the determinant of the lattice.

这个定理说明 在集合S的内部一定能找到两个点 这两个点的差是一个格点

And what the theorem says is then inside the set, you must find two points whose difference is itself, the different vector is itself the lattice point.

注意我们这里要求集合S的体积大于格的行列式

Notice that is necessary to require the volume to be bigger than the determinant,

因为如果我们任意取一个基本区域的话

because we call that if you take any fundamental region,

基本区域中可能找不到这么两个点 使得点的差值为一个格点

then in the fundamental region, you never have two points whose difference is the lattice point.

因为在基本区域中有且只有一个格点

Totally for any cause if only have one and only one representative,

如果取基本区域的话 这个结论不成立

and this will never happen when you take the fundamental region.

所以我们需要集合S的体积大于行列式

So it is necessary to require a volume bigger than the determinant.

这就是这个定理的证明方法 用几何方式来证明这一点

And this is how you have proved it. So the idea is actually very geometrical.

我们选一个袋鼠 这是一个很肥的袋鼠

We take a kangaroo, and they really very fat, very fat kangaroo.

袋鼠的体积大于格的行列式

And the volume of the kangaroo is bigger than the determinant of the lattice.

幻灯片上的图像很形象 因为这是个二维空间 我不想把示意图画的太复杂了

So the picture is pretty because it’s in two dimensions, because I want to not to make it too ridiculous,

图中的袋鼠很肥 它的体积大于格的行列式

but figures of this kangaroo is being fat, and being bigger than the determinant of the lattice.

如果你仔细看的话 可能会觉得袋鼠的体积还是有点小 但实际上已经足够大了

Actually I think it is, if you look at it, it seems like much smaller, but is being bigger.

现在我们有了一个袋鼠 这个袋鼠就是我们取的集合S

So, I have a kangaroo, and now what I’d like to do is… This is my set S, ok, so the S is the kangaroo’s set S.

我现在要做的是让这个集合S覆盖整个平面 也就是把这个袋鼠图放在每一个格点上

And now what I want them to do is I want to repeat this set cover all over the place. So take translations of the set by lattice points.

所以我在另一个格点上也放一个 所有的格点上都放一个

So I take another one here. And you know, bring all the others in.

很好 这是个袋鼠动物园了 他们都放置在了格点的相同位置上

Great. There is a zoo of kangaroos. And they all should be exactly positioned in the same place,

只不过根据格点的位置来平移而已

just shift them by lattice points.

现在我们可以观察一下如何形象地证明这个定理

And the cube of the vision now to make this work is that,

因为袋鼠很肥 袋鼠的体积大于格的行列式 因此总会有重叠的区域

because the kangaroo is fat, because it is bigger than the determinant of the lattice, It has the room. It has to overlap with itself.

大家想一想 格的行列式恰好等于每个格点周围预留的空间体积大小

Because remember, the determinant of the lattice is exactly that kind of the volume that dedicate to each lattice point,

如果所选的集合体积大于这个体积 那就没地方了

and if you’re fatter than that, there’s no room.

可以很容易证明 袋鼠和袋鼠之间一定有重叠的区域

And you can prove that, you can easily prove it, that they must be the intersection between the kangaroos.

一定会有重叠区域 空间中没那么大的地方

They can be joint, they can be by volume arguments. That’s just not enough room.

如果取一个非常大的空间 你有一大堆袋鼠 每个的体积都很大 袋鼠就没地方放了

If you take it, say large area of space, you’ll see too many kangaroos and too much volume. And there’s nowhere for them to fit.

从图上看的话 你会看到一些重叠区域

So now you take this picture, now you know this space for the intersect.

比如这个区域和上面袋鼠的耳朵…我不太熟悉袋鼠的生理结构…

For instance, you see this section here, you see that this area between the ears of the kangaroo… I’m not good at the kangaroos…

上面这只袋鼠的耳朵和下面这只袋鼠的脚后跟重叠了

and not only, but this area in the ears is the same as the… what’s it like, the ankle of the kangaroo.

所以袋鼠耳朵和脚后跟部分有重叠的区域

So the intersect, the area between the ears bring the intersect with the ankle of the kangaroo.

我们现在要做的是观察重叠的部分

And what I’m going to do now is look at in the intersection point

我们在一个袋鼠的重叠区域中取两个点

and take one kangaroo and take this two points on one kangaroo.

比如取这只袋鼠 在耳朵上取一点 再在脚后跟上取一点 画一条线

I take this, I take the area between the ears of one kangaroo and also the ankle of the same kangaroo and I draw a line.

注意到这两个点是在一个袋鼠里面的 在一个集合S里面的

And all of the domains… So notice that these are two points inside the same kangaroo, inside the same S.

那么这两个点的差所构成的向量就是格中的向量

And the difference between these two points in this vector is a vector of the lattices.

这是一个格向量 这是因为我们是按照格向量来平移袋鼠的

This is a lattice vector. This is because, as I told you, the translation of the kangaroos is via lattice vectors.

所以在上面的群中和中间的群中各取一点 它们的差值也是格点

OK, so the difference between the topping group and the middle one is a lattice vector.

总会形成这样一个向量 你可以证明这一点

It has a vector, as you can see it here, as something you can prove.

这个向量就是格向量

But this vector is really the lattice vector.

我们现在讲的是一个非常基础的结论

So what we show this very basic statement,

只要所取集合的体积大于行列式 这里面一定有两个点 他们的差值是一个格点

whenever we have something of the volume bigger than the determinant, it must inside contain two points whose difference is the lattice point.

这个结论证明了Minkowski定理的90%了 我后面会讲剩下10%怎么证明

And this is all the 90% of the work towards proving Minkowski’s theorem, which is the next thing I’d like to mention.

Minkowski定理是这样描述的

This is the Minkowski’s theorem.

取任意一个格Λ 然后再取一个集合S 现在S要满足一定的条件

It says that if you take any lattice, again Λ, and you take any set S, but now S has some requirements on that,

现在这个集合S不能是一只袋鼠了 它需要是一个凸集合

so that’s cannot be kangaroo any more. It should be a convex set.

比如幻灯片上的这个集合 是一个凸集合

You should see something like that, the convex set,

而且这个集合必须是关于原点对称的

And it should be also symmetric around the origin.

即如果x在S中 那么-x也在S中 这就是关于原点对称的意思 即S与-S相同

So it means that if x in S, also –x in S, so symmetry around the origin. So S is the same as –S.

比如幻灯片上的这个集合

So something like the body over there.

那么定理表明 如果S的体积足够大 不仅大于行列式 要大于2^n乘以行列式

Then what the theorem says is that if the volume of S is now bigger, not just bigger than the determinant, but actually bigger than the 2^n times determinant,.

那么集合S中一定包含一个非0格点

Then there is actually, I should say a non-zero lattice point in S.

我要强调包含一个非0格点 很明显 0点也一定在S中

I should have said non-zero, otherwise it’s obvious, 0 is obvious in S,

因为集合S关于0点对称 集合S中一定存在一个非0格点

because we assume the S is the symmetric around zero. Yes, there exists a non-zero lattice point in S.

这就是Minkowski定理

So this is Minkowski’s Theorem.

这是一个很强的结论 它告诉我们在特定体中一定包含一个格点

This is a very powerful statement. It also tells us there must be lattice point inside certain bodies.

当然 为了证明可以通过 我们所取体的体积很大 但这个结论依然非常重要

And it tells us… So the simple proof for some big field. But this is quite a remarkable statement.

证明方法如下

Here is the way to prove it.

在证明之前我可能要解释一下为什么我们要求是一个凸集合

Maybe I should… Before proving it, I should say why it needs the set to be, say, the convex,

如果集合不是一个凸集合的话 大家可以想象一下

if the set is not convex, maybe you could imagine

我们可以在格点的空隙中构造一个集 这个集合不会碰到任何一个格点

the certain kind of goals in between the lattice points, and never touches lattice points.

集合可以很大 体积可以特别大 但是碰不到任何一个格点

The set can be huge, can have a huge volume but never touches any lattice point.

但是在凸集合中 我们就构造不了类似这种奇怪的集合了

because on the convex you cannot do strange things like that.

我们还要求它是一个0点对称集合

And it also needs to be zero-symmetric.

你可以想象一个不是0点对称的凸集合

You can imagine the convex but not zero symmetric set,

比如一个很高的三角形 三角形碰不到任何整数格点 但体积也可以非常大

like a very tall rectangular, never touches the S squared, never touches the integers, but still has a huge volume.

所以凸集合的确是个必要条件 并不是因为我或者Minkowski很懒 非要加这么个条件

So convex form is necessary, not just because I’m lazy, or Minkowski was lazy.

下面就是证明过程

And here is the proof.

第一个思想是取一个集合S 然后缩小2倍

So the first idea is to take S and shrink it by factor 2.

我这里取一个集合S 然后在每个维度上都缩小2倍

So I take S and I kind of shrink it by factor 2 in each dimension.

我就得到这样一个集合 我们叫它S/2

So I get the set, which I called it S/2.

我们现在有了集合S/2和S

Just take all S/2 for S and S.

我们首先要注意集合S/2的体积 我们是在n个维度下对S进行缩小

One thing to notice, which is maybe that the volume of S/2, is not to have the volume of S with the n dimensions.

如果每个维度都缩小了2倍 那么体积会缩小2^n倍 就像我们缩小球体体积一样

So if you shrink something by 2, the volume shrinks by 2^n, like from former forms for area of the serious lower for volume of spheres.

因此S/2的体积比Λ的行列式大

So S/2 therefore has a volume bigger than the determinant of Λ,

根据前一个定理 我们知道S/2中一定有两个点z\_1和z\_2

and by the previous theorem we know that there are 2 points, you see, z\_1, z\_2, inside S/2, inside this half S,

z\_1减z\_2所形成的向量是一个格点

and the difference with the vector between them, z\_1 minus z\_2 is the lattice point.

到这里还没结束 因为我们要在S中找到这个格向量

Why we’ve not done because we want an actual vector, the lattice vector inside S.

我们现在只在S/2中找到了2个向量 它们也在S中 且差值为格点

Currently we just have 2 vectors in S/2 was differences inside S.

但是我们差不多找到了 但还需要做下面两步操作

But that we’ve almost done. We just have to do one more step, or two more steps.

因为z\_1在S/2中 根据定义 2倍z\_1也在S中

Because z\_1 is in S/2, by definition twice of z\_1 is in S, just by definition.

所以我们把z\_1乘以2 我们得到2z\_1在S中

So I multiply z\_1 by 2. I get 2z\_1 in an S.

类似地 我们取z\_2 乘以2 再取负数

And similarly, I take z\_2 and I multiply by 2, also negated.

我们可以取负数 因此我们取的集合关于0点对称 所以-2z\_2也在S中

It’s also negated because I know that my set is zero-symmetric. So -2z\_2 is also in S.

我们现在知道2z\_1和-2z\_2都在S中

So now I have both 2z\_1 and -2z\_2 are both in S.

这有什么好处呢？

And why that’s so nice?

现在我们可以取这两个点的平均数 平均数也一定在S中 因为这是一个凸集合

Because now the average, take the average of these two points, which must be in S, because this is a convex.

所以我们在这两个点中取一个中点 正好等于z\_1 – z\_2

So take the average between these two points, and what’s the average of these two points is exactly on z\_1 – z\_2.

2z\_1 - 2z\_2的中点 这两个点的中点就是z\_1 – z\_2

OK, the average of 2z\_1 - 2z\_2, the average of these two things is z\_1 – z\_2.

我们知道这是一个格点 因为证明的第一部分说明z\_1 – z\_2是一个格点

And we know that this is a lattice point, because of the first item of the proof said z\_1-z\_2 is a lattice point.

这就是Minkowski定理的证明方法了

So this is the proof of Minkowski’s Theorem.

这个定理在格中怎么用呢？

And what is essentially lattice to do?

这个定理实际上说明格中有一个短向量 或者至少说明格中有这样一个向量

The lattice is to say that the lattices have short vectors, lattices… to say that lattices must have the vector of something like that.

我们要用这个定理的一个推论 这就是推论了

This is what you can use the corollary where you can derive the corollary now. Here is the corollary.

取任意一个格Λ Λ的最短向量最长不会超过√n乘以Λ行列式的1/n次方

Take any lattice Λ, it tells the shortest vector of Λ must be a short vector of length at most square n times the determinant of Λ to the 1/n , to the power of 1/n.

如果大家第一眼看到这个推论

Because if you can see for the first time,

可能会觉得Λ行列式的1/n次方很奇怪 但实际上这是很自然的结论

maybe I should say the determinant of Λ to the 1/n is actually very natural,

我们可以忘了这个1/n次方这只是一个放缩系数而已

you actually forget it, it cannot has to be there. This is simply a scaling factor.

假设我们取一个格Λ 然后把它放大10倍

So imagine I take a lattice Λ and I just scale it by factor 10,

那么所有的参数都会放大10倍 这样等式左边肯定就被放大了10倍

just scale everything by factor 10. So the left-hand side obviously is just also scaled by factor 10.

为了能够让等式正确放缩 我们需要等式右边也放大10倍

So for this inequality to make sense to scale properly, I also want the right-hand side to be scaled by factor 10.

这就是行列式1/n次方的由来

This is what the determinant to the power 1/n does,

因为如果我们把格放大了10倍 根据矩阵的性质 格的行列式会被放大10^n倍

because when I scale the lattice by factor 10, the determinant goes up by factor 10^n, the determinant of n by matrix.

所以我们要取1/n次方 这样等式左右两边放大的倍数就一样了

It takes a part 1/n hence both left and right hand side,

等式两边都放大了10倍 这是我们希望得到的结论

both of them scaled by factor 10, this is what we expect.

所以对格放缩和对矩阵放缩是一样的 没什么区别

You don’t expect anything to happen when you scale the lattice.

行列式的1/n次方是非常自然的结果

So kind of the determinant to 1/n is very naturally.

必须要取1/n次方 只是一个放缩系数

It should be there, it’s very… just a scaling factor.

比较有趣的是√n这部分

And the interesting thing is just square root of n.

这个推论的一个等价论述是

So it tells that an equivalent way to say this corollary is that

对于任意行列式为1的格 一定有个向量 它的长度最长不会超过√n

any lattice of the determinant 1 must have a vector of length to most square root of n.

我们可以把这个常数再优化一点

Actually you can improve the constant a bit.

如果你觉得这个常数还不够好 你可以把这个常数再优化得稍微好一点

So if you really worry about the constant, you can have a slightly smaller factor,

我们可以把系数优化为C乘以√n 其中C小于1

（注：这里应该是Oded的口误 C应小于1）

you can have C times square root of n for C is less than one.

但我们不用关心这一点 这个优化点本身没什么意思

But we usually don’t care about that too much, if you notice it’s not an interesting question.

怎么证明这一点呢？

So how do you prove that?

这个推论只是前面那个定理的特殊情况

To the proof, it is a special case of the previous theorem.

我们取一个半径为√n的球 注意这个球的体积大于2^n

We just take a ball of radius square root of n and only have to notice that its volume is greater than 2^n.

如果你取任意一个行列式为1的格且半径是√n的球 体积大于2^n

So if you I take any lattice of the determinant one, then the ball of radius square root of n

则球中一定包含一个格点 这个格点的长度一定小于球的半径 小于√n

has volume greater than 2^n, hence must contain a point. And any point it’s in the ball of radius square root of n

根据定义我们可以直接得到 这个向量的长度最多不会超过√n

is the by definition has norm at most square root of n.

为什么半径为√n的球 球体的体积至少为2^n呢 因为这个球内部包含一个方盒

And the reason of ball of radius square root of n contains if it has volume at least 2^n is because

方盒的每一个维度都为[-1,1] 这个方盒为[-1,1]^n

it contains this box [-1, 1]… the interval is [-1, 1]^n.

为什么？因为任意一个在方盒中点的范数最大不会超过√n

Why? Because the norm of any point inside the box is the most square root of n

最大的范数也只能属于一个点 这个点的范数等于√n

The largest norm is the only one point, which has norm square root of n.

大家可能觉得这个定理证明的时候 浪费了很多空间 但实际上并没有浪费

It seems very wasteful, but actually it’s not the wasteful argument.

球体的体积并不比2^n大多少

The volume of the ball is not much bigger than 2^n.

我在这里就不详细说明这一点了

So I lost to be about that.

本质上 这个阶最多到√n 或者√n前面乘以一个常数

But it’s essentially square root of n up to a constant before this square root of n.

现在我们找格中短向量的方法都是从数学角度取找

So what we manage to do here, finding short vectors in lattice is something that so far we can only do mathematically.

我们不知道如何从算法角度找格的短向量

We don’t know how to do it algorithmically, exclusively.

这也是为何可以基于格构造密码学方案的本质原因

This is essentially what crypto is based on.

格密码学所依赖的假设是 这些问题是困难问题

The lattice-based crypto is based on the assumption these things are hard,

我们不知道如何高效地求得格的短向量

we don’t know how to find such vectors efficiently.

如果你仔细想想的话 这个假设很神奇

If you notice and look more carefully, this argument is some kind of magic.

我们已经讨论了好几年了 格问题究竟是不是个困难问题

Some non-explicit argument that went in years, some kind of original principles,

感觉这个困难问题和Blichfeld定理是冲突的

some kind of collisional argument that went in the Blichfeld’s theorem.

我们可以利用方盒做很多事情 证明很多定理

We just had too many things that they can fit into one box.

但是这个定理没有告诉人们如何得到一个短向量

But this theorem doesn’t give you any way to actually find the shortest vector.

这个定理只告诉人们存在一个短向量 我们不知道如何高效地得到一个短向量

It tells you the short vector exists. But we have no idea how to find it efficiently.

而得到短向量的困难性就是密码学所依赖的困难问题

And this is the difficulty to find this thing efficiently is what we are building the cryptography on.

这部分有什么问题吗？

Questions on this part?

很好 我们现在开始讲最后一部分内容 我们来看看格中的计算问题

Good. OK. So, this now brings me to the last topic of the... Today is… We show computational problems.

这实际上是一个很大的领域 我们的讲座不可能涉及到所有的方面

This is actually, this is huge area. Don’t think it can cover any wedges.

我这里只是希望让大家看一下 这个领域发展到了什么地步

I just want to just give you a glimpse on what’s happening in that area

后面几天我们要用到这些困难问题

Usually, in the next few days, we just, you take these problems.

假设这些问题是困难的 基于这些问题构造密码学方案

Assume they are hard, and do crypto based on.

我们不需要太多关注困难问题本身

They don’t want to worry about the problems too much.

Vadim的讲座会稍微详细讲一下这些困难问题 我在周三的时候也会讲到

We’ll see a tiny little bit in Vadim’s talk, may in my talk on Wednesday.

但我认为绝大多数讲座不会过多讲解这些问题本身

But I think most of the talks won’t have to deal with these problems too specifically.

我们来看看要使用的困难问题吧

So let’s see what the main problems are.

首先 我们来看看一些比较简单的问题

First, maybe this starts with some easy problems.

首先 这是一个简单的格问题

First, this is an easy problem. It’s the easy lattice problem.

如果给你一个基和一下向量v 你是否可以验证v是否在格L(B)中？

If I give you a basis and a vector v, can you check if v is in the lattice L(B)?

我们如何验证？如何解决这个问题？

How do you do that? How do you solve that?

我们可以用高斯消元法 很好 可以用高斯消元法解决这个问题

Now you can do any Gaussian elimination, good. Now you can do Gaussian elimination.

取这个向量 这里我给定的向量是v

Take the vector, the vector I gave you as v.

然后把这个向量表示成基向量v\_1, …, v\_n的线性组合形式

And just express it as the linear combination of your basis vectors v\_1, …, v\_n.

这种表示方法是唯一的 可以用高斯消元法得到这种表示

There is unique way to express it, then you can find it using Gaussian elimination.

v\_1, …, v\_n是线性无关向量 只用最基本的线性代数就能做到

v\_1, …, v\_n are linearly independent, just basic linear algebra.

然后我们要做的是检查系数是否均为整数

And now what we have to do is just check if the coefficients are integer?

如果系数是整数 就是格点 如果系数不是整数 就不是格点

If the coefficients are integer, it’s a lattice point. If not, it’s not a lattice points.

这很容易验证 在后面的讲座中我们还会用到这个算法

So it is very easy to do, and we actually need that in some of the talks.

这个算法很有用 记住它

So this is a useful fact, keep it in mind.

这个问题之所以简单 因为它不是一个几何问题 而是一个线性代数问题

And kind of the reason this is easy is because it’s not geometrically. It’s really the question of linear algebra, not the geometrical question. This is why it’s easy.

与之类似的一个线性代数问题是

A similar equation, also a linear algebra question is

如果我给定你两个基B\_1和B\_2 你能否判定 是否这两个基生成的格是同一个格？

if I give you two bases B\_1 and B\_2, can you tell if they are the same, generate the same lattice?

我们有很多方法可以解决这个问题

So there are several ways to solve it.

第一种方法是用我们前面得到的一个结论来解决

And one of them was using what we’ve solved before.

我讲过 B\_1和B\_2是等价的 当且仅当B\_1等于B\_2乘以U 其中U是一个幺模矩阵

I told you that they are going to be equivalent, B\_1 and B\_2 are equivalent if and only if B\_1 is B\_2 times U for some unimodular U.

所以你可以试着求一下U 计算B\_1乘以B\_2的逆

So you can try to compute U. So you take B\_1 times B\_2 inverse.

你会得到一个矩阵 检查一下这个矩阵是不是幺模矩阵

You get the matrix. You check if it’s unimodular.

可能我们根本不需要用这么复杂的方法解决这个问题

Maybe this is a bit unnecessarily complicated.

还有另一种更简单 更基本的方法来解决这个问题

You can maybe do its more basic at first principles.

这另一种方法是取基中的任意一个元素 任意一个向量

One way to do, another way to do it would be take any element of the basis here, any vectors,

对于每一个向量 检查这个向量是否属于另一个格

for each one, check if it’s in the other lattice.

所以可以取n个基向量 对于每一个基向量 检查这个向量是否属于另一个格

So take the n basis of elements, for each one, check if it’s in the other lattice.

如果每一个基向量都属于另一个格 你会得到L(B\_1)包含于L(B\_2)中

If they are, you know that L(B\_1) is contained in L(B\_2).

由于加法封闭性 如果基向量在L(B\_2)中 则基向量的全部整数组合也在L(B\_2)中

because if each of the basis element is in L(B\_2), also all the integer combinations, and lattices are always close under addition.

我们反过来再检查一遍

And you just do the same thing in reverse,

取B\_2中的每一个基向量元素 检查每一个基向量元素是否在L(B\_1)中

take each of the basis elements of B\_2, and check if it is in L(B\_1).

如果这也满足的话 你就得到这两个基是等价的

And if this is also the case then you know they are equal,

因为第一个格属于第二个格 而第二个格又属于第一个格 所以这两个格等价

because one contained in the other and the other contained in the first, then you know they are equal.

还有很多方法可以解决这个问题 这几种解决方法都很简单

There are lots of ways to solve it. Both of things anyway are easy.

我们现在开始讲一个比较困难的问题

The difficulty comes in once we start…

一旦格问题是一个几何问题 比如讨论长向量 短向量什么的

Once we’re trying to do something geometrical. Argue about long vector, short vector,

那么这个问题就会变得很难 比想象中要难得多

then it becomes harder, much harder it seems.

这是一类典型的格问题 这个问题叫做最短向量问题

And this is kind of the prototypical problem. It is called, known as the Shortest Vector Problem.

这个问题是 给定一个格 我的意思是指定一个格

And what they ask is, I give you a lattice, which is also to mean I give you lattice.

大家需要理解一下何为给定一个格 这很重要 我们一般用一个基来描述格

This is actually important to understand. A lattice is always specified by a basis.

我指定格中任意一个基 这个基可能不是一个好基 基可能包含非常长的向量

I specify an arbitrary basis. Not a nice one, it can be like very long vectors.

现在我的问题是 你能否在这个格中找到一个短向量？

And then I ask you, can you find the short vector in this lattice?

你能找到v\_1, …, v\_n的一个整数组合

Can you find the combination of these vectors v\_1, …, v\_n,

使得各个系数能够互相消除 所得到的向量成为一个短向量？

but somehow cancel magically become a short vector?

这就是最短向量问题

And this is essentially the Shortest Vector Problem.

我们在密码学中常用的困难问题是它的一个变形 也就是近似最短向量问题

The actual, the variant that we need here that is useful in crypto is in the approximations of the variant.

这里有个参数γ γ是一个近似因子

So there is the parameter γ. γ is the approximation factor.

我要求找到一个向量 它可能不是最短向量 只要小于最短向量长度的γ倍就可以

So what I ask you to do is ask you to find the vector, that’s the not only the shortest, but only within the factor γ of the shortest.

这就是SVP\_γ问题

So this is the goal in SVP\_γ.

找到一个向量 这个向量的长度小于最短向量长度的γ倍 也就是γ乘以λ\_1

It is to find the vector whose length is at most γ times the shortest, γ times λ\_1.

在后面几页幻灯片中 我会讲到这个问题的当前研究进展

I’ll mention that in the next slide, a few slides on what is known about this problem.

这是一个典型格困难问题 想象一下这个问题的难度…

This is one problem. It’s kind of the prototypical. Imagine if you are others…

我需要指出的是 这个问题还有很多变种

The first I should mention, there is another variant of this question,

这个格问题要求计算得到这个短向量

which in here the lattice thing is the shortest vector, but we already knew that.

这个问题还有一个变种 这个变种问题像是一个判定性问题

And here is another variant of this question. This is more a decision version,

变种问题是一个间隔问题 我现在不让你找到这个向量

a gap version where I don’t ask you to find the vector,

你只需要告诉我近似最短向量的长度是多少

I just ask you to tell me the length of the vector. Actually I ask you the approximate length of the vector.

举例来说 我要求你告诉我 近似最短向量的长度小于1 还是大于γ？

So, for instance, I can ask you to tell me the length of the vector is less than one or more than γ.

我不会深入细节展开讲解

The details actually don’t even matter.

大家只需要知道 这个判定性问题在密码学中起到了重要的作用

But this problem, the decision version turns out to be more useful for crypto, usually.

大多数密码学构造是基于这个问题的

Most crypto constructions are based on this,

我还要讲一个问题 SIVP问题

or the next problem that I will show, the SIVP.

再强调一下 SVP和GapSVP这两个问题都涉及到一个近似因子γ

But in case both problems, SVP and GapSVP that has an approximation factor γ.

这是一个非常重要的参数 随着γ的变化 这个问题会变得更难或者更简单

This is a very important parameter, because as you change γ, the problem becomes either hard or easy.

当γ很大时 这个问题会很简单 因为我们不要求向量长度很短了

When γ is very big, the problem is easy, because I don’t ask you to do so much.

但是当γ很小时 这个问题就会变得非常困难

But when the γ is small, the problem seems to become very hard.

我在后面几页幻灯片中会详细讲解γ的取值范围

So we see exactly what’s going on about in a few slides.

我们再来讲几个格困难问题

So mention a few other problems.

这个问题在密码学中很重要

This problem is quite important in crypto

因为很多密码学构造 大多数密码学方案构造都依赖于这个问题

because somehow most constructions are, many constructions are based on it.

这个问题叫做最短独立向量问题

This is known as the Shortest Independent Vectors Problem.

这个问题涉及到的参数是λ\_n 和SVP问题有些类似

And this is simply the λ\_n, the analogue of the previous one.

SVP问题让你求得最短向量 或者求得γ倍的最短向量

So previous is asking you to find the shortest vector, something that’s within γ of the shortest vector.

SIVP问题中 我要求你找到格中的n个线性无关向量

And now what I ask you is to find n linearly independent vectors in the lattice.

并且要求这些线性无关向量的长度不能比最短线性无关向量长太多

And the goal is that for these linearly independent vectors to be not so much stronger than the shortest possible.

最短线性无关向量最长范数为λ\_n 这是由定义得来的

The shortest possible is λ\_n. This is by definition.

根据定义 我们能得到的最短线性无关向量最长范数为λ\_n

By definition of the λ\_n is the shortest radius where you can have,

这是n个线性无关向量所能得到的最短范数值

the shortest norm where you can have in n linearly independent vectors.

现在我让你找n格线性无关向量 他们的长度最多不能超过λ\_n的γ倍

And I ask you to find something that’s at most λ bigger, the most factor γ bigger.

我们举前面的一个例子 我们有这两个基向量v\_1和v\_2 它们生成了一个特定的格

So now again, we have these bases, v\_1 and v\_2, it generates a certain lattice.

我们不知道它们产生的格是个什么样子

We don’t know what lattice generates,

很难从计算的角度指出产生的格是什么样子

the computation is hard to figure this out.

在这种情况下 我要求找到两个短向量

And I ask you to find, in this case, the two vectors that are kind of short.

这实际上是最短向量

So this is actually, I think so, is the shortest.

这实际上就是λ\_2 不过一般情况下只要找到两个近似λ\_2长度的向量就可以

This is really the λ\_2, but in general, it can be some approximation factor away fromλ\_2.

你可能看到过其他一些格问题 我们在周二会讲到这个问题的一个等价定义

So if you mention a few others, we leaved at Tuesday, an equal definition.

所以我们不用记用这个方法定义的问题

So we don’t have to remember that.

不过这个问题很自然 叫做最近向量问题

But it’s quite natural so the Closest Vector Problem.

最近向量问题是指 我们有一个格

The Closest Vector Problem is now we have a lattice,

我们有一个点v 我让你找离点v最近的格点

we have a point v, and I ask you to find a nearby lattice point.

有多近呢？不一定要特别近 距离比系数γ倍小就可以

How nearby? It doesn’t have to be the closest. It can be within the factor of γ of the closest.

这个问题叫做最近向量问题

So this is known as the Closest Vector Problem.

我们有一个点 试着找一个离这个点比较近的格点

So, again, I have a point, and trying to find a nearby lattice point,.

不一定离得很近 和最近点的距离小于一个系数倍数就行

It doesn’t have to be the closest, but only within the γ factor of the closest.

这就是CVP问题 最近向量问题

This is the CVP, the Closet Vector Problem.

这两个问题实际有一定的关系 这是一个很好的习题

And one actually cute this off, this is a nice exercise, is this.

这个关系是由Goldreich, Micciancio, Sofra和Seifert在1999年发现的

It’s due to Goldreich, Micciancio, Sofra and Seifert from 1999.

这个关系可作为一个很不错的习题

It’s a nice exercise.

有很多类似的困难问题关系 但是这个结论更好 是格困难问题中的一个经典结论

They show, I mean, there are lots of now there are many exercises like that. But, this is specially a nice one. One of the classical results in this area, showing that

这个结论指出 对于任意近似因子γ SVP问题不比CVP问题难

for any γ, for any approximation factor γ, SVP is not harder than CVP.

换句话说 如果你能解决CVP问题 比如在近似因子等于10的条件下解决CVP问题

Or in other words, if you ever to solve CVP, if you have a way to solve approximate the Closest Vector Problem within the factor of 10,

那么你也可以在近似因子等于10的条件下解决最短向量问题

you can also solve the Shortest Vector Problem within the factor of 10.

如果在二维空间下考虑这个问题的话 这个结论挺直观的

If you think about it in two dimensions, it seems maybe, first, it seems obvious.

我们可以怎么解决这个问题？利用0点

What’s obvious thing to try? 0.

我该怎么找到一个短向量？我可以让CVP问题帮我找一个离0点最近的向量

So I want to find the short vector. So what I do? I simply ask, give me the close point to 0,

希望能找到这样一个短向量

hoping to find the short vector.

但问题在于我得到的也会是0向量 因为0是与0最近的向量

But the problem with that is I just get 0, because 0 is the closest point to 0.

这个结果没什么用处

And it’s not really useful,

因为我们实际上要寻找的是最短非0向量

because what’s I actually ask is the shortest nonzero vector.

这里面涉及到一个很聪明的技巧 你可以试着想想这个技巧是什么

So this is a clever trick that is used. And you can try to think about it.

你可以尝试修改归约算法

And you can show that you can modify this reduction,

比如像CVP问题问询一个不是格点的点

such as the point you ask, somehow is not in the lattice.

或者取一个子格 问点与子格点距离比较近的点

So you take a sub-lattice, and then ask for a point meeting from this lattice.

这需要一些证明技巧 这个证明很漂亮 你可以思考一下如何证明 这很重要

And it requires some sort of things, and it’s a nice proof. OK, so you can try to think about it. It’s all really important.

最后我给大家讲解一些直观结论 即这些问题互相之间都有一定的联系

For the rest, just to give you some flavor. And there are lots of connections between all these problems.

这些问题有强烈的互相关性 虽然学者们还没有证明其中的一些相互关系

So these problems are strongly interrelated, even though a lot of things isn’t understood.

第一个与之相关的问题是定距离编码问题 或称BDD问题

One of the problems is what’s known as the Bound Distance Decoding or BDD.

这就是定距离编码问题

So it is the Bound Distance Decoding.

这个问题与CVP问题类似 即与最近向量问题类似

It’s like the CVP, like the Closest Vector Problem.

我现在告诉你 点v是一个与格点很近的点了

But now I tell you, the point v is already pretty close to the lattice.

现在你需要精确解决这个问题

So you have to remain precise.

你已经有了一个与格点很近的点了 你需要找到这个最近的点

And you have a point v that is pretty close to the lattice, you just have to find the closest point.

这个问题在格密码学中占有重要的地位 这就是BDD问题 定距离编码问题

This problem places important role in lattice based crypto. OK, this is BDD problem, Bound Distance Decoding.

针对这些问题 我们已经得到什么结论了吗？

OK. So what’s known about all these problems?

我们已经知道了很多结论

There are a lot of things known.

当然 我不会向大家详细讲解证明过程的

I won’t be able to show you the proofs, of course.

我希望大家能够明白格的历史进程

I want you to know this actually important for you to get the right history,

以及为什么我们认为格对密码学来说非常重要

the right bigger on why we think these problems are good for crypto.

你可能觉得 这些问题不应该这么难吧？

You might think, maybe there’s no reason the things are hard.

但实际上 学者们已经深入研究了这些问题

But, actually, there have been lots of work on these problems.

从算法角度我们得到了哪些结论呢？

What do we know in terms of algorithms?

学者们一直在尝试提出算法来解决格困难问题

So there are lots of attempts to find algorithms to solve lattice problems.

我们学到的第一个算法1982年提出的LLL算法

The first one we mention is the LLL algorithm from 1982.

我最后会告诉大家这个算法是怎么回事

Now I can finally tell you what the algorithm does.

这个算法是一个多项式复杂度算法 是一个高效的算法

The algorithm is a polynomial time algorithm, which is an efficient algorithm.

你可以运行一下这个算法 现在已经有很多现成的软件包支持这个算法了

Actually you can run it… lots of software packages just for running it.

这个算法可以在高维空间中运行 比如几百维空间中运行 运行起来也很容易

It tries very well even in high dimensions, like several hundred, it can easily do it.

这个算法能输出什么呢？

And what the algorithm produces?

能输出一个短向量 问题在于 输出的短向量只是比较短

It produces short vector. The only problem is that it produces very mildly short vector.

更准确的说 近似因子接近为2^n 所得到的向量比最短向量要大指数级的2^n倍

More precisely, the approximation factor gets essentially 2^n. You get the vector that’s essentially longer by factor of 2^n than the shortest vector, because it’s exponentially longer than the shortest vector.

即便这样 算法构造也不那么直观

It’s still highly non-trivial,

特别是当n很小时 如果我们在低维度环境下运行这个算法

especially if you think n has been small, if you try to do it in small dimensions,

这个算法实际上一般也在低维度下使用

which is often leaded in practice for many applications,

在密码学攻击算法中 所涉及到的维度n都比较小

for many cryptographic attacks, all you need is small n’s.

当n比较小时 你可以用这个算法做很神奇的事情

If you tends, and you can still do magic with that.

但是在高维空间下 在密码学中使用这个算法时

But in high dimensions which is about to use in crypto,

比如维度等于500时 很重要的一个结论是

so you like 500, this starts to be very important,

这个算法输出的结果非常差 会输出一个非常长的向量

and the algorithm produces very bad results, produces very long vectors.

2^n是一个指数近似因子 是一个很糟糕的近似因子

2^n, the exponential is a very bad approximation.

这是Lenstra, Lenstra, Lovasz提出的算法

So this is work of Lenstra, Lenstra, Lovasz,

Schnorr, Ajtai, Kumar, Sivakumar提高了算法的性能

which is improved by Schnorr, Ajtai, Kumar, Sivakumar.

最近学者也提出了另一个工作路线 这个路线很有趣

There is another line of work, more recent, very interesting one,

这个路线从另一个角度破解密码学方案 这个路线是找到一个算法

trying to attack from different angle, trying to find exact algorithms,

这个算法确实可以解决最短向量问题 但学者们要做的工作是提高算法的性能

trying to really solve Shortest Vector Problem, but doing it as efficient as possible.

这个工作很有意思 算法的复杂度不再是多项式级的了

This is an interesting kind of work.It’s no longer in polynomial time, of course.

我们已知的最好的算法复杂度是2^O(n) 需要指数级的运算时间

The best we know how to do is in time that essentially 2^O(n), exponential time.

这个领域还有很多有趣的工作

And there exist a lot of nice works in this area.

这个工作是2002年由Ajtai、Kuma、Sivakumar开创的

It started with one of the main results of Ajtai, Kuma, Sivakumar in 2002.

最近Micciancio、Voulgaris也得到了一些这方面的结论

And even more recently, paper by Micciancio, Voulgaris,

Gama和Nguyen这几年也提出了新的结论

and more recent work of Gama and Nguyen.

我们已经得到了很多有关解决格困难问题的结论 里面有很多有趣的想法

There’re lots of nice results that show interesting ideas of how to solve lattice problems.

有些想法是用体积比较大的几何体来证明 正如前面我讲到的一些定理证明思想

Lots of interesting ideas using volume noise of the thing I mentioned before,

比如用一个椭球体 很多很有趣的想法

using an ellipsoid recovery, so lots of nice ideas.

但是现在学者们还没找到算法 能在小于2^n复杂度下解决格困难问题

But somehow none of them is able to go below time 2^n.

似乎很难把算法复杂度优化到2^n以下

It seems very very difficult to go below time 2^n,

这让密码学家们相信 与格相关的困难问题确实很难

which gives us cryptographers hope that these problems are really hard.

正如我前面提到的一样 与整数分解问题相比 格困难问题可能更好一些

Especially I can… maybe it’s nice I should have said before, this is nice to contrast with things like factoring

你会看到 针对整数分解问题我们已经得到了一些比较好的算法

where if you see factoring had important algorithmic improvements.

刚开始的整数分解算法很慢 需要指数时间复杂度

Factoring started with slow, with exponential time algorithms,

现在学者们得到了次指数复杂度等更快速的整数分解算法

and then came quadric fields, and number fields, and…

这些算法的效率非常高 都是次指数复杂度算法

These algorithms are amazingly efficient, sub-exponential time algorithms.

在人们提出整数分解问题时 人们都没想到能找到这么高效的算法

They do things that were considered impossible to be for their invention.

与整数分解相比 再看看近30年来格困难问题的发展情况

Contrary to that, if you consider the last 30 years and lattices,

学者们也提出了很多好的想法 提出了很多漂亮的算法

there have been a lot of ideas, lots of beautiful algorithms.

但是如果从性能提升的角度思考的话 并没有什么过多的进展

But if you actually consider the improvements we get, it isn’t much happening.

我们仍然需要2^n时间来解决这些困难问题

We still need time 2^n to solve these problems.

这是一个积极的信号 告诉学者们这些问题确实很困难

And this is maybe encouraging sign that these problems are maybe really hard,

可能如整数分解问题一样 我们还可以提出更高效的格问题算法

whereas factoring seems like there’re maybe more to do there.

至少从数论角度上来说 人们已经提出了很多令人惊讶的算法

But at least the number theorem is an amazing algorithm with them,

性能应该还可以再提高的

But I don’t see any reason why that would be the last word.

性能总是还可以再提高的

OK, it seems to be strange for that to be the last word.

是的 是的 我没说现在有这样的算法 但有没有算法 和能不能构造算法还是有区别的

Yes, yes, I didn’t say. But it’s a way to interpret between the two.

我认为学者们未来可以构造一个复杂度为2的√n次方的算法

And I think you can do something like set approximation 2 to the square root of n

在2的√n次方时间内能解决格困难问题

in time 2 to the square root of n, yeah.

我们不能只关注二十世纪八十年代的结论

It seems we cannot be just see the 80s.

另一个比较好的特性是 同样与整数分解问题相比 比较好的特性是

Another nice thing about this, again contrary to thing like factoring is

学者们也没找到量子算法

there is no quantum algorithm known advantage.

如果你从量子算法的角度思考这个问题

If you look in what’s going on to the quantum algorithms,

你会发现即使从量子算法的角度考虑 学者们也没提出比较好的算法

there is really not much in different nothing if you improve this algorithms.

所以可能格困难问题确实很困难

So it seems that the problem is really hard.

我提到了 是的

So, I meant, right.

为什么我们要关注归约算法呢？

What should we care about the reduction?

我的意思是归约算法对于密码学来说不太重要

But, what I meant, I should say that this is trivial for crypto.

我应该告诉你这一点的 谢谢

I should have said to you that, thanks.

应该说即使是多项式近似因子也是2^n

I should have said the polynomial, even the polynomial approximation is 2^n.

没有什么提高

It hasn’t been improvement there.

我们现在只关心近似因子

This is actually the kind of approximation facts we cared about here.

精确近似因子值的话 我不认为是2^n… 我们后面考虑下这个问题

But even for the exact, I don’t think the 2^n… So, we can think about it later.

我不太确定

I’m not sure.

这是从哪里得到的答案？

Answers from the?

是的 不过是不是线性算法？我不认为是线性算法

Yeah, but this is, is there linear, in linear. I don’t think.

我们一般用多项式算法

Usually use for polynomial.

我很希望看到学者们能提出一些性能比较好的算法

OK, if increasing so I’m happy to say that.

这里有一些多项式复杂度归约

I think there is some polynomial reduction usually.

如果能给我性能提高的启发性思想我也会很高兴的 你有些启发性的思想？

No, I would be very happy if there are heuristic, give me any heuristic?.

不用告诉我严格的证明过程…

Yeah, I don’t need anything any proofs just give me….

不是常数值？我不这么认为 好的 这是对的

Not even a constant. I don’t think, ok, that’s true.

我们可以对LLL算法做一些修改

There are some modifications of LLL,

LLL算法的改进算法就是这个算法 这是学者们能得到的最大启发了

modifications so far became that algorithms, that are most heuristic.

但是性能只在常数部分有一些提升

But I think that after the same word part…

这个算法只是给我们了一些启发 是的

I won’t say there is, give, heuristic is all we need, right.

我不认为这些启发性的思路真的可行 是的

I don’t think there is any heuristic really seems all perform, yes.

它们的基本思想都是一样的

Yeah, I think it’s pretty much the same.

这个数轴上有两个区域

OK, there is kind of one region. The region.

我们一直在讨论的是近似因子为指数的右边这个区域

Now we discuss as far I can said when the approximation is exponential,

当近似因子大于2^n时的这个区域

when dealing with the approximation is bigger than 2^n.

另一边是NP困难区域

The other end is the NP-hardness region.

我们认为这些问题是NP困难的

Here we have NP-hardness results,

这里涉及到的近似因子很小

but this is only for very small approximation factors.

密码学上这边区域的假设没什么太大的用途

This doesn’t really matter for crypto as much.

我希望大家了解到的是

But this one I want you to know is that

如果γ非常小 比如小于n的一系列log n次方什么的

if γ is very small, like say, smaller than n to the series of logs again,

我们得到的是NP困难问题

And we have NP-hardness results.

这些就是GapCVP问题了 Villas Boas在二十世纪八十年代早期提出了这个问题

And this was done for GapCVP, actually all the… Villas Boas in early 80s did that,

Dinur、Kindler、Raz和Safra得到了有关这个问题的更强结论

and more recently stronger results Dinur, Kindler, Raz and Safra,

对于SVP问题 也得到了类似的结论

and also for SVP, has been some work.

这些结论表明 这些问题的困难度非常高

Both results show very strong hardness.

这里我们可以看到一个小区域

But still there we can see very low region.

这里近似因子小于多项式级 但是比常数近似因子好一些

It’s less than any polynomial, this is very comparative to that is a very small number.

这些问题是NP困难的 但是密码学上我们不过多关注这个区域内的困难问题

So the problem is NP-hard, but it’s not in the region that we care about for crypto.

密码学上关注的区域是…

And the region we care about for crypto…

Vadim会在下面的讲座中讲到

And this will be Vadim’s next talk.

这个区域差不多是从这个近似因子开始的

The region is started form the approximation factors roughly

近似因子等于n或者大于n

I would say n and n above.

密码学关注近似因子γ落在这个区域的困难问题

When the approximation factor γ is essentially that in the domain or above,

比如近似因子等于n 或者等于n的1.5次方

so it is like the n, or something that n to the 1.5.

基于这个区域的困难问题 Ajtai和其他一些人基于格问题构造了单向函数

And there, things to the work of Ajtai and others, we have one-way functions based on lattice problems.

我们也构造出了公钥密码学方案

And we have public key cryptosystems.

在冬令营后面的讲座中 大家可以学习到这些方案

This will be the rest of the winter school, so you’ll get to see that.

我想说的是密码学所依赖的困难问题 近似因子落在n到n^2之间

But I should say all these results are in the area of n and n squared,

也就是落在多项式近似因子上

so polynomial approximation factors.

近似因子是n

The approximation factor is n.

我们并不适用NP困难的区域

It’s not in the NP-hardness region.

学者们也在另一边得到了很棒的结论

And in another results, which is very nice,

我们称其为不可近似受限性

is something one can call limits on inapproximability.

这方面的结论称 这些困难问题不像是NP困难问题

The results show that the problem is unlikely to be the NP-hard.

它们指出 当近似因子落在√n的右边时 这些问题是NP和coNP之间困难的

It shows that as you go above the square root of n, as you go the right of square root of n, the problem is in NP interset coNP.

因为这些困难问题落在NP和coNP之间 所以他们应该不是NP困难问题

And because of it in that class, it’s unlikely to be, hardly like to be the NP-hard.

这实际上告诉我们 我们所设计加密系统的安全性是基于困难问题的

So what this tells us is that we do encrypt based on the hard problem,

但是我们不能期望基于NP困难问题来构建密码学方案

but we should not expect to do the crypto based on the NP-hard problem,

因为NP困难问题有很多限制条件

cause it’s limited on much one can hope.

这里我稍微讲一下为什么我们相信这些问题是困难的

And let me tell also the reasons why we would believe it’s hard.

基于NP困难问题构建密码学方案应该很难做到

It should be difficult for the possible to do crypto based on the NP-hard problems.

但这已经超出我们讲座的内容了

But this is beyond the topic of this talk.

但我们可以证明的是

But this is specifically here you can really show that

假定将计算复杂度理论应用于困难假设时

these problems are not NP-hard,

我们发现这些问题不是NP困难问题

assuming reasonable complexity theories to the assumptions,

因为我们可以证明这些问题是NP和coNP之间困难的

because you can show this is in NP interset coNP.

我们最后总结一下接下来几天我们要讲的内容

OK, so let’s show summaries of what we need to do for the rest of the day and the next few days

我们关心的问题是近似最短向量问题 SVP或者SIVP问题

is that what we care about is the problem of approximating shortest vector problems SVP or SIVP,

其中近似因子为n或者n^(1/3) 或者n^2

to within the polynomial factors like n or n to the 1/3 or in square.

我们认为这些问题是困难的

And these problems are believed to be hard.

已知可以解决这些问题的最好算法需要指数级时间 2^n

The best known algorithms are required exponential time, 2^n.

时间与维度成2次指数关系

This is a stand for 2 to the dimensions before.

不仅是解决原始困难问题

So not just for exact algorithm,

对于近似因子为n或者n^2的困难问题 最好的算法也就是这样的

but also for the approximation factors of n or n squared,

我们没有找到更好的算法

nothing better is known.

并且我们没有找到量子算法

And there is no better quantum algorithm.

所以这些问题很困难 即使在量子算法条件下也很困难

So it’s hard, seems to be equally hard also for quantum algorithms.

另一方面 我们认为其中一些问题是NP困难的

On the other hand, some believe to be NP-hard,

这些问题似乎没有什么好的应用价值

but still this doesn’t necessarily mean bad use.

我们只是认为这些问题非常困难

So the problem is believed to be very hard.

我们基于这些问题构造密码学方案

And this is what we do crypto based on.

现在轮到Vadim上台了 我们明天应该还能再给大家做一个讲座

So I think it should let Vadim take the stage and see you again tomorrow I guess.

非常感谢

Thanks!