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

早上好 感谢主办方的邀请

Good morning. Thank the organizers for inviting me.

感谢主办方为此次活动提供的大力支持

And thanks for all the effort to put into this event.

我估计随着讲座的进行 大家可能会更感谢演讲的老师们

I think you should keep the thanks to speakers after you see the talks.

总之 非常感谢

But thanks anyway.

我们今天早上以什么为开场呢

So what I’d like to do to start the morning?

我将从学术的角度为大家讲一讲这一领域的历史进程

I’m gonna actively campus tell you a bit about the history of the area.

有关格的简短介绍

It’s a bit of introduction to lattices.

后面大家可以看到 我们的讲座速度非常快

And as you’ll see later today, we’ll catch up some more speed.

而且讲座的内容比较深入

And we’re here more in-depth talks about this.

所以 我认为第一个讲座应该是以“简介”为主题

So, and I guess since this first talk is supposed to be introductory,

希望大家都能够理解格是什么 如果有问题 大家随时可以打断我

I want everyone to understand and feel free to interrupt me.

好像很多朋友是从很远的地方过来听讲座的

And I just realize there are some people here

有来自美国的 来自加拿大的

from America, from the U.S. or Canada.

所以如果你觉得要睡着了 就告诉同桌把你叫起来

So if you feel you fall asleep ask your neighbor to wake you up…

不过我会试着讲得有意思些 这样大家都不会睡着

But I’m trying to make sure that you are all awake.

但是从这些时区来的朋友们 让大家不睡着可能有点困难

But most are coming from those time zones. So… I might have difficulty to…

这个冬令营的主题是什么？

So what’s the topic of this winter school?

主题是格和格在密码学中的应用

The topic is lattices and how to do crypto with them.

我们先来讲一讲格是什么

So let me start this talk with talking about lattices.

我想大家要首先了解的是 格是什么？

And I guess the first thing is, I should say is, what are lattices?

可能有些朋友根本没有听说过这个名词

Maybe some of you have not heard of it before.

这里有一个例子 这是在某个区域里有规律的图形

So here, an example, at least in area it’s the same thing.

这就是格 这其实是卷心菜 不过表示的是同一个东西

This is a lattice or the lettuce but in this result it’s almost the same thing.

它们都按照某种模式而规律排布

It’s actually, well, they are arranged in certain a pattern.

这种规律排布卷心菜的方式就是格

And the pattern of lettuces is a lattice.

这个例子也一样

So here it’s kind of the same thing.

所以我们有两种表示方法了 卷心菜 或者格

But I guess a broad two different notions for lettuce and lattice.

这更像是格了

And this is more like a lattice.

什么是格？

So what is a lattice?

格是高维空间中的点集

A lattice is a set of points in high dimensional space typically.

这是一个3维空间

This is 3-dimensional space.

且点的排布呈周期性规律

And the arranging is periodic-like manners.

这是3维空间格的一个例子

So this is an example of a 3-dimensional lattice.

但这实际上不能称之为格 因为这里点的数量并不是无穷多的

But it’s not quite a lattice because it doesn’t go all the way to infinity.

不过这个图能把格的意思表达出来

But I guess it gives the picture.

我这里不会详细讲解格的定义

So I’m not going to go into too many definitions right now.

我们会在讲座的第二个小时详细讲解 这里只是给大家一个概念

We’ll have that in the second hour but just make what I’m talking about.

格可以形式化地定义为这样的形式

More formally a lattice is the forward thing.

格是按照这种方式定义的点集

A lattice is a set of points that’s defined in the following way.

我们在R^n中选择n个线性无关的向量

So we take n linearly independent vectors in R^n.

然后我们选取这些向量的全部整数组合而构成的点集

And then we take all the integer combinations of those points.

幻灯片上是一个格的例子

So for instance, here’s the example.

我选择了v\_1和v\_2 这是两个向量 然后我选取它们的全部整数组合

I took v\_1 and v\_2. These are two vectors and then I take all the integer combinations.

举例来说 我们有2v\_2, v\_1+v\_2, 2v\_1

So for instance, you have 2v\_2, v\_1+v\_2, 2v\_1.

我们也可以选择负数 比如-v\_2 这个点没在图里面

And those there are also negatives like –v\_2. I don’t show you it in the picture.

但实际上 点集会遍布整个空间 遍布所有方向

But this kind of goes all over the place, goes in all directions.

很显然 (0, 0)这个0点永远包含在格点中

This of course, (0, 0) is always, always is in the lattice, the 0 point.

我想大家可以从图中对格有个直观感觉了

Any kind of you… I guess you get the pictures.

这些都是这两个向量的整数组合

So these all are integer combinations of all these two vectors.

所形成的点集有点像网格 我们把这称为格

And the form is kind of a grid. This is what we call a lattice.

有些人会把这两个点称为基

And the set of points, some people call them basis.

点集v\_1, v\_2, …, v\_n是L的基

The set of points v\_1, v\_2, …, v\_n is the basis of L.

这个定义很像向量的线性生成空间 只不过我们这里只选取整数组合

It’s a lot like the linear span of the vectors, except you take integer combinations.

我们通过这种方式得到了规律点集的形式化描述

This is what gives the descriptive structure.

在第一个小时我不会讲的特别深入

So I’m not going to talk too much about that in this first hour.

我们来讲一讲为什么格在密码学中有如此重要的地位

Let me just say one thing which is kind of the key to why a lattice is so important in crypto.

举例来说 我们选择这两个向量 这样更形象些

This is the fact that, take, for instance, these two vectors are in charge of demonstrate.

我们选择这两个向量 v\_1和v\_2

Let’s take these two vectors v\_1 and v\_2,

然后想一想这两个向量生成的格是什么

and try to think what kind of the lattices do we generate.

一眼看上去 你可能会觉得

If you think one more second you would say

v\_1 + v\_2是一个特别长的向量 和原点离得很远 所生成格的形状可能很奇怪

v\_1 + v\_2 is gonna be a very long vector that will go far away. The lattice has some kind of strange shape.

但实际上如果你仔细思考一下的话 就会发现这才是所生成的格

But actually if you try to look at it more carefully, you’ll see so this is actually the lattice they generate.

对比开始时想象的结果 你可能会觉得很惊奇 所生成的格点可以离原点这么近

And maybe the first thing, first time you see it, that might be surprising how come that vector series so close to the origin,

但实际上向量是可以相减的

but then the fact is that you have cancellations, right?

你可以计算3v\_2-4v\_1 你就会得到这个和原点距离非常近的点

You could have 3v\_2 and then minus 4v\_1, and you get back to the point here close to the origin.

这就是格之所以如此有用的一个原因 因为即使基向量很长…

So this is kind of the key why lattices are so useful because even though the basis might be long, basis vectors are long…

我想给大家讲解的是 格的基向量并不是唯一的

Now I’ll show the vector to the lattice in particular.

格基并不是唯一的 这是格的另一个性质

The basis is not unique. It’s another feature.

v\_1’和v\_2’这两个向量也是同样一个格的基

These two vectors v\_1’ and v\_2’ are also the basis of the same lattice.

这在密码学中有很重要的作用

And this is kind of the key in crypto

大家在后面也能看到 我们可以试着用格基隐藏格原本的结构特性

because we’re trying to, as you’ll see later today, we’re trying to hide the structure of this lattice, the dramatic structure of the lattice.

我可以不给你v\_1’和v\_2’ 我可以给你v\_1和v\_2

So it’s instead of giving you v\_1’ and v\_2’, I’ll give you v\_1 and v\_2.

来看看格的历史

Ok. So, a bit of history.

这里是研究格的数学家先驱们

Here are some of the mathematicians who worked into this.

照片上他们看起来好像很不开心 我也不知道为什么

They look a bit unhappy. I don’t know why.

最近数学家的照片看起来会开心些 可能是因为格理论有了些进展

Recent mathematicians are happier. There is nothing wrong with the area.

历史上 在19世纪早期 人们主要是从数论的角度、从数学角度研究格

Literally in the early 19th century, people started looking into those lattices mainly from number theory. I mean that math was mainly about that.

人们关注格的数论方面性质 而不是格的应用

And people cared about number theory, not the applications.

1801年高斯开始研究格 然后是Hermit、Minkowski

This started from Gauss in 1801, Hermite, Minkowski.

我认为Minkowski在格的研究方面做出了突出的贡献

I think Minkowski is the one who really made the major progress in this area,

甚至今天大家都可以学习到Minkowski的研究成果了 也就是Minkowski定理

that’s you will see some of it even today, even Minkowski’s theory.

他们当时关注的是不同的问题 并没有关注密码学应用

They cared about slightly different questions and didn’t care about crypto at that time.

但这个定理直到现在都有重要的应用价值

But their ideas are still used today.

所以Gauss、Hermite和Minkowski

So Gauss, Hermite and Minkowski,

他们是格领域主要的数学家们 我们今天会看到他们的一些成果

many are the main mathematicians you might see later today.

近期 在29世纪 我们会看到新一代的数学家们给出了新的研究成果

So more recently, this is in 19th century. I’m gonna tell you about like newer stuff.

近期学者们提出了一个非常重要的研究成果 我认为每个人都应该学习这个成果

A more recent, very important result is something that I guess everyone should know,

这个成果叫LLL算法 3L算法 或者叫L立方算法

it’s something called the LLL algorithm, the triple L algorithm, a cubic L algorithm

是Lenstral、Lenstral和Louvasz提出的

by Lenstral, Lenstral, and Louvasz.

这是他们三个的照片 他们三个看起来开心点了

And here is a picture, see I promised they’ll be happier.

不知道大家看不看得出来 照片里好像挺冷的 但是他们还是挺开心

I’m not sure if you can see that. They are cold, but they are happier.

我也不知道他们谁是谁

This is… I’m not sure if I can tell which one is which,

但这确实是1982年的照片 照片中的三个人分别是两个Lenstras和Louvasz

but this is the two Lenstras and Louvasz in 1982,

他们在一起研究这个算法时拍摄的

when they were working on this algorithm.

这个算法的结果非常令人惊异

And the algorithm is something pretty amazing.

如果你以前从来都不知道这个算法也没关系 明天每个人都会学习到这个算法

So if you’ve never seen it before, it’s something that everyone should know it tomorrow.

Vadim会为大家讲解这个算法

Vadim will tell us more about this.

这个算法非常令人惊异

It can do many amazing things.

这个算法最开始用于寻找格中的近似最短向量 当然我们现在也会这么用

Some of the applications or at least the original formulation of the original application is… this is the way we usually use it, this to what’s known as approximate the shortest vector in the lattice.

我们后面会看到近似最短向量是什么意思

We’ll see later what that means.

本质上说 这个算法可以用来寻找离原点比较近的一个格向量

Essentially the algorithm is able to find vectors of that close to origin.

但实际上 这个算法最初被用于 在实数域分解多项式

But actually, the original application was for something known as factoring polynomials over the rational numbers.

以及在固定维度下解决整数规划问题 这是Lenstra在后面的论文中提出的

And for solving integer programs in a fixed dimension, was a later paper by Lenstra.

这个算法在整数分解问题上具有重要应用价值 什么叫在实数域分解多项式呢？

These all amazing things of factoring, what the factors of polynomials over the rationals

就是把一个多项式分解为低阶的多项式 并且是在实数上进行分解

so you put polynomials factoring to lower the degree of polynomials, in cases about the below the rationals.

另一个应用领域是下面这个

Another cool application is this.

我们现在也可以在类似Maple的数学工具中做这样的运算

This is something you can do in… I guess in mathematical Maple?

假设我们计算得到了一个数

Assume we have some calculation,

经过了很长时间的计算 你最后得到了一个数：6.73205…

some very long calculation, you ended up with this number 6.73205…

你估计会觉得这可能是个很特殊的数

You kind of wondering maybe it’s a nice number,

可能是某个数的平方根 或者是某个数的立方根

maybe square root of something, cube root of something.

大家觉得呢？这个数是不是看起来挺熟悉的？

Any idea? Does it look familiar?

实际上这是个非常简单的例子 大家可能知道结果是什么

Actually it’s a very easy example. Something you might know.

这个数等于√3+5

So this is square root 3 plus 5 turns out.

这是个非常简单的例子

This is a very simple example.

我们可以使用LLL算法来实现这样的功能

This is something you can easily do through using the LLL algorithm.

实际上我们可以在任意数上做这样的操作

But you can have any algebra number.

你只需要输入一个数 算法就会告诉你它可能是哪些数运算得来的

And you just type a number. And actually this happened to me.

上周我就一直在做计算 并最后得到了一个非常复杂的数

Last week, I was running some in… I compute some integer, and it’s very complicated.

但后来我发现这个数实际上等于√5+4 结果挺让人开心的

Then I was happy to find out it’s actually square root 5 plus 4.

这就是LLL算法的另一个重要应用

So this isn’t any worry. This is a very nice application of the LLL algorithm.

大家可能想知道这是怎么做到的 不过我们现在要看看其他的内容了

You may want to see it, but let us see other things.

继续往下

OK.

这是一些历史知识 这是1982年的事情

So this is a bit about history. It's still 1980s, in 1982.

最近 学者们意识到格在密码学中有重要的应用价值 这也是我们冬令营的主题

More recently, this is the focus of this winter school. People realized that lattice can also be used for crypto.

当开始讨论密码学时 大家时差综合症就要犯了 因为有点理论…

So let me just say if you think about crypto some case, you have completely in jetlag you have no idea whatsoever….

什么是密码学？密码学是一个很大的领域

So what is crypto? Crypto is actually a big area.

我认为密码学对于电子商务很重要…

I think it’s important for economy, important for…

密码学被用在生活中的方方面面 比如信用卡、护照、手机、互联网等等

It’s everywhere, in credit cards, passports, mobile phones, Internet.

密码学的重要程度不用我过多强调了吧

You know that. I know you knew that.

绝大多数系统都是基于RSA密码学系统的

And most systems these days are based on the RSA cryptosystem.

这是Rivest、Shamir、Adleman在1977年提出的 他们发明了这个密码学系统These are Rivest, Shamir, Adleman in 1977, who invented this cryptosystem.

我们一直还在使用这样的密码学系统

And this is the one we essentially all use today.

基于格的密码学系统从某方面讲 将成为密码学系统的另一种选择

I guess part of the goal of lattice-based crypto is now for alternative.

我们将在接下来的4天时间里让大家了解到这一点

And this is what we are trying to convince you in the next four days.

格密码学有成为另一种选择的潜力 也是一个好的备选方案

There is a reason to refer alternative. This is also a good alternative.

我们将在接下来的4天时间里为大家讲解这一观点 希望大家能认同这一观点

So this is something we’ll see in next few days and hope we eventually convince you.

很显然 有很多理由支持我们转入基于格的密码学系统

Apparently, there are many other reasons to look at lattice-based crypto.

这就是RSA密码学系统 我后面还会提到

So this is RSA cryptosystem which I will mention again later.

如果我没记错的话 密码学和格第一次牵手 是在LLL算法提出没多久以后

And I guess the first connections between crypto and lattices came shortly after the LLL algorithm.

人们想 我们现在有密码学算法了 也有LLL算法了 我们可以做一些很棒的事情

So people said, OK, we have this crypto algorithm and the LLL algorithm, it can do wonderful things,

没准我们可以用LLL算法破解密码学方案 或者用于对密码学方案进行分析

maybe it can also be used to break… also to suggestions for cryptosystems.

实际上 即使现在 密码分析学也是LLL算法的一个重要应用领域

I mean, this is still these days one of the most important applications of the LLL algorithm.

它可以作为一个密码分析学算法 用于破解密码学方案 给出方案应用方法的建议

So it can be a cryptanalysis algorithm. It can break, also to suggest proposed cryptosystems.

举例来说 LLL算法可以破解基于背包问题的密码学系统

For instance, certain Knapsack-based cryptosystems

这是Lagarias和Odlyzko在1985年提出的

worked by Lagarias and Odlyzko in 1985.

最近 LLL算法可以破解RSA的变种方案

And more recently, variants of RSA.

Blömer做出了杰出的工作 还有Hastad和Coppersmith

There is Blömer working on these days, so Hastad and Coppersmith.

如果你在特定的配置环境下使用RSA算法 则可证明应用LLL算法可以破解RSA

If you run RSA in certain strange of configurations, it turns out you can break RSA using the LLL algorithm.

这是一个非常棒的研究成果

This is a quite nice work.

这个领域也有很多分支

It’s a lot of fork in this area.

我今天并不会过多讲解这方面的内容

This is not actually what I’ll tell you about today.

估计在后面Vadim会讲解 这的确是一个很棒的成果

I think you will see Vaidm will say more about this. it’s very nice.

在冬令营中我要讲解的内容更正能量一些

I will tell you about here in this winter school is actually positively applications.

我们要讲解如何构造密码学方案 如何应用格来构造密码学方案

We’ll tell you about how to create crypto. How to do crypto using LLL… using lattices, not using LLL.

学者们在20世纪90年代中期意识到了这一点

So this realization started in the mid 1990s.

1996年Ajtai首先进行了尝试 这个想法非常令人惊奇 是突破性的想法

This is worked by Ajtai in 1996. This is an amazing idea, a break-through idea.

这就是冬令营的主题 如何使用格代数结构 使用格计算问题构造密码学方案

This is exactly topic of all in this school. How to use this mathematical structure, this lattices, and their associated computational problems? How to use them for creating crypto?

如何构造公钥密码学方案 或者构造其他密码学函数 构造很多其它的密码学方案

How to create public key cryptosystems, another functions? Many, many, many other things.

这是1996年开创的领域 Ajtai首先进行了尝试

This idea started in 1996, Ajtai is the first one to realize that.

为什么我们关心这个领域呢？

And why do we care about that?

为什么我们要用格来构造密码学方案呢？

Why should we do crypto using lattices?

这是我在第一小时的简介中要为大家介绍的内容

This is something I try to convince you in the introduction.

事实证明 格密码学有很多整数分解或者离散对数等传统假设不具有的特性

It turns out that there are a lot of nice features that you don’t usually have using other more traditional assumptions, like factoring or discrete log.

格的安全性更高

So it has strong notions of securities.

我们现在在考虑数学证明 但是首先我们要注意

I’m thinking the mathematical proof, but first, one should be careful.

我们不能证明所有的方案都是安全的

I can’t prove that anything is secure.

无法证明基于一系列NP问题构造的密码学方案就一定是安全的

The sequence NP, or, you know, these things can all be insecure.

但我们可以证明 这些密码学方案的安全性与其他一些问题是等价的

But still, we can prove certain things. We can prove that these cryptosystems are as hard as other problems,

方案安全性与格的特定困难问题等价 而且我们相信这些困难问题确实很难certain lattice questions that we believe that are hard.

在后面几页幻灯片中我会讲到这一点

I’ll talk about that in the next few slides.

另一个很棒的特性是 格密码学方案可以抵御量子计算机的攻击

The other nice feature is that this lattice of cryptosystems are resistant to attacks by quantum computers.

现在可能绝大多数人已经知道 量子计算机可以解决整数分解问题

So by now, most of you must have heard that quantum computers can break factoring,

量子计算机可以解决离散对数问题 可以整数分解 量子计算机可以破解密码学方案

can do discrete logs, can do factoring, so it can break our schemes in particular.

量子计算机还可以做很多其他的事情

It can do many, many other things.

我想说的是 其他可以抵御量子计算机攻击的候选算法中

I would say among the only remaining candidates, among the only remaining cryptosystem on the security of quantum,

格密码学可以排在很高的地位 可能可以排在最高的地位

I would say lattices rank quite high, maybe the highest.

这也是为什么我们要研究它的一个原因

This is one of the main reasons we’re trying to investigate this.

我们现在还没有量子计算机

You know there are no quantum computers these days.

这仍然是未来才可能出现的东西 科学家们仍然在尝试构造量子计算机

It’s still a thing of the future, still under construction.

但如果你想构造一个10年内安全的密码学方案 你应该需要开始担心量子计算机了

But if you want to have crypto in 10 years, you should start to worry about this now, I think.

我们知道 我们可能在10至12年之内将得到量子计算机 所以我们需要担心这一点

You know, we know we might have quantum computer in 10-12 years. That is something you should worry about.

对于我们现在发送的一些数据 我们至少需要在10年内保证加密的安全性

You know even if not to… you know, the thing we transmit today, should remain encryption for at least 10 years,

所以我们现在就应该开始担心量子计算机的诞生了

again, something we have to start worrying about today I believe.

这很严峻 因为我们没有太多的备选密码学方案

It’s kind of serious because we don’t have so many alternatives.

如果你了解量子计算机的最新研究进展情况的话

If you look today in what would happen in science of quantum computers,

你会知道我们没有太多的备选方案

we don't have so many alternatives.

但是我们又不得不依靠密码学来实现一些功能

And it would be very hard to live without crypto.

我们需要使用密码学来在网上使用我们的信用卡 使用网上银行什么的

We can get used to this nice idea of talking our credit card in website and doing banking in your life.

所以我们需要一些备选密码学方案

So it would be nice to have an alternative.

而格密码学是抵御量子计算机攻击的备选密码学方案

And this is one of the things we’re trying to do.

我在后面会提到格密码学的另一个优点

And another advantage was something I’ll mention later.

在某些情况下 格密码学的运算速度比较快 并且格密码学可以提供更多的功能

In some cases, lattice-based crypto is even faster. And it can offer more things, offer more functionalities.

这是近几年才提出的成果 我在后面几页幻灯片中会讲到

Something realized in the last few years. I’ll mention that in the next few slides.

我们大致讲了讲为什么我们要研究格密码学

This is kind of very roughly why we care about lattices.

现在 我们尝试高屋建瓴地看看 如何使用格构造密码学方案

And here’s an attempt just to show you very high level how one does crypto?

为什么格可以用来构造密码学方案

Why is there a chance to do crypto here?

简单来说 假定我们有这样一个格

And the very rough idea is that we have this lattice.

我们一般使用高维格 不是这种格

This is going to be in high-dimensions, not this.

这是个2维格 我们一般使用更高维度的格 比如500维的格

This is in two dimensions, but going to be in high-dimensions. And you know things like dimension 500.

基本思想是 如果我们随机选择一个格点 比如这个点

The idea is that if I take a lattice point, I can take a random lattice point like this one.

然后我进行扰乱 把点移动到这里

And I would perturbative it, and move it there.

我选择了一个格点 把它往下移动了一些

So you know I took it one there and move it to be down.

现在 从计算角度很难得知这个点是否从这个格点移动来的

And now it seems very hard of computation to figure out that this point came from there.

很难得知这个点是从哪个格点移动来的

It’s hard to figure out where the point comes from.

在2维格中 这个问题看起来比较简单

In two-dimensions, it seems easy.

但如果是500维格的话 我们可以沿着很多方向移动 向上 向下…

But when there are 500-dimensions, there are lots of directions you can go, you know, up and down…

在高维中我们可以有很多移动方向

Many directions you can try to go in high dimensions.

这就是格问题困难度的来源

So this is kind of where the hardness comes from, even though,

实话实说 大家在后面的几个讲座中可能看不到这些困难问题

to be honest, you won’t see that in the next few talks,

因为现在这些困难问题都被进一步封装 使得其可以更好地应用于密码学方案中

because today, these things are going to be encapsulated in to meet the problems that you see today,

比如SIS问题 LWE问题 后面的几个讲座中大家可以学习到

So SIS, LWE problems, you’ll see that in the next few talks.

实际构造时可能用不到这个问题 不过这是格密码学构造的基本思想

But this is kind of the idea underlying lattice-based crypto. The crypto itself you see won’t involve this, but it’s the underlying idea.

我们来更深入的讲一讲 格密码学有什么优点？

So, let me go a bit more detail on what are some of the advantages of lattice-based crypto?

我前面说到 第一个优点是可证明安全

So, as I said, there is a thing called provable security.

现在 我们在构造密码学方案时 一般会伴随一个特定的安全性证明

So the constructions we have usually are associated with certain proof almost always these days.

这个安全性证明指出 如果你能破解密码学方案

And the proof says, if you break the crypto system,

如果你能破解公钥加密的私钥 或者可以对单向函数求逆

you know, if you ever to break the public key encryption’s key, or you ever to revert the one way function,

即使破解的概率非常低 几乎可以忽略 也会发生很奇特的事情

even with some small invisible probability, then something amazing happens.

如果能破解密码学方案 你也就可以解决格中的困难问题了

Typically it would be… you can solve hard lattice problems.

这非常好 这实际上告诉我们在构造密码学方案时 我们没有漏掉一些构造细节

This is very nice because it really gives us kind of, tell us that we are not, you know, missing some obvious details in the construction of the crypto system,

我们没有留下安全缺口或者脆弱性漏洞

we’re not missing some safety vitality or subtlety.

我们可以证明如果方案被破解了 就会发生奇特的事情

We know that we can prove that if you ever to break the system, then something really amazing happens.

这非常好 在传统密码学中 我们不一定能得到安全性证明

And this is very nice. I should say that usually in the standard crypto, we don’t always have that.

有些时候我们能证明安全性

Sometimes you have that.

但是即使RSA系统 我们不知道是不是破解RSA一定意味着可以解决整数分解问题

But even for RSA, we don’t know if it’s breaking RSA, it applies factoring.

有的密码学方案具备这样的特性 并不是所有的密码学方案都有这个特性

It’s true that in crypto you sometimes have that, but not always.

但在格密码学中 最近10年左右的方案构造 都可以得到安全性证明

And in lattice-based crypto, we would say it’s typically this is the case at least in last decade or so. that’s usually what you get.

这就是第一个优点了

So, this is one of the advantages.

格密码学与传统密码学的第二个对比点是 格密码学所得到的安全性更高

The second comparison, comparison here is that the type of security you get.

这一点我会在下一页幻灯片中详细讲解

This is something I’ll elaborate on the next slide.

这个优点非常好 叫做最糟糕困难安全性

This is something very nice, something called worst-case security.

这是格中所独有的优点 在其他代数结构中似乎都没有这个优点

And this is said to be something that’s quite unique to lattices. This is something you don’t usually see elsewhere.

这是个非常棒的优点 我会在下面两页幻灯片中详细讲解

This is a very nice feature while I’ll mention this in the next two slides.

这个优点的意思是如果你可以破解密码学算法

So it’s something that… it means that if you’re able to break the system,

那么你就可以解决最糟糕情况困难问题 你可以对任意格解决一个困难问题

then you can solve the worst-case problem, you can solve any instance of the lattice problem.

在下一页幻灯片中我还会提到 传统密码学是没有这个特点的

I’ll mention this in the next few slides. In standard cryptography, usually this doesn’t happen.

传统密码学的安全性一般基于平均情况困难问题

Usually you get security based on certain average-case problem.

下一页幻灯片中我会讲解这一特性的重要性

I’ll mention why this is important in next slide.

这是一个非常棒的优点

This is a very nice advantage.

这也是为什么Ajtai在20世纪90年代中期的工作得到了这么广泛的关注

This was one reason why Ajtai’s working in the mid 90’s has attracted so much attention.

这是一个新的思想 是一个新的特性 这个特性很奇特

It is really a new idea, a new notion. It’s a pretty amazing thing.

这个优点中可能包含一些我自己的主观判断 对我来说这是格密码学的一个优点

Here, I guess this is my own bios of view why this is an advantage. But for me this is an advantage because it’s based on lattice problems.

与整数分解相比 这是一个相对较新的困难问题

This is a new problem compared to factoring.

如果把数学家的工作考虑进来的话 这个困难问题已经有200年的历史了

But still it’s been around for… Actually if you take into account the work of mathematicians, it has been around for two centuries.

但在计算机科学领域 这个困难问题只有30年的历史 是个新提出的困难问题

But in the computer science it’s already been around for three decades. This is an established problem.

另一个优点是格困难问题不能被量子算法解决

And there is another advantage of not being broken by quantum algorithms.

我们对这个困难问题很有信心

So we kind of have good confidence in this problem.

这确实是一个很困难的问题 我们可以基于它构造密码学方案

It’s a really hard problem we can base crypto on it.

到现在为止 量子算法还不能解决这些困难问题

And yeah, it’s not broken by quantum algorithms.

当然我不知道是不是一定不能解决

Of course I don’t know, but this is a big open question

寻找解决格困难问题的量子算法是个很重要的公开问题

to find quantum algorithms for this lattice problem.

学者们已经寻找了15年 到现在为止还是没有能解决格困难问题

It’s been around, you know for more than 15 years, and it hasn’t been any problem solved yet.

所以可能格问题确实很困难 可能即使在量子算法下 格问题仍然很困难

So it seems like it’s really hard. And probably it’s hard for quantum algorithms as well.

我还要提到的一个优点是

And maybe one thing I should say is that

如果使用基于整数分解的密码学系统 此类方案经常要进行乘法、指数运算

if you actually look at the kind of crypto system you get, if you do system-based factoring, you always have to do some multiplications, exponentiations,

这类运算的计算开销对于笔记本来说并不大

you know, which is maybe not a big deal for a laptop,

但对于小型设备 比如智能卡 乘法和指数运算这类运算的计算开销就比较大了

but for small devices like, you know, smart cards, this does require quite a lot of effort, all these multiplications and exponentiations.

对于格密码学来说 很棒的一个特性是

One of the nice thing is if you do lattice-based crypto,

格密码学的计算开销很小 只涉及到加法运算 只是进行一系列加法运算

it is really not much going on. It’s by additions, just doing a bunch of additions.

特别是最近几年 学者们提出了很多非常高效的格密码学方案

And especially in recent years, due to some work in recent years, there are some proposals of extremely efficient lattice-based crypto.

这实在是令人惊讶 不需要做太多运算 只是一系列加法运算就能实现密码学方案

I mean it’s really amazing. You really don’t do many things, just sum bunch of numbers, to sum them up and that’s all you do.

所以即使从效率的角度考虑 格密码学也具有巨大的优势

So you know, even in the efficiency front, it seems like the lattices are doing quite well.

最后一个优点是我的同事Vadim、Chris和Craig发现的

And one last thing is that, and this is mainly due to work in recent years of my colleagues here Vadim, Chris and Craig,

我们可以用格实现其他令人惊讶的密码学功能 比如全同态加密

we realize lattice can do more things, a lot of amazing things we can do use lattices including the otherwise. For instance, the fully homomorphic encryption,

Craig会在周二和周三为大家讲解全同态加密

this is something Craig will tell you a lot about it in Tuesday and Wednesday.

这是一个开创性的工作成果 我们可以用格实现传统密码学无法实现的全同态加密

Just, you know, a really amazing breakthrough. Things you can do with lattice, which you couldn’t do before.

此方案的设想是在20世纪90年代中期提出的 没人能想到这竟然可以实现

And this is something in the mid 90’s, no one dreamed that this can be possible.

没人能想到我们可以利用格实现实现整数分解无法实现的一些密码学特性

No one dreamed that one can take lattices and do things that they couldn’t do using factoring.

以前学者们一直认为整数分解这类数论假设是最强的一类假设

You know, previous thought factoring in numbers theoretic assumptions are strongest as it gets,

但实际上格也可以实现很多功能

but turns out you can do a lot things using lattices too.

随着冬令营的进行 希望大家可以逐渐认同学习格密码学是很有价值的事情

OK, hope by now, you are convinced lattice is worth listening to the rest of this winter school.

我们来深入讲解一下这些特点

Let me tell you a bit about these kinds of issues a bit more in detail.

第一点是可证明安全性

This first figure is about provable security.

正如我前面提到的 我们不能证明某些方案是绝对安全的

As I said, we are not really proving something is secure.

没人可以证明如果P等于NP 那么所有方案都可以被破解

No one can prove that if P equals NP, everything’s break.

那可证明安全又是什么呢？

But what provable security is about?

可证明安全将密码学构造与一些我们认为可信的结论联系在了一起

It’s about connecting the cryptographic construction to something that you already know and love and trust.

可证明安全是一种归约算法 它将密码学方案的安全性归约到一个困难问题上

So, it’s a reduction. It’s a reduction from an established hard problem,

这个困难问题不是一大早你拍脑袋想出来的 而是一个新的、已确定困难的问题

and not a problem you just came up this morning, but a new and established hard problem

我们把一系列密码学函数归约到这类困难问题上 这就是安全证明

to a set, to a cryptographic function. OK, so this is security proof.

我们希望利用安全证明来讨论方案的安全性

And this is something really you want to have.

我后面可能就会提到 在很多情况下 安全性证明很有用

I’ll mention, maybe later, this is, you know, in several instances, this really helped.

它告诉我们应该如何使用一个密码学方案 我后面还会提到这一点

And it gave us kind of the hint what we should be doing. So I’ll mention this in the next few points here.

安全性证明会告诉我们不要做哪些蠢事

So it tells that we are not doing anything stupid.

它告诉我们如果可以破解密码学方案

It tells that if the cryptosystem is broken,

如果存在一个攻击者以某个小概率破解密码学方案 就会发生一些奇特的事情

if there is a attacker that can attack with even small probability, then something amazing happens,

也就是我们的困难问题不再困难了 整个假设都被破解了

namely our hard problem is no longer hard, so we know the whole assumption is broken.

不光是你构造的算法不再安全 所有依赖于这一困难问题的密码学构造都不安全了

Not only our cryptosystem is no longer secure, but, you know, also all other constructions based on that problem.

这是个非常好的结论 我们总希望能得到这样的结论

This is something, you know, very nice to have. This is something that we always want to have.

我们并不是经常能得到这样的结论 但格密码学可以得到更强的安全性结论

Not always get. But, with lattice-based crypto, as you see, essentially always get such a stronger security guarantee.

这是个非常好的特性

This is really something very nice.

安全性证明会告诉我们所构造的方案没什么问题

It kind of tells us if we are doing the right thing.

如果你有一些疯狂的想法 想要去构造密码学方案 首先你需要得到安全性证明

So if you have this crazy idea, let’s try to construct this cryptosystem. The first thing, if you have the security proof

能把所构造的方案与一个困难问题联系到一起 你就会知道你的想法是正确的

that you can connect to something you already know, connect it to a hard problem, then you know you’re going the right way.

可能你忘记了加什么东西 因为不加或者不做某个运算 方案可能不安全

It’s sort of like you might have forgotten to add, find some worries, because of that something is broken.

但是如果有安全证明的话 这种事情就不会发生

And let me mention, without the security proof, this can’t happen.

我们在周二会看到一个例子 方案并没有安全性证明 人们认为这个方案是安全的

We’ve seen some of those. We will see one example on Tuesday. You don’t have a security proof. But you think everything is secure, right?

人们构造了一个密码学方案 很长一段时间人们都认为它没什么问题

You constructed the cryptosystem, everything seems fine, you think about it for a long time.

但实际上这个方案并没有和某个已确定困难的问题联系到一起

But there’s nothing that connected to an established problem.

随后人们发现这方案有问题

Then, you know, problems can happen.

我们会在后面提到这样的例子

And we mention some of that later in this week.

另一个很棒的地方是 我后面可能会提到 我估计Chris在今天他的讲座中也会提到

Another nice thing, maybe, we might mention. I hope Chris will mention later today.

安全性证明可以告诉我们如何选择方案的参数

It can also give us hints as to choice of parameters.

这也和下面的一个特性相关

And this also relates to my next point.

你有了一个密码学方案 但里面涉及一些参数 比如某些特定的数要大于其他数

So, you have the cryptosystem, but there’re some parameters, like you know, the size of a certain number should bigger than some other number.

大多数情况下 你可能不知道如何选取这些参数

Often, you don’t know how to choose these numbers.

但如果有安全性证明的话 它就会告诉你 如果安全性证明能通过 则m>n^2

But if you have some proof security, then the proof security tells you that, for the proof security to work, you know, m should be bigger than n squared.

所以你在使用方案的时候 会把m选为大于n^2的数 这样安全性证明才能过得去

Then you say, maybe we should really take m to be bigger than n squared to make the security proof work.

密码学历史中也发生过这样的事情 Chris可能也会提到

And as one nice thing happen in history, maybe Chris will mention that.

我们证明了一个方案是安全的 安全性证明告诉我们 一部分参数要大于n^2

we had security proof, the security proof told us the several parameters have to be bigger than squared n.

我们不知道为什么 但我们会说 用的时候就让这些参数大于n^2吧

We didn’t know why, but we would said OK, let’s use these parameters bigger than squared n.

那个时候我们还不知道为什么

And we didn’t know why at that time.

但是5年以后 学者们提出了另一个算法 这个算法指出

But like five years later, came another algorithm that showed that

如果这些参数小于n^2 我们可以在次指数时间内破解密码学方案

if these parameters are smaller than squared n, you can break the system in sub- exponentiation time.

我估计Chris在今天他的讲座中也会提到这一点

I hope Chris will mention this later today.

大家可以看到安全性证明的威力了

So this kind of tells you the strength of something like security proof.

它会告诉我们怎么做才是正确的 它会告诉我们是否走在了正确的道路上

It really tells you what’s the right thing to do. It tells you that if you’re going down the right path.

我来举个例子 来看看安全性证明大概是个什么样子

So let me just mention one example of one security proof might look like.

这也会涉及到下面一个优点 也就是最糟糕情况 平均情况困难性这个优点上

And this will bring us to the next point, to the issue of worst-case, average-case hardness.

这是个很简单的例子 是一个基于模平方的单向函数

This is a very simple example for one-way function based on modular squaring.

这只是给大家解释下基本思想

Just gives you illustration of this idea.

假设你希望构造这样一个单向函数

So as you might want to construct the following one-way function,

这个单向函数是将输入值求平方 输入值为x 输出为x^2

the one-way function is that squares the sequence of numbers. It takes x and outputs x^2.

我们就用它了 令N为两个大质数的乘积

So I do it. Let N be some product of two large primes.

我不知道如何选择参数N 后面几分钟我就会提到怎么选择参数N了

I don’t know how to choose these. These will be mentioned more about in a few minutes.

现在就选择某个N 这个N是两个大质数的乘积

So choose somehow N of product of two big primes.

现在考虑这个函数 求x的平方模N

And now consider the function simply squares x and reduces modular N.

在习题中 我会让大家试着证明如果想求这个函数的逆函数

So in our next exercise, to show that being able to invert this function,

也就是给定x^2 其中x是随机选取的 找到任意一个x’ 使得x’^2=x^2

being able to find pre-images of this function on the randomly chosen x. So I give you x^2 for randomly chosen x to find the pre-image, meaning find any x‘, whose value does map to x^2.

在习题中我们可以很容易证明 如果可以找到原像 你就可以分解整数N

Then it’s easy to show in the next exercise that, if you can do that, if you can find pre-images, then you can also factor N.

这就是安全性证明的一个例子

OK. So this is one example of security proof,

它证明了如果你可以求逆函数 那么你就有更高的概率

because it shows that if you are able to invert this function, even with more probability,

可能有1%的概率 或者某个不能用代数式表示的概率下 你可以分解N

even probability of one percent or any none algebra probability, you can factor N.

如果你不知道如何证明 或者没见过这个单向函数 试着想一想

If you forget how to do that, or haven’t seen it yet, try to think about that.

我们实际上就是在证明这个单向函数的安全性

But there’s actually one point I’m trying to make here.

我们现在有了安全性证明 这很好

So we have the security proof, and this is actually quite good.

它证明了这个单向函数的安全性与整数分解相关 这是个挺不错的单向函数

It shows us that at least this is related to factoring, so this is not a bad one-way function.

但是这引入了另一个问题

But still, this brings me to the next point,

这个函数 这个安全性证明叫做平均情况困难证明 基于平均情况困难性

this function, this security proof is what I would call average-case, based on average-case hardness.

基于平均情况困难性的原因是 我们没告诉你怎么选择N

And the reason it’s based on average-case hardness is because of this N. I didn’t tell you how to choose N.

你会问我如何选择N 我会告诉你N等于两个大质数的乘积

And you should ask me how should I choose N. I told you to take N equal to be the product of two large primes.

我们还需要知道其他一些限制条件吗？

But is there anything else we should know?

大质数本身是不是要满足一些性质？是不是有所谓好的质数和不好的质数？

Should primes set by some properties? Are they good primes or bad primes?

这好像不那么显然 可能没有什么直观的结论

And this is not such obvious. There’s no obvious answer to that, I would say.

这实际上是一个很严重的问题 一个大问题

I think this is a serious, this is a big question.

这引发了下面的问题 也就是我们如何选择N

And this kind of brings me to the next story, which is how do we choose N.

我们在RSA下看看这个问题 你想用代码实现一个RSA 如何选择参数模N？

Let’s say do RSA, you want to write the code for RSA. How to choose the modulus N?

如果你上过这方面的课程 你会知道 N不能是个偶数 偶数N不是个好选择

So OK, if you took a course about what you already know, it shouldn’t be even right? because you take N to even is a bad idea.

所以你估计会选择两个大质数 然后把它们乘起来 教科书上一般就讲到这里

So you want to take the product of two primes, two large primes. And this is kind of the basic thing usually learn. But this is that all about.

但有可能某些质数比其他质数好？大家觉得呢？

Maybe there’re some primes better than others. What do you think?

你觉得什么质数算是个好质数？

Do you know what a good prime should take?

安全质数？

Safe primes?

还有什么答案吗？

Any other proposals?

这是个好问题 我们其实不知道这个问题的答案是什么

So, it’s a good question. I don’t think we really know the answer.

让我来告诉你历史上发生了什么事情

But let me tell you about what really did happen in history.

1978年 学者们开始研究整数分解算法了

In 1978, people realize, so people were working algorithms on factoring, right?

那时候RSA出现了 人们意识到整数分解是一个重要问题 试着寻找整数分解算法

RSA came out and people realized the factoring is an important problem, and try to find algorithm for factoring.

1978年 学者们发现存在一个高效的算法分解整数

In 1978, people realized there’s an efficient algorithm,

前提是p-1或者q-1的最大质因子要比较小

when the largest prime factors of p-1 or q-1 are small.

所以人们说 如果你让RSA抵御这类攻击的话

So people said, OK, if you want to prevent attacks using this kind of attack,

你需要选择p和q 使得p-1和q-1的最大质因子比较大 不能太小

you should choose p and q such that p-1 and q-1 are large, have large… the largest prime factors are large. OK, that shouldn’t be small.

于是在1978年 你在代码里面增加了一个检查机制 检查p和q是不是满足条件

So this was 1978, fine. So you add this to your code. You add this check. It’s usually OK. But anyway, you add this thing to check that p and q are fine.

接下来 1981年又出来另一篇论文 对于另一个特定情况又有了另一个算法

Good, so next thing, 1981. Another paper comes out. There is another algorithm for another special case.

p+1和q+1需要有比较大的质因子

p+1 and q+1, they should have a large prime factor.

很好 你又增加了一行代码 保证p和q也满足这样的条件

OK, good. So you add another line to your code. So make sure this also happens.

又过了1年 又出来了一篇论文

Two years, one year passed actually and that is in another paper.

如果p-1和q-1的最大质因子是p’和q’ 那么p’-1和q’-1也要有大的质因子

And now the largest prime factors of p-1 and q-1… If they are p’ and q’, then you really want p’-1 and q’-1 to also have large prime factors.

好的 你再检查下这个是否成立

OK, you checked that too.

大家估计可以猜到接下来发生什么了吧？

And you can guess what comes next.

1984年 学者们发现如果p+1和q+1的最大质因子是p’和q’

1984, the largest prime factor of p+1 and q+1 is p’ and q’,

则p’-1和q’-1也需要有比较大的质因子

then now we want those two also have… p’-1 and q’-1 to have large prime factors.

大家估计没听说过有这么个历程 原因是我们现在有了更高效的大整数分解算法

So maybe you never thought of this before. The reason you have never seen this before is because these days we have much more efficient factoring algorithms,

我们发现了数筛法 所以现在这4个要求已经没这么重要了

the number sieving was discovered. So actually today, these four lines are no longer relevant.

我们已经有了更高效的整数分解算法

We have much more efficient factoring algorithms.

对于所有N等于p乘以q的形式 上面这4个算法的效率和数筛法效率差不多

And those at least, as far as we know, those seems to perform equally well, you know, for all N of the form p times q.

可能对于不同的p’和q’ 整数分解的效率还是会有所不同

This is, as far as I know, maybe there is still some difference for certain p’s and q’s.

但是我想这不会意味着故事的结束

But you know, that’s not necessarily to be end of the story,

我不是一个数论专家 但我估计在接下来几年 学者们可能会提出更高效的算法

I mean, I’m not an expert in number theory. But, you know, in a few years, people might come up with more efficient algorithms.

我相信我们要根据所提出的算法 选择更为特殊的质数

I believe that there might still be certain choices of primes that might be more efficient algorithm.

这确实是一个问题 我们该如何选择参数N

So that is an issue here. How to choose the parameter N?

为什么会有这样一个问题？问题出在安全性证明上面

And why is there an issue here? This is because of the kind of security proof we have.

安全证明称如果能破解RSA 并不意味着能分解所有的整数

We have… breaking crypto system, breaking RSA, doesn’t mean that you can factor,

只意味着能分解某个特定的整数 但不意味着能分解所有的整数

just means that you can factor, doesn’t mean that, but the essential meaning you do something non-trivial,

只能分解一个特定的N 不是所有的N 只是一个特定的N

with very certain N, with specific N, not all of them, with very specific N.

这就是所谓的平均情况困难性

This is what we call average-case hardness.

所以如果你相信破解RSA 或者破解其他基于整数分解问题的方案是困难的

So, to believe that RSA in other system based on factoring are hard,

你相信的是在平均情况下 整数分解是困难的

what you need to believe is that factoring is hard, on the average-case.

你需要相信的并不是所有整数分解都是困难的 只相信一部分整数分解是困难的

You need to believe, not just that it’s hard to factor all the numbers, but it is hard to factor, even the small fractions of them.

我们这里假设的是 可能有1%的数是容易分解的 这1%的RSA方案是可以破解的

So it might be a hypothetic situation where there is 1% of all numbers that are easy to factor. And then break RSA because 1% of case factor,

但是在最糟糕情况下 整数分解还是困难的

that’s really hard to factor a worst-case number.

所以可能我们对某些特定的数可以很快进行整数分解 但不是全部的数

So might be some number with certain constructions we can factor, but not all numbers.

这确实是一直困扰我们的一个问题 但格里面没有这样的问题

This is exactly the kind of thing that bothers me here. And this is kind of thing that lattices don’t suffered from.

这是一个重要的核心优点 格困难问题是最糟糕情况困难问题

This is one of the main advantages. This is the worst-case hardness in lattices. Enjoy.

我们从密码学函数的角度再理解一下

And let me try to tell cryptographic function again.

如果你基于平均情况困难性进行安全性证明 你得到的是类似于这样的映射关系

If you’re doing a security proof based on average-case hardness, you look for something like that.

你选择了这个参数N 右边你就得到了一个基于N的密码学函数

So you have this parameter N, and on the right side you have the cryptographic function based on N.

这有点像双射 有点像一对一映射

It looks just like kind of bijection, just like one-to-one correspondence.

对于每个N 你都证明了存在一个相同的N所对应的密码学函数

For each N, you show that you can map it through a cryptographic function at the same parameter N,

如果密码学函数被破解了 你就能分解这个N

such as if the cryptographic function is broken, then you can factor N.

确实有点像一对一双射

But it’s really like kind of one-one bijection.

如果你突然发现密码学方案被破解了 这就意味着1%的密码学方案也被破解了

So if suddenly, the cryptographic system is broken meaning, say would, probability 1% of the cryptographic system is broken.

你得到的只是N中1%的数可以被分解

All you get is that 1% of all N, or 1% of all numbers N are broken, a factor.

不是能分解所有的数 只意味着能分解1%的数

But that doesn’t mean they can factor all of them. It only means you can factor 1% of them.

而在格中我们可以得到更好的结论

This is exactly the kind of thing we can improve using lattices.

这就是所谓的最糟糕情况困难性

This is what’s known as worst-case hardness.

什么是最糟糕情况困难性？

So what’s worst-case hardness?

安全性归约 或者说安全证明更像是一个完全映射

The security mapping, the security proof looks more like complete graph in some sense.

在左边选择一个任意的格 把它映射到右边 它会映射到右边所有的密码学函数

You take any lattice here in the left-hand side. And you map it to the right-hand side. You mapped it to cryptographic function. It’s a map to uniform cryptographic function.

这个映射关系非常令人惊讶

Its link looks pretty amazing.

下一个讲座中 Vadim将为大家讲解这是如何做到的 很神奇

In the next talk Vadim will show you how this is done, this magic.

这就是最糟糕情况困难的主要思想

This is really the main idea of worst-case hardness.

从某种程度上 你将任意一个格困难问题都映射到了密码学函数的整个空间里面

In some sense, you connect any lattice problem, any instance, I should say, any instance of the problem, any lattice you can map to a uniform distribution on this cryptographic function.

这非常令人惊奇

This is pretty amazing.

这也意味着 如果1%的密码学函数被破解了 则任意一个格问题实例都可以被解决

What it says is again, if now you have 1% of your cryptographic function is broken, meaning your probability 1% of cryptographic function is broken, then any instance of this lattice problem, any lattice can be now solved, can be broken.

从很多方面看 这都是一个很好的性质

This is a nice thing in several perspectives.

首先 这使得安全性强度更高了

First it’s a strong security guarantee, right?

不光是安全性强度更高了 而且它也告诉我们 不需要担心如何选择参数N了

But, it is not only a stronger security guarantee, but also tells you don’t have to worry any more about how to choose N.

你不需要担心如何选择所涉及到的参数 只要按照某个分布选参数就可以了

You don’t have to worry about choosing the distribution. It tells you that you should, just as a natural distribution, it just works with.

安全证明会告诉你如何选取参数 你只要这么选就好了

The security proof tells you what the distribution is. You can just work with.

不需要担心哪个是好的N 哪个是不好的N

You don’t have to sudden worrying what is the right N, what is bad N.

安全性证明会告诉你如何正确选取参数的

So it tells you that this is what you should be working with.

我这里要强调一下 这是格的简介讲座 这只是一个简单的介绍

So let me just say, this was kind of introduction. This was the introduction to the area, I should just say.

在进入技术部分的讲解前 我们要再讲一讲历史 在最近15年中又发生了什么？

One more thing before we go to more technical part, this is kind of for more recent history. What happened in the last decade and a half?

正如我前面所说的 Ajtai和Ajtai-Dwork在1996年开创性地提出了格密码学

So as I said, the big break for the seminal work of Ajtai and Ajtai-Dwork in 1996.

他们意识到我们可以做很多很棒的事情 我们可以使用格做很多很棒的事情

They realized that you can do lots of nice things. You can do amazing things using lattices.

他们告诉我们如何实现单向函数 如何实现公钥密码学系统

They showed how to do one-way functions. They showed how to do public key cryptosystem.

但是他们只提出了一个概念指出这些是可以用格来实现的

These were, I would say, mainly proof of concept. They showed that these things are achievable.

他们也指出如何得到最糟糕情况安全性

They showed how to obtain the worst-case security guarantees.

但从性能方便考虑的话 他们构造的方案效率非常低

But if you look for practical side, these things look still extremely inefficient.

如果使用他们所提出的系统 密钥可能是GB级的

I would say, to actually use the system, you would have keys of gigabytes, which is…

性能实在不太好 运算效率很低 很麻烦

It’s not so nice. It looks seemingly slow, cumbersome.

我认为他们的构造更多地是从理论的角度考虑的

And I think that should be, these things seemed like, should be mainly of theoretically interest.

但是 最近几年 格密码学有了翻天覆地的变化

It’s kind of what has changed in last few years.

现在我们已经得到了非常高效的格密码学方案

These days we really have very efficient constructions that are competitive,

其效率甚至可以与RSA比拟

competitive with things like, even things like RSA.

开始时方案看起来根本没法用 也没什么可扩展的能力 只能做公钥密码学系统

So it also seems like it is very hard to use these things, very hard to extend to do other things, just do public key crypto system,

而且绝大多数公钥密码学系统的构造都很繁琐

most basic public key crypto system seem really quite difficult.

仅获得选择明文安全性已经非常困难了 而且几乎没有任何可扩展性

Even just chosen-plaintext attack seemed quite difficult. And the extending seemed almost impossible.

随后Louis和Warick试着定义格密码学系统 延伸这个思想

And then Louis and Warick tried to identify, improve this idea and define them.

最近几年 学者们集中研究格密码学的两个核心问题

And I say, in recent years we, kind of, converge the two center of problems,

这两个核心问题也是接下来两个讲座的主要内容

which may be the topic of the next two talks.

Vadim会讲到Short Integer Solution问题

Vadim will talk about the Short Integer Solution problem.

Chris会讲到Learning With Errors问题

And then Chris will talk about the Learning With Errors problem.

用这两个问题可以构造非常高效的密码学方案 非常高效的单向函数

And these problems allow to construct very efficient constructions, very efficient cryptosystems, very efficient one-way functions.

而且这两个问题把格的一些内容抽象化了

And they also allow to abstract a way of the lattice stuff.

当你设计格密码学系统的时候 你不需要去考虑最糟糕情况困难性的安全性证明

Now when you try to do lattice-based crypto, you don’t have directly to worry with the worst-case security proof.

你只需要基于SIS、LWE这两个中间问题构造方案就可以了

You just base on one of these intermediate problems, the SIS, LWE, and take it from there.

有人已经帮你们把最困难的问题解决了 帮你们完成了最糟糕情况证明

Someone already did all the hard work for you. Someone already did the worst-case hardness proof for you.

你只需要选一个问题 用这个问题构造密码学系统就好了

You just have to take it from there and use these to construct cryptosystem.

当然这方面还有很多工作要做 但至少一部分工作已经完成了

There’s still a lot of hard work from there too. But, at least, part of the work has already been done.

这些将是后面两个讲座的主题

And this would be the topic of the next two talks.

这就是近几年格密码学的研究主线

So this is one main line of work in recent years.

同时 格密码学还有另一个研究主线 我想大家在周二的时候会学习到

In parallel, there is another line of way which we will see, I guess, on Tuesday.

Vadim会在周二的讲座中讲到

Vadim will tell us about it on Tuesday.

这个主线所构造的密码学方案非常高效

And this was, I said here, this was very efficient.

我不得不说 这类格密码学方案实在是太高效了

And here I would say, this is extremely efficient systems.

他们是基于特定格代数结构而构造的 这两个问题被称为Ring-LWE以及Ring-SIS

And they are based on certain structured lattices, so the problems we called the Ring-LWE or Ring-SIS.

2002年Micciancio开创了这一领域

And these start the work of Micciancio in 2002,

2006年Chris Peikert、Alon Rosen、Lyrubashevsky和Micciancio进行扩展

and then Chris Peikert and Alon Rosen in 2006, and Lyrubashevsky and Micciancio in 2006.

他们意识到如果不考虑一般格 只考虑循环格这种满足特定性质的格代数结构

They realized if you consider certain types of lattices, not the most general lattices, but lattices like cyclic lattices that take some structure,

就可以大幅度提高方案的效率 构造安全高效的密码学方案

then you can make everything much more efficient to get the correct improvement.

而且所构造的密码学方案确实非常高效

And these constructions really can be truly efficient.

我不会讲解太多这方面的内容 不过大家可以感受一下

I would say, keys of size, kilobytes, something that is out of my introduce,

密钥的长度是KB级的 而计算效率甚至可以和哈希函数的计算效率接近了

even the competitors in the computation for hash functions.

我想Vadim会在周二详细讲解这方面的内容

I guess Vadim will tell you more about this on Tuesday.

这就是格的简介部分

OK. So, this is the introduction.

在进入技术部分的讲解之前 大家准备好 很烧脑

Before I go to the technical part, you can power on the brain.

有什么问题吗？

Any questions?

没有问题

No question.

请问？

OK. So, go.

问题是格问题到底有多难 比如最短向量问题有多难

OK, the question was how hard are certain problems, like the shortest vector problem.

我在下面的技术部分讲到这些内容 你会看到更多这方面的内容

I’ll mention it in the next part. You’ll see more. I’ll get to the more technical slides.

我们会来看看我们在使用什么样的困难问题

And we’ll see what kind of problems we are dealing with.

如果到时候我忘了回答你刚才的问题的话 提醒我一下

But if I don’t answer your question, ask me again.