In Conversation with Jintai Ding

Nathan Thomas Carruth

Biographical Sketch. Jintai Ding is a professor at the Yau Mathematical Sciences Center at Tsinghua University and the Yanqi Lake Beijing Institute for Mathematical Sciences and Applications. He received his Bachelor's degree in computational mathematics from Xi'an Jiaotong University in 1988, his Master's degree in number theory (proving a conjecture by C. L. Siegel) from the University of Science and Technology of China (USTC) in 1990, and his PhD in quantum groups from Yale University in 1995. Following a postdoc at RIMS in Kyoto, Japan, he joined the faculty of the University of Cincinnati in 1998. Currently he works in many fields related to postquantum cryptography, an area he joined while it was in its infancy, and to which he has made groundbreaking contributions.

In this interview, he gives us a window into his experience in mid-1980s China, shares anecdotes from his time at Yale, discusses his sharp transition into postquantum cryptography in the mid-2000s (things were beginning to get "boring", he says), and talks about some of the highs and lows he has experienced in cryptographic research. Along the way, he gives us an introduction to various concepts in classical and postquantum cryptography, shares his views on the beauty of applied mathematics (to him, it is as beautiful as pure mathematics), and discusses matters of credit and priority. He also explains how his varied background has come together in surprising ways, talks about his current research, and reminds us that the quantum challenge to classical cryptography is a worldwide problem. Professor Ding's confidence is contagious ("I was convinced that, with sufficient time, I could understand anything"). He is also a very animated speaker; we hope at least some of this vitality comes through in this written record.

NTC: *I have some overview background questions, and then I have some more specific research questions and then if there is time I had some reflective questions.*

JTD: I have plenty of time if you don't mind.

NTC: *I was curious first of all what led you to study mathematics in the first place.*

JTD: Well, that's actually a very interesting question. China was a very interesting country during my time. We had a national exam – we still have it. At that time actually I wanted to study physics. Then that year I did the national university entrance exam and that year the math exam was the hardest ever in Chinese history. 1984. I got a 68. It turns out that was very good. My number one choice was actually USTC, I think, but I couldn't get in and then they just sent me to the math department in Xi'an Jiaotong University. I didn't want to go, but my parents forced me to go. In China at the time it was very difficult to change major so I just went on. I was always interested in physics; I thought I would be a physics students, but then –

NTC: *So you went into math just because of your score on the gaokao [national university entrance exam].*

JTD: I did very well in math. Remember I had a 68, but it was already very good, so they put me in the math department because I did well in math.

NTC: *Well, in Cambridge 70 is a good mark.*

JTD: That year – you can check, people still talk about it. 1984. So that's basically it.

NTC: *Then in your undergraduate degree you studied more applied –*

JTD: Computational mathematics. I was at this university called Xi'an Jiaotong University. It's in Xi'an, it's one of the Jiaotong universities – they have five of them, in Taiwan, Shanghai – it was originally in Shanghai, and then because there was a possibility of fighting a war with Russia, they sent them to the west of China. Then some people went back to Shanghai. So that's that. It still was a major engineering university in China, in Xi'an.

NTC: *Then in your Master's degree you went on and did number theory –*

JTD: Yes, then I did a Master's degree in USTC in number theory. I studied – my parents, my father was there so I studied –

NTC: *Your father was at USTC?*

JTD: Yes, my father was a chemist.

NTC: *He was a professor there?*

JTD: He was a professor there, yeah. I went to USTC. I studied number theory. I was really fed up with computational math. I thought there was not much content. I did a lot of numerical analysis, I was writing programs. I was very good at programming, but I hated it. At that time we didn't have self correction; whenever I did programming, most of the time I spent doing typo checking. Because you know we had – I mean, 1984, China –

NTC: *Right. What kind of computers did you have actually in 1984?*

JTD: I forgot. It was a mainframe, we had to fall in line – we had to wait. Sometimes we had to go to the computer at 3:00 or 4:00 AM because there was a line there. We had very limited time. I barely had time to run my program because it would just say, Error! Well, where was the error? I couldn't find the error! It turns out it was mostly typos. I had a terrible experience. So I decided to do something more theoretical. I decided to do number theory. I did very well in the exam. I prepared for the exams myself. I had studied computational math. Most of number theory I taught myself. In four years of my life in university I almost never took classes. I almost got kicked out because I didn't take classes. I did the exams; I feel I did quite well in the exams, but they didn't like me. I almost got kicked out. I have to apologize to everybody publicly, because I didn't go to class and it affected all the people in my dorm room. One funny thing is in the end I think I was the only one in my room who finished on time. Everyone else's graduation was delayed by at least one year. At that time things were very interesting. In my time, when you finished, you were given a job; you didn't look for jobs. Because I did all these bad things, I realized if I could not go to graduate school they would send me to some really terrible place.

[Laughter.]

JTD: So I had to take the exams and go to graduate school, otherwise I would have had no future. I did extremely well in the exams for USTC, so I got in to graduate school. I studied number theory for two years. I was lucky, within three months of starting I proved a small conjecture by C. L. Siegel.[1] So I really only studied for three months.

After my Master's degree I applied to graduate programs in the United States. Because I had proved this conjecture I got this prize in China called the Zhong Jiaqing Prize.[2] It's pretty prestigious in China (not internationally). So I got several offers, including Columbia and Yale. So I decided to go to Yale. So that's the story. I'm a very unconventional Chinese, I think. Very unconventional.

NTC: *I'm actually interested as well – this wasn't on my question list but I didn't know this much about your background. Do you have any insights from your experience – being in China, and then seeing other parts of the world, and so on – about how China could better support people who have a lot of talent, but who are less conventional?*

[1] See [6].

[2] This is a prize given to outstanding graduate students in China, in honor of the Chinese mathematician Jiaqing Zhong. Professor Ding was one of the earliest recipients of this prize.

JTD: Overall China still has a long ways to go in that respect. I think Asia in general.

I was in college in the 1980s. I think the 1980s was a very interesting time in China. There was a big discussion on the topic of, What is truth? They would discuss the topic in the People's Daily; what is truth, how do we know something is true. They were sorting things out after the Cultural Revolution.

NTC: *Yes, I thought it was interesting that you were studying theoretical mathematics back in the 1980s when China was still in the process of opening up.*

JTD: Yes, at that time we were talking about truth – we were talking about great ideas and so on.

NTC: *What was the overall math atmosphere like in China in –*

JTD: Oh, it was bad back then. I preferred to study by myself because I didn't think most professors understood what they were talking about (and I was right, I think). In China, even now, they emphasize techniques – for example, matrix methods, not abstract concepts of linear algebra. When we do linear algebra in China we essentially just talk about matrix theory – you know, how to make this entry 0, how to make this upper triangular. They don't talk about invariant subspaces or about bases or linear transformations; the emphasis is on the techniques. Also, the subjects are very disjoint.

China was disconnected from the world from the '50s. They learned something from the Russians – all the books we had were Russian. Only after the 1980s when China started to open up did we get new books. I think China missed the whole period of modern mathematics.

NTC: *The Sino-Soviet split didn't affect the math?*

JTD: No, not too much. They had some access to Russian math. The Russians were also the enemy of China after the 1960s so then China just did whatever they – during the Cultural Revolution there was no math anyway so it stopped for 20 years. Maybe you heard about this guy [Chen Jingrun] – but I don't know, maybe you never heard about this guy –

NTC: *Hua Loo-Keng?*

JTD: – This guy – no, Hua Loo-Keng was pretty good. He was a king – he dictated the math world in the whole of China: what math should be done, and what math should not be done.

There was one guy [Chen Jingrun] who proved the best result about – you know the Goldbach Conjecture, right? Every even number can be written as a sum of two prime numbers. I think a German mathematician started from 9 times 9 (which means every even number can be written as a product of 9 prime numbers plus a product of 9 prime numbers); then they went down to 8 times 9, and so on. This guy proved 1 plus 2.[3] So it was a big deal. But in the U.S. people never heard of him because he was basically using sieve methods, and it just amounted to increasing a Taylor expansion from 50 to 100 terms. Idea-wise, there was nothing new.

[3] See [4].

One day at Yale I was talking to my doctoral advisor – talking to him enabled me to connect everything I learned before. He emphasized structure. This is something I learned from him. Not formulas; formulas always come from structures. You ought to try to understand the structures. And after that many things in mathematics made sense – the whole thing became one subject, which was an insight none of my teachers in China had had.

NTC: *I see, I see. And of course I think that's one of the main beauties of mathematics: everything is –*

JTD: – they're all related –

NTC: *You have this unity, and everything ties together –*

JTD: Yes, yes, yes.

NTC: *– and one has all of these beautiful structures.*

JTD: You specialize, of course, but in the end there are just a few fundamental problems.

So I went to Yale – in 1990 when I arrived in JFK – I think it was JFK – I had fifteen dollars in my pocket. But I had a friend in New York, so he picked me up, and then I borrowed money from him. I went to Yale, I borrowed money from the math department. That was how I started everything because we were so poor at that time.

NTC: *We were talking about pure math, how pure math talks about relationships between things, and how that is part of the beauty of pure math. You've done a lot of work in applied math also – do you see anything in applied math that you would call beautiful? Perhaps not in exactly the same sense, but maybe a different kind of beauty?*

JTD: I find it strange that people differentiate between applied math and pure math. For me, first, the best math ideas always come from the real world. For example, differential equations. We didn't invent any interesting equations – most of the interesting differential equations are all related to physical problems; basically we didn't invent any. Number theory is an exception. In all other subjects, the fundamental things are related to some kind of application. Even Galois theory: prove that there is no formula for solving polynomial equations of degree 5 and above – this is actually an applied question, in my opinion.

Therefore, in the things I do – for example, RSA. It's a fantastic, beautiful idea; an unbelievably beautiful idea, just as beautiful as anything else. And all of us rely so much on it. There is another protocol called Diffie-Hellman key exchange, and it's related to commutativity: it requires finding two functions that commute under composition. It turns out there are very deep mathematics behind it. So it's – how should I say it: true applied math is really as beautiful as pure math. I don't see it as anything else. Of course, even in applied math there is a lot of tedious work (engineering type of work). But even in pure math – like the proof of the classification of simple groups is engineering in my opinion.

In cryptography we just do algebra; it is all about algebra – also geometry, related to lattices and so on. There are a lot of actually interesting questions. But we have a different perspective because we care about computation. In pure math, people don't really care – often we prove existence, and as to how to actually find it, many people don't care. But we do care – we want to know how to find it; we care about computations. So that's the difference.

I think applied mathematics is just as beautiful.

NTC: *Do you have any perspective on why it is that so many mathematicians seem to think pure math is better – that pure and applied math are different, and they prefer pure mathematics to applied mathematics?*

JTD: That is an interesting question. I think partly it is just prejudice: some people like to think that their specialties are better than everyone else's. (This happened to a good German friend of mine, Johannes Buchmann. His father was a professor in English literature and was very upset when he went into mathematics because he thought the humanities were higher than every science.) So I think it is just prejudice.

You think of people like von Neumann, or Turing – they did both. But it is a very unfortunate situation. Applications of math can really drive the development of math. For example, you see what [S.-T.] Yau does is in some sense driven by string theory.

NTC: *Right.*

JTD: I don't know. For me, once I started cryptography, I found that those people are very smart. There are very interesting ideas involved. So the work is just as important, and the ideas are just as interesting – as fascinating. They invent new zero-knowledge proofs, homomorphic encryption, multi-party computation, public-key cryptography – it is all backed by fundamental math. There are so many fundamental math problems we don't know how to solve.

NTC: *Right, right. Do you have any perspective on how one might encourage pure mathematicians to be more interested in applied mathematics?*

JTD: There is a very interesting gap here. Compared to physicists, mathematicians like to work alone. But when you do applied math, you cannot do that – for example, in cryptography one has to write programs; whatever you say often requires you to compute it, to test the speed, and to attack it – you have to perform, to make sure it works. So I work with people who can program. I very rarely write papers alone. In pure mathematics we tend to hide our ideas because of all of these priority arguments – there are many stories. So maybe that's one reason too. This culture has to change.

NTC: *You mean this culture of working alone?*

JTD: Yes, I think applied math fundamentally requires working with other people. Because you're not an expert in that application area. You have to talk to people and understand the fundamental problems. Maybe that's the reason. You have to learn how to work with other people.

People in mathematics normally are not good at working with people. For me that's probably the biggest problem – go and talk to people. Because normally, when you start to work in some applied area, you have to talk to the best experts in the area and listen to what they think first. That can be difficult to take for those super-good mathematicians – they think they know everything and that's it, period.

NTC: *[laughs] Yes, I know what you mean.*

JTD: In my opinion, physics has an error-correcting mechanism: you can write whatever crazy theory you want to write; then somebody will do an experiment to check it. And 99% of the theories are wrong. But nevertheless it's OK – we can find the good ones. But now you don't have this filter, things are very different. Actually, my old area – when I went to Yale I studied the representation theory of quantum affine algebras; it's closely related to string theory, quantum groups, and that stuff. It was too much – I don't know what it is, I don't understand the physics so well so I don't understand those speculations in physics. So I feel it's too much.

NTC: *Did that play a role in your ultimately moving into cryptography?*

JTD: In some sense. At Yale I also had an interesting story – why did I go to Yale: I did number theory, and the person I wanted to work with – the reason I went to Yale is because of this guy called Ilya Piatetski-Shapiro. He's one of the Wolf Prize winners; he was in modular forms. He was Russian too – my advisor was Russian; at that time the graduate program director at Yale was Igor Frenkel. I went to the first class of Shapiro's. I didn't know he had Parkinson's disease. His speech was very blurred – he had a strong accent, plus Parkinson's disease, so I didn't understand anything he said. But at the end of the class he asked one question I did understand: he asked the whole class, Do you understand my English? This I understood! I said, No. I was the only one that said no. He had a very interesting face, and very sharp eyes, and he stared at me for something like five minutes. I said to myself, "Oh no, now you've offended the guy" – I thought the U.S. was a democratic country, you know. "Sorry about that." So no more number theory. Then I talked to Igor Frenkel and said, I don't want to do this anymore. He said, Why don't you study with me? I said, OK. I didn't even know what he was doing. It was a very random choice. So I started to work on quantum groups, but I learned something new, and I like to do new things. So that's what I did.

At Yale it was almost the same story again [as at USTC] – I almost finished my thesis the first year. Igor Frenkel asked me to do some question; I solved it quickly, I did some more work, and in the end – I spent a lot of time doing other things. I told you I like to read history. So I went to Yale and borrowed a lot of history books, especially about Chinese history and world history. I took my time and I finished in five years. I had nothing to do with cryptography. In the third or fourth year I met [Tetsuji] Miwa and [Michio] Jimbo. They offered me a job right away; they said, Why don't you come to Japan? I said, Why not? So I took a job offer even before I finished. When I finished I went to Japan for three and a half years. I had a great time in Japan. I did some work, but I mostly enjoyed Japan. I made some Russian friends; I became friends with Boris Feigin and a few other people. Japan is also a very interesting country, so I spent time reading about Japan and understanding Japan. But nevertheless I enjoyed my time there; I lived in Kyoto for three and a half years. It was a beautiful city and I was treated pretty well; I had a big house – there were always parties in my house; I met my wife there, and my son was born there. Then I left – I decided there was no way I could live in Japan; especially, I could not raise children there. I was at RIMS in Japan for three years. Then I got a job offer in Cincinnati.

Then I started to feel bad in the following sense. I don't know how you feel about it: by 1998 I did pretty well, publishing-wise; I felt it was becoming a routine – I had learned everything I needed to learn, I had many problems to work on, whatever problem came I had a fixed way to do it, more or less, I could get a reasonable result. I could see the way my life would turn out if I continued doing the same thing. Also I could see what some old people were doing. I wasn't happy about all this. (Maybe it was also because funding in the U.S. at that time was difficult, and Cincinnati was definitely not a top-notch university; but it was OK, I mean, I don't care.) I had spent three years at RIMS in Kyoto University; after I had been at Cincinnati for two years – in 2000 – they decided to give me tenure. (At the time I applied for tenure there was actually another professor who applied to be a full professor: my file actually looked better than her file, so I realized that was a good opportunity for me to apply for tenure, because they couldn't reject me.)

After I got tenure I said, Maybe I should do something different, because this is getting boring. That was the year 2000. In 2000 I happened to see this article about this person named Isaac Chuang; he spent $15,000,000 to factor $15 = 3 \times 5$. I said, This is fascinating! And I realized, Oh, the cryptography has to change. Let me do the new cryptography, cryptography that a quantum computer cannot break. So I started there. I just gave up everything; I made a conscious decision and stopped working there completely – of course I finished writing the paper I was writing at that time – so if you see my citations in Google, you can see I have citations, and then it's totally dead, and then it starts again. So you can see when I stopped working – you can see it.

Then I just moved to cryptography. I learned it by myself. I didn't have any collaborator; I didn't have any students – oh, I had one student at the time already, yes, I had started to take on PhD students. So I started learning cryptography in the year 2000, and I started publishing in cryptography in 2004. It took 3–4 years before I started publishing again. That's the history. I basically made a bet – at the time nobody was doing postquantum cryptography; I just saw it on the news and said, Oh, this is interesting. But I had tenure – for me I am very grateful for the tenure system because there's no pressure; what could they do to me?

NTC: *That is a question I had, actually (though it hadn't occurred to me that you already had tenure by then) – did you feel there was a lot of risk involved?*

JTD: There is a risk – I didn't know anything about the area. But I also felt, I have a PhD from Yale and so on; I should be able to learn anything in the world – I should be able to understand it. I firmly believed that I could understand anything as long as I had enough time. I had that confidence. There was definitely a risk; I went to a new area, even the way of writing papers was very different. I needed programs so I had to work with other professors. My programs were terrible – there's a professor called Dieter Schmidt, he is a colleague at Cincinnati, in the computer science department. I wrote a program; it was very short but didn't run well at all. He made it ten times longer, but a hundred times faster. When you write a program, sometimes you should save certain results for reuse, but usually I never did that, I just ran the routine again. He said, How can you do this??? How can you do this??? How can you do this??? My response: OK, then, it's your job! We started to work together – at least, I convinced him this is an interesting question, so he helped me go over it.

NTC: *Makes sense. So is this course a course you would recommend to other academics in a similar situation? For example, if there are other researchers who feel –*

JTD: Stuck?

NTC: *– who feel stuck, right.*

JTD: I think this is very high risk. I tend to take risks – even at Yale, I feel I was reckless to just change like that. But if you really feel bored, you should, I think. Otherwise, what's the point of your life? As for where to go, I have no idea. It's better to go to an area where you see greater potential, rather than go to a new routine area, I think. I would suggest a new area, because there you are almost on an equal footing with others; if you go to an old area then you are far behind already. But think carefully – really think carefully. I think it is more important to bet for the future, for the big directions.

For a while this area was almost totally ignored, at least in some ways. In 2000 I started. In 2000 nobody was doing it – I was not aware of anyone. By 2006, there were a bunch of us who started to talk together, mostly in Europe. Then we started to organize a conference – in 2006. I organized what was actually the first serious conference, in 2008, in Cincinnati. So things started moving. In 2015 everything changed, because the U.S. government wanted to make the next generation standard, and it became very hot. Right now it's very hot.

NTC: *And of course Rainbow was part of –*

JTD: Yes, Rainbow was part of it. But I also did one other thing. I gave a lecture at Harvard; you can Google it, it's on YouTube. I can tell you the story, actually it's interesting. There's a very famous key exchange algorithm. In public key cryptography, one of the fundamental applications is to be able to share the key: you and I want to talk, but the first thing we would like to do is to have a shared secret, which nobody else knows. We can talk over the Internet, but everybody can read our emails. What do you do? So this is the first Diffie-Hellman key exchange, very nice. [Writes on blackboard.] You choose a large prime number p; everything is done mod p. Then you choose a random number g mod p. Alice and Bob secretly choose numbers a and b; Alice computes g^a and Bob computes g^b. a and b are secrets. Then they exchange these computed numbers openly. Then once Alice gets g^b she can compute $(g^b)^a$ (mod p, of course); also Bob can compute $(g^a)^b$, and they're equal because they're both equal to g^{ab}. This is the famous Diffie-Hellman key exchange.

When I started to work on postquantum cryptography the first thing I wanted to do was to rebuild this using something which does not rely on number theory at all. I spent five years working on this and couldn't do it. At the end of the fifth year I found a very interesting paper written by this guy called Joseph Ritt, in 1923.[4] I found it by pure accident; I don't even know how I found it. Ritt proved the following. On the complex plane, if you have two rational functions $f(x)$ and $g(x)$ which commute under composition and are nontrivial (I will define nontrivial later) then there are only three choices: (1) power functions; (2) Cheybshev polynomials; (3) rational functions coming from elliptic curves. (On a rational curve you can multiply points and you can raise to a power by hiding points, and you get rational functions.) There are only those three classes. This was terrible for what I was doing because quantum computing can solve all three of them, because they are all coming from group theory: the automorphism group of the punctured plane and essentially the complex unit circle. So Shor's algorithm can solve this, and the other one comes from elliptic curves – people had already studied it. So that looked like the end of the road for what I was trying to do.

(By nontrivial, I mean that I exclude cases where f and g are both powers of the same function – those always commute.)

So I gave up; I realized we already knew all of these, and they were all useless. Then in 2017 there was something else called learning with error bound (LWE). The guy who invented this is called Oded Regev; he was at the Weizmann Institute, now he's abroad. I realized that I could use this to do a new kind of key exchange by using the following facts: given three matrices, you can multiply from the left or multiply from the right – they commute. But you cannot use this by itself. Learning with error bound is basically a linear algebra problem with noise – it's linear algebra with discrete noise, and then they reduce the problem to a lattice problem. Based on this, I invented the first postquantum key exchange. It's very interesting.

Probably you've heard of RSA, right? So in cryptography there are three families of systems: one is encryption, one is digital signature, another is key exchange. It turns out people had invented many, many kinds of encryption, and many, many kinds of signatures, but they could not invent a single new key exchange. So I invented the first one.

In 2011 I invented the first postquantum key exchange and filed a patent, and then in 2012 I published a paper. Then in 2014 there's a guy whose name is Chris Peikert; he just basically rewrote my paper and claimed he invented this.[5] I invented a new function called a signal function; my function looked like this [draws on blackboard] – 1, 0 – and he invented a new function that looked like this – 0, 1, 0, 1; so this is my invention, he just refined it and claimed it was his invention. But the good thing is, I had a patent.

[4] See [15].

[5] See [8, 7, 13]. We reached out to Professor Peikert for comment but did not hear back in time for publication. For a thorough study of the matter which supports Professor Ding's account, see Daniel Bernstein's blog post [3].

So then I started to fight. But there was not only that – in 2016 Google used it. Google used a variant of this – called New Hope – you can read about it online.[6] Then I told them and they stopped using it. We were negotiating it; they wanted to buy my patent for a very ridiculous number. I said, No way!

So this is what I did before. Rainbow is important even though it lost – but I'm actually preparing for one more round because they need more signatures, there's one more round for the NIST signature competition, the deadline is June 1. So I'm working hard to submit some new things. In the end they chose four algorithms: one is key exchange – that's Kyber; my name is on Kyber. Not only that, I also licensed the patent to NIST. They realized without my license people may not use it – I can sue everybody. Then the others are signatures, but NIST are not happy with the signatures so they are going to make one more round. I think we have a good chance to win – a very good chance to win – so right now I'm preparing. Rainbow is dead for this new round, but I have some new ones.

NTC: *I see, I see. Rainbow is dead – was this because of the March 2022 –*

JTD: Yes, there was a new attack. I can easily correct it, but then I decided to – because there's another one called UOV [unbalanced oil-vinegar]. UOV is even simpler – its efficiency is a little bit lower. After much thinking I think we should just do UOV and stop doing Rainbow. Rainbow is still OK, I can easily correct that by changing the numbers a little bit, just a tiny bit.

NTC: *How would that impact Rainbow's usability?*

JTD: Not much. It was just a simple mistake we made. Efficiency wise, not much – it would be a 10–15% difference.

NTC: *Rainbow also used unbalanced oil-vinegar –*

JTD: Yes, it was based on unbalanced oil-vinegar; so I just do unbalanced oil-vinegar but with some improvements I thought of recently. I went to the U.S. for three months and worked with a bunch of people who do UOV itself. On the flight back home, I had an idea to fundamentally improve UOV. So now I'm working on it; hopefully I can catch the deadline.

You understand the background, right? Quantum computing can do factoring, it can do discrete log, so it can break all current public key cryptosystems.

NTC: *I mostly know it can do factoring.*

JTD: No, no, no, no, no – it can actually solve a much more general problem: it can find the period of a function. And this can be applied to factoring and to discrete log. Given a function, if it has a period then you have a very good way to find the period. For example, as a multiplicative group you can consider the function g to a power – this is a periodic function. And discrete log is all about period. You can find cosets – basically you can count the coset sizes. It's very interesting: it can do both factoring and discrete logs; and that is why it can also break elliptic curve cryptography, because it uses discrete log.

Rainbow is multivariate, so that's based on algebraic geometry – it is related to solving quadratic equations. And then this key exchange is related to lattices.

[6] See [2]. For more on the history of Google's short-lived experiment with postquantum cryptography, see Daniel Bernstein's blog post [3].

So these are two families I'm working on. There's another one related to isogeny of elliptic curves, and then there's another family related to coding theory. Then there's also a family related to hash functions. So we basically have five main families; I work on two of them, multivariate and lattice.

NTC: *For lattices, what problem is it that's hard to solve?*

JTD: The most interesting problem for us is, given a lattice and a random basis, find a short vector in the lattice. Then the shortest vector is the secret key. I don't give you that – I give you a random basis; suppose you have that in a simplified form. The public key is a random basis but the volume is the same: since it's a basis it still spans – as a free group it still generates the whole group. But in order to do encryption you need a short vector, so your job is to find a short vector. There are fascinating theories related to it. The most famous algorithm related to it is called LLL reduction. It is actually related to Siegel modular forms.

So my story is interesting: I learned computational math, so I understand computational theory; then I studied Siegel modular forms when I was a graduate student – it turns out it's all related to lattices. Siegel reduction is actually about reducing a lattice; it's the same thing. So when I started learning about lattices I said, Wow, this is what I learned as a Master's student. And then I even learned some of the fundamental ideas in quantum theory: for example, I proposed a perturbation which is related to quantization (in spirit). Somehow I've used everything I've learned.

NTC: *I have had the same experience – sometimes some random thing I've read over here relates to some other thing over there.*

JTD: Exactly, exactly. Cryptography is particularly fascinating: we don't care how you solve a problem – especially in cryptanalysis, when you attack we don't care at all what you do as long as you can break a system. So people think about very, very interesting ways to attack a system. For example, I can tell you a story about RSA. So in RSA you have to compute huge powers of huge numbers; to make it usable, what you do is you use a square method to accelerate the computation. It's the same in Diffie-Hellman: when you compute g to the power a, g and a are huge numbers – for example, 512 bit numbers. You cannot compute it by multiplying out one by one. So what you do is to perform a binary expansion, and then you square – square – square – square, and then multiply them together; so the number of multiplications is only $2\log n$. But computers are fascinating. At least in the RSA case, we found out that when you multiply two different numbers it consumes more power than squaring. So we knew when you were squaring or when you were multiplying, and by doing that we could read your secret keys.

NTC: *I've heard about timing attacks.*

JTD: These are related to timing attacks. We care a lot about timing attacks; timing attacks allow us to read all your keys. Very interesting, isn't it? We have to develop a new way to do things to make sure that the timing is consistent. Constant timing becomes very important. I do some work on this, not too much.

NTC: *When doing cryptanalysis, would you say there's any overall techniques people try to use –*

JTD: Yes, always.

NTC: *– or is it really very ad hoc?*

JTD: Also ad hoc. But the fundamental mathematics are, for example, lattice reduction: this is actually related to Siegel modular forms, as I said. If you look at Siegel modular forms, you realize finding the fundamental domain of a Siegel form really amounts to finding the shortest vector. Do you know about Siegel modular forms? Basically, you apply them to the upper half plane, but it's a very special half plane. You have A plus i times B – A is a symmetric matrix, B is a positive-definite matrix. This is your upper half plane. Then you do symplectic transformations – if you have a symplectic group, you can do symplectic transformations. It is almost like the two-dimensional case. It turns out you have a fundamental domain problem. Fundamental domains come up in the following way: you have all of the positive definite matrices, and then you have an action of $SL_n(\mathbb{Z})$ on it by multiplication by T and T^t. Then your job is to find the fundamental domain, find the coset, define the moduli space. This is called a Siegel fundamental domain. Basically, because each positive definite matrix can be understood as a metric of a lattice, you have a lattice, you have a basis, and you can multiply it together with its transpose: this is exactly the metric matrix. An $SL_n(\mathbb{Z})$ transformation is basically a change of basis. They are the same basis. Therefore the Siegel modular form is essentially finding the class of lattices which are equivalent.

Then Siegel defined the fundamental domain. One of the key ingredients of the fundamental domain is basically finding the shortest vector: you do a change of basis such that the first vector is the shortest, and then basically the second is, independent of that one, the second shortest. Siegel doesn't have a formula how to find it, but Siegel defined this and proved everything. This is the fundamental mathematics.

A few years ago I hired an assistant professor – his name is Seungki Kim – he was a graduate student of [Akshay] Venkatesh – this guy at Stanford who won the Fields medal; he's now in Princeton [at the IAS] – and they used this Siegel modular form theory to prove some very interesting things related to cryptography. That's why I hired him.

In multivariate cryptography, one fascinating story is Gröbner bases: the whole theory about finding Gröbner bases can be applied. It basically addresses the question, Given two varieties, how do you know they are equivalent? You find Gröbner bases; if the Gröbner bases are the same they are equivalent, if not they are different.

There are certain things which are fundamental. But of course there are new things – for example, this attack on Rainbow: I didn't observe something – somehow we missed it, for 20 years we missed it.

Another interesting problem we do is called the minrank problem – probably you haven't heard of it. There are fascinating stories. How we develop algorithms, step by step, to become faster, faster, faster. There was a fundamental breakthrough two years ago – three years ago by people from NIST. Very interesting ideas. So I do a lot of work on that.

NTC: *When you talk about making the algorithms faster and faster, are you talking about cryptography or cryptanalysis?*

JTD: Cryptanalysis. I do a lot of work on that. There are a lot of fundamental math related things. For example, there's a student who is a graduate student with me now; his name is Guo Hao. As his undergrad thesis, I asked him to do something – I knew more or less how to do it. He did a great job. He proved – there are two different attacks; we had no idea how they are connected. He built a way to convert one attack into another, both ways. It was very nice. [S.-T.] Yau gave him a prize at the ICCM – he won one of the gold medals – and the paper was published and everything. There are a lot of fascinating mathematics. We need to do more, especially on lattice reduction. Another graduate student – his name is Zhao Ziyu – he just became my graduate student in the fall. In his third year, he came to talk to me, and then he started to work on something. In cryptography, in lattice crypto security we have a challenge – it's a global challenge, it's a published challenge: if you solve the problem then they put your name on it. It is getting higher and higher and higher. Within half a year, he became top in the world in one category. So there is a lot of work to be done. I was shocked – he's very good, but I was shocked how easily he was able to get on top. That means how much work we need to do in this area.

NTC: *Could you explain a little bit more what this was about?*

JTD: So as I said, an attack on lattice cryptography basically amounts to finding a shortest vector. We believe 1000 dimensions is secure, but we want to know how secure it is, so we have an algorithm to attack. Then people give challenges – 50 dimensions, 51 dimensions, 52 dimensions, we go up and up; we want to see what the best we can do now is. Then we see how far we are from being broken. So it is a challenge with a moving target.

NTC: *How many dimensions can you break?*

JTD: Exactly. The complexity is related to dimension. It is exponential – to the nth power; but it is a very strange power, not 2 to the nth power, 2 to the zero point something.

But you have a lot of heuristics – there is a lot of engineering work to be done too. But also there are fundamental mathematical ideas we need. For example, I can tell you one more fascinating story, about reduction. In reduction, to evaluate the quality of the reduction there is something called the root Hermite factor. So the error algorithm – we have a theory to prove that the root Hermite factor should be 1.04 – or 1.02? I forgot; 1.04, I think – but when we do experiments, it's way better: theoretically we thought it was a much harder problem, but practically it is much easier. We have zero idea why, even now. So our complexity estimates on lattice algorithms are all very heuristic, which means we do experiments and then we do interpolation. We have no theories supporting that – unlike multivariate cryptography.

Something else – in lattice crypto you will have something called provable security. You have to understand that proof doesn't mean you have – they claim we can prove it's secure, but it's not true at all: "provable security" by definition means that if you can solve this problem you can solve a family of cryptosystems – you can solve a family of hard mathematical problems. I don't care how hard the problem is, but I think it's hard.

NTC: *Like NP-complete or something?*

JTD: No, no, no, not even – we cannot reduce to NP-hard problems. We do some problem we believe is hard – we do not even know how hard it is. Then into a practical attack – we have no theory supporting us, so given a practical one, you ask how hard it is, we have no theory that says. But for multivariate, we don't have provable security – we cannot prove this. Now I think I have a new version for which I can do it, but anyway – we cannot prove it secure. But we have a theory which says, Give us a system; we can tell you how much time it would take to break. We have a theory, and then we do experiments: they match 100%.

NTC: *But then for lattice you don't have that kind of thing yet?*

JTD: No, they just do speculations.

So there are a lot of interesting things going on. Of course, as a mathematician coming to cryptography, there are a lot of clashes, you know. We are humans; they think this is their territory. I think people are unhappy I picked it up, because they have been thinking about it but they couldn't solve the problem; at the last moment I took the best result. I think they are kind of against me. But I have a patent. But you have to protect yourself – even in math it happens. But with a patent I think I managed to protect myself. But without a patent I would be completely out of luck. People used to call it Peikert exchange; nobody says that anymore. I call it Ding key exchange, but people don't say that. They try to avoid the topic. It's a fascinating story.

You can check the older versions of the New Hope paper – IACR preprints have a history and you can retrieve any version.[7] At first they called it Peikert key exchange; then they said it's a variant of Ding's key exchange. So I have no problem with them anymore. But still as a community they try to not give me enough credit, I think. Obviously, this is life. At least they added my name on Kyber. This is life, people are human, but cryptography is particularly nasty and there are many stories like this. Partly because there's money involved.

Normally it happens that there are a bunch of people who really care about ethics, and they make sure people get proper credit. In other areas they don't have strong leadership in dealing with people who are not following the basic ethics. I think this is what happened to me. It has happened quite a few times. There's a Japanese mathematician – he's also a cryptographer. He invented something – identity-based encryption; it was first published in Japanese. Then he submitted the paper to Asia Crypt, in October, and the paper was rejected in December. More or less the same paper appeared later in Crypto, by someone else. I feel lucky in that sense but at the time I knew that so I realized I must have a patent to

[7] IACR stands for the International Association for Cryptologic Research, which hosts a preprint server for cryptography-related research (see `eprint.iacr.org`). For the changes in the New Hope paper, see [2, 1], particularly footnote 1 on p. 1 of [2].

protect myself. I had to write my patent myself, I could not afford it – it was too expensive – so I learned how to write a patent. There are some similar patents; I just copied the language and so on.

But I like cryptography. One thing – you know, in mathematics, opinions are quite important. The big guys say, You wrote a good paper, and that's it. In cryptography, at least there's one thing: it doesn't matter how big you are – you design a cryptosystem, and I break it: everybody knows it – there's no way you can hide it. Also, in math, who is better: it's difficult to judge. So you're Perelman, and you prove the Poincaré conjecture: we all know you're good. But otherwise, it's just a bunch of guys who decide. In this aspect I like cryptography – once you've done something good, nobody can deny it. They tried to do that to me too, but it's harder.

NTC: *Maybe one last question – it seems that cryptography has political overtones, between the U.S. and China –*

JTD: Not really, no.

NTC: *Maybe I misunderstood –*

JTD: What do you mean? Give me an example.

NTC: *I thought it was sort of a sensitive issue, if you're working in cryptography –*

JTD: Yeah, it's a sensitive issue. The U.S. has a law called export control. Before 1990-something you couldn't send encryption as runable code. In China, too, for a long time cryptography was not allowed, except with government approval.

In the U.S. we have to thank Daniel Bernstein – because he sued the government, we don't need government approval to publish papers.[8] But export control rules still apply, so that's one of the reasons I don't touch programs. Only my students program; I never touch programs. I tell them about the risk, and then if they send the program to somebody they shouldn't have, they go to jail, not me. I warn them about it and then they decide.

Postquantum is very popular now – it's super-popular now, because NIST is making standards. And then there's something called NSM-10 (National Security Memorandum 10): basically the United States government decided that by 2035 everything has to be changed. There's a huge rush to do that. In NIST there's – well, under NIST – this organization called NCCoE (National Cybersecurity Center of Excellence). They organized big companies doing this. Then I attended another conference in Japan. So people are really going to do it. Then the U.S. government is doing something related to quantum – at least right now they are doing some control on quantum technology. Cryptography not so much yet – not so much yet. But I'm very careful – I'm a U.S. citizen actually, I only do open research; I don't do any classified work, regardless of whether it's for China or the United States. I don't want to be in any trouble of any form. We have to be very careful.

[8] For some background information on this lawsuit, see [9].

Nevertheless, what I believe is that this quantum threat is a global thing – it is a threat in China and the United States, so we should work together to solve this. Otherwise, we will have two different networks that can't talk to each other. I believe that; I went to NIST, I went to a big conference in Europe, I went to a conference in Japan.

There's this thing called postquantum migration – it's a big area: essentially every cellphone, every computer – the security under these has to be changed. What worries people most is chips: we have these cryptographic chips – those chips have to be taken out and replaced.

NTC: *And suddenly the idea of having hardware cryptography doesn't sound as good as it did before.*

JTD: No, hardware is still definitely much more secure: we talk about side-channel attacks – timing attacks – if you put things in hardware and seal it properly then people cannot probe it that easily. So hardware encryption is still quite important for highly sensitive information.

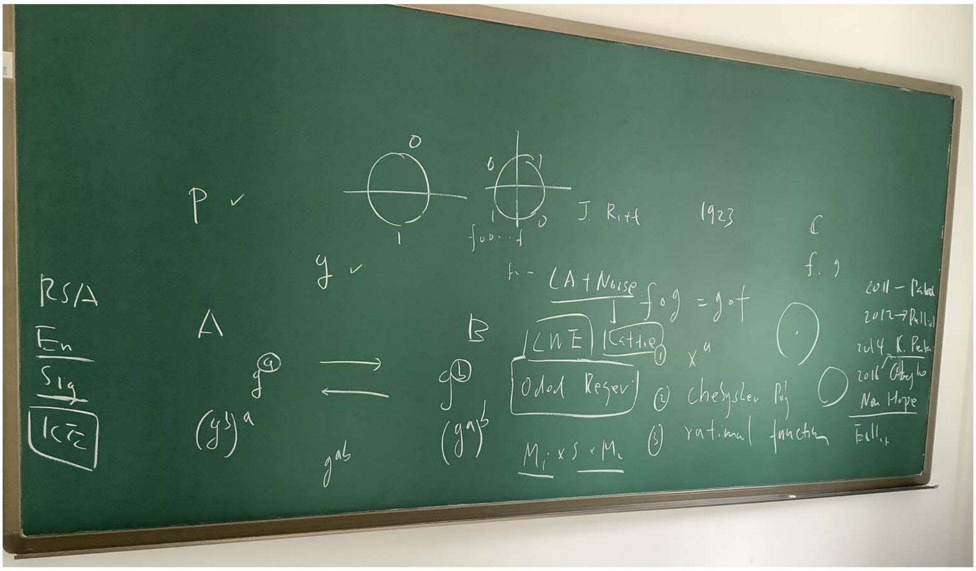
But cryptography is something for everyone – it's not like it was in the Second World War when only military people were using it. One needs to protect privacy. We also do a lot of work at BIMSA on privacy. There's something called multi-party computation: it's related because this also has to be quantum proof, or quantum resistant. So we do a lot of related things. But everything I do is open: I publish it, I file patents – patents are open.

Any other questions? anything else?

NTC: *I think that's all.*

JTD: Good to talk with you.

NTC: *Good to talk with you too.*



Appendix

(Derived from a draft kindly provided by Jintai Ding and Hong Xiang (Chongqing University). Responsibility for all errors lies with the current author.)

In this appendix we provide a brief history of modern computer cryptography and some other supplementary information, together with references for interested readers.

While modern cryptography can be said to have started with Claude Shannon's 1945 report [17], cryptography's centrality to our everyday lives only came about due to the Internet. Following the invention of the Internet's predecessor, ARPAnet, in the late 1960s, encryption algorithms such as the Diffie-Hellman key exchange proposed by W. Diffie and M. Hellman at Stanford University in 1976 [5], the RSA public key encryption algorithm invented by R. Rivest, A. Shamir and L. Adleman at MIT in 1977 [16], and the elliptic curve public key encryption algorithm ECC proposed by American and Canadian scholars in 1982 [10] were introduced to solve the problem of key distribution in large-scale networks. To this day, these algorithms still play a major role in guaranteeing the security of online interactions.

While extremely secure against attack by classical computers, these algorithms are vulnerable to attack by sufficiently large-scale *quantum* computers, via Shor's algorithm [18]. This has necessitated the introduction of postquantum encryption algorithms. One major influence on the work which has been done in developing such algorithms is the U.S. National Institute of Standards and Technology's (NIST) standardization effort, to which Jintai Ding has made great contributions, and which was mentioned briefly in the above interview. NIST's early 2009 report [14] and subsequent 2013 conference talk [11] were followed by the announcement at PQCrypto 2016 by the NIST postquantum cryptography team of a roadmap to producing standards for postquantum cryptography [12]. After three rounds of evaluations, on July 5, 2022, NIST announced the selection of three digital signature algorithms (Dilithium, Falcon, and SPHINCS+) and one key encapsulation/key negotiation algorithm (Kyber, which includes work by Jintai Ding, as mentioned in the interview). All of these, except SPHINCS+, are lattice-based. A further multivariate digital signature algorithm, Rainbow, proposed by Jintai Ding's team, did not make it to the third round, as mentioned in the interview.

The 2016 Google experiment mentioned above (see also [3]) layered a postquantum encryption scheme *on top of* industry-standard elliptic curve encryption. Such a technique allows one to take advantage of the well-established classical security of conventional encryption while simultaneously protecting data against attackers who may store it for ultimate decryption by a future quantum computer (see [3]).

Finally, it should be noted that *key size* is a problem for almost all postquantum cryptographic algorithms: see Table 1 for a comparison of classical and postquantum cryptographic key sizes. This adds extra overhead to network communications.

Table 1. Comparison between key size of PQC and the first-generation public-key cryptography (128-bit security level, all sizes in bytes)

Name	**Type**	**public key size**	**secret key size**
Kyber	PKE/KEM	800	1632
Dilithium	Signature	1312	2528
Falcon	Signature	897	1281
SPHINCS+	Signature	32	64
RSA	PKE & Signature	384	384
ECC	PKE & Signature	32	32

Interview held on 12 April 2023

References

[1] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, *Post-quantum key exchange – a new hope* (2015), URL https://eprint.iacr.org/archive/2015/1092/20151110:173312, Electronic preprint.

[2] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, *Post-quantum key exchange – a new hope* (2019), URL https://eprint.iacr.org/archive/2015/1092/20190710:103122, Electronic preprint.

[3] D. J. Bernstein, *Plagiarism as a patent amplifier*, Blog post (accessed May 26, 2023), January 2022, URL https://blog.cr.yp.to/20220129-plagiarism.html.

[4] J.-R. Chen, *On the representation of a larger even integer as the sum of a prime and the product of at most two primes*, in: Goldbach Conjecture (Wang Yuan, ed.), World Scientific, 1984, pp. 253–272, doi: 10.1142/9789814542487_0019, URL https://www.worldscientific.com/doi/abs/10.1142/9789814542487_0019.

[5] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **22** (1976), no. 6, 644–654. MR0437208

[6] J. T. Ding, *A proof of a conjecture of C.L. Siegel*, Journal of Number Theory **46** (1994), no. 1, 1–11, doi: 10.1006/jnth.1994.1001, URL https://www.sciencedirect.com/science/article/pii/S0022314X84710018. MR1268360

[7] J. T. Ding, *A simple provably secure key exchange scheme based on the learning with errors problem* (2012), URL https://eprint.iacr.org/archive/2012/688/20121210:115748, Electronic preprint.

[8] J. T. Ding, *Cryptographic systems using pairing with errors*, January 2016, U.S. patent office, Patent 9,246,675.

[9] Electronic Frontier Foundation, *Eff sues to overturn cryptography restrictions*, Accessed May 26, 2023, February 1995, URL https://www.eff.org/press/archives/2008/04/21-42.

[10] V. S. Miller, *Use of elliptic curves in cryptography*, in: Advances in Cryptology – CRYPTO '85, Santa Barbara, California, USA, August 18–22, 1985, Proceedings, Springer-Verlag New York, Inc., 1986. MR0851432

[11] M. Mosca, L. Chen, and Y.-K. Liu, *Practical impacts of quantum computing*, 2013, ETSI Quantum-Safe Cryptography 2013 Program Committee.

[12] NIST, *Submission requirements and evaluation criteria for the post-quantum cryptography standardization process*, URL https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[13] C. Peikert, *Lattice cryptography for the internet* (2014), URL https://eprint.iacr.org/archive/2014/070/20140204:155644, Electronic preprint. MR3278403

[14] R. Perlner and D. Cooper, *Quantum resistant public key cryptography: A survey*, IDtrust 2009, Gaithersburg, MD, April 2009, doi: 10.1145/1527017.1527028.

[15] J. F. Ritt, *Permutable rational functions*, Transactions of the American Mathematical Society **25** (1923), no. 3, 399–448, doi: 10.1090/S0002-9947-1923-1501252-3. MR1501252

[16] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures*

and public-key cryptosystems, Communications of the ACM **21** (1978), no. 2, 120–126. MR0700103

[17] C. E. Shannon, *A mathematical theory of cryptography*, (1945), URL https://www.iacr.org/museum/shannon/shannon45.pdf.

[18] P. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, in: 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 124–134. MR1489242

Nathan Thomas Carruth
lutianci@mail.tsinghua.edu.cn
YMSC, Tsinghua University, Beijing (China)