Interview of Phillip Rogaway by Gerardo Con Diaz, 2024.09.24

Con Diaz: It is September 24th, 2024, and this is an interview for the Charles Babbage Institute's National Science Foundation project "Mining, a Usable Past: Perspectives, Paradoxes, and Possibilities in Security and Privacy." I'm here with Phillip Rogaway, emeritus Professor of Computer Science at the University of California Davis. I'm in Davis, California, and Phil is in Thalang, Phuket, Thailand. This is an oral history being conducted through video conference over Zoom and the meeting is being recorded.

Phil, thank you for joining me today. Let's begin with some very basic biographical questions. Can you tell me when and where you were born?

Rogaway: I was born in Los Angeles, California, in 1962.

Con Diaz: And what were your parents like?

Rogaway: I had a wonderful mother. She was a librarian and schoolteacher. She valued reading, education, her many friends, and doing good work in the world. She was Jewish, liberal, and very sociable.

I had a less wonderful father. He had wanted to be a cinematographer but ended up selling insurance. He didn't pay a whole lot of attention to the two kids, me and my older sister Jodi. But he was okay towards us; it was my mother with whom he could not get along.

Con Diaz: What was your childhood like, Phil?

Rogaway: Not happy. My parents fought constantly, and I dreamed of escape. I didn't know what I was, but was sure I wasn't human. I didn't form friendships until high school. Maybe that was because I'm profoundly faceblind. Unable to tell kids apart, I assumed it didn't matter.

I was academically focused from early on. I also loved the mountains from early on. I wanted to escape to the mountains, and occasionally did.

Con Diaz: What would you do in the mountains?

Rogaway: I'd go backpacking. Hike and live for a few days in a beautiful and peaceful space. The antithesis of LA.

Con Diaz: By yourself?

Rogaway: Sometimes by myself, but sometimes I could convince someone else to go, like a friend of my sister's.

Con Diaz: When did you start doing that?

Rogaway: Around 12 or 13. Kids grew up faster then, and my mother believed in giving children all the independence they could handle.

I couldn't get to the mountains often, but I would study maps of the Sierra Nevada and dream. Those old, 15-minute quadrangles.

I never fought with my mother or sister, but one time my father tried to keep me from going to the Sierras. I felt I had to go. We fought on the front lawn. I left, took a Greyhound bus to Lone Pine, and went up from Whitney Portal. But severe weather beat me back.

Con Diaz: You mentioned you were academically motivated from an early age. What were your favorite subjects?

Rogaway: I liked all the sciences. A grade-school science teacher, John Phipps, took me under his wing and let me use the school laboratory whenever I wanted. I was particularly interested in enzymes. But it wasn't just math and science: by high school, I also loved reading, writing, and literature.

Con Diaz: Is this something your mother nourished in you?

Rogaway: She encouraged a love of books. She didn't care what kids read, as long as they were reading something. So yes, my mother nourished that.

Con Diaz: And where did you attend elementary and high school?

Rogaway: In Beverly Hills. My parents borrowed money from family and my father made what might have been the only good financial decision of his life: to buy a duplex in the cheapest corner of Beverly Hills. They did it to get my sister and me into the school district, which was quite good at that time. I think the duplex cost $70,000, and the $300/month mortgage was probably their outer limit. But within a couple of years the upper was bringing in enough rent to cover the mortgage, and they effectively got to live in the lower rent-free. A middle-class family can't do this sort of thing anymore.

Con Diaz: Did you have any exposure to computers early on?

Rogaway: I did. The high school had a minicomputer with a terminal room and time-sharing. An HP 2000E. Kids from the elementary school could go and use it. I never saw the actual machine, but taught myself to program it in BASIC.

Even before then, though, I liked algorithms. I wanted to find the simplest algorithm possible to play an unbeatable game of tic-tac-toe. I still remember the algorithm I invented for that.[1] I guess I had rediscovered the *idea* of an algorithm, and the game of trying to find an elegant one. I remember working on the case analysis during a family summer camp at UCSB. My family had no idea what I was doing, and probably assumed it was meaningless.

Con Diaz: How old were you when you did this?

Rogaway: I'm not sure. Maybe around 10.

I tried to build a hardware embodiment of the algorithm, too, on a big wooden board. It never worked. On the other hand, getting it to run in BASIC was easy. My aim was to minimize the number of lines of code, or the number of characters.

I learned another programming language, APL, from a pretty book by Gilman and Rose. APL's aesthetic coincided with my own: it encouraged you to write extremely concise programs using a syntax full of mysterious and beautiful symbols. I had no access to a machine with APL, but I didn't care: the interesting part of programming was done on paper.

Con Diaz: After high school, you went to UC Berkeley and majored in computer science. Could you tell me about your time at Berkeley?

Rogaway: Life was so much better at UCB. Going to college meant moving in with my girlfriend. I was 17. She was Persian. Extraordinarily beautiful, intelligent, and deep. I worked hard in college, probably more than 60 hours a week, but the girlfriend was my focus. We were poor, nervous, and depressed, but life felt authentic and intense.

I started off as a neurobiology major. I thought I might go to medical school, but found that I was better at CS than stuff like biochemistry. You didn't need to remember anything for CS. So I switched to that, but spent very little time near computers. I wrote my programs carefully on paper, typed them in, and they

worked. No debugging. Compared to so many other things, CS was embarrassingly easy.

I wanted to take all the classes on offer. There was this amazing catalog that described thousands of courses. You were allowed to take almost any of them. You could also show up to classes you weren't enrolled in; nobody would notice or care. I liked the anonymity of UCB.

Con Diaz: What were some of your favorite classes?

Rogaway: Organic chemistry was fun. I didn't like laboratory work—the smells would impregnate your clothes and make you sick—but designing syntheses on paper was great. So was the thick blue book that we used.

I liked Paul Chernoff's point-set topology class. It used his beautiful course notes, typed on an IBM Selectric typewriter. I remember Prof. Chernoff with sadness.[2]

But the most fun thing was theoretical computer science. I chanced upon a class that Manuel Blum was teaching on NP-completeness. I loved both the way Blum taught and the style of thinking that NP-completeness proofs required. I decided to keep taking whatever Blum taught.

I tried to model my way of problem-solving, and later my way of teaching, off of Manuel Blum. Yet I was too scared to talk to him. In the culture at UCB at the time, at least as I understood it, a brilliant professor lived on a different plane.

In fact, I never spoke to any UCB professor until my final year, when I had to ask Prof. Blum for a letter of recommendation. Now *that* was scary.

It wasn't just professors I didn't talk to: I didn't interact with TAs or other students, either. I thought that a good student was one who sits invisibly in each class, gets help from nobody, but aces all exams and problem sets. I suppose I missed much of what one should get from going to college. But I had my girlfriend, maintained close contact with my mom, and didn't yet see the point in other human interactions. That was stupid, of course.

Con Diaz: The 1980s in the Bay Area were an exciting time for computing. What was it like to be near so much action at Berkeley?

Rogaway: I don't think I was aware that any such action was taking place. I mean, Berkeley was a *politically* active place. Things that would have seemed worthy of

attention would have had to be personal, political, or intellectual. Anything technological or corporate would have seemed unimportant.

Of course, nowadays lots of young people are extremely interested in what passes for progress in Silicon Valley. But in the early 1980s, I don't think that was true, especially for students who saw themselves as aspiring intellectuals.

Con Diaz: After college you moved to Massachusetts and went to MIT for your graduate studies. Could you reflect on your time as a graduate student?

Rogaway: I loved being a grad student at MIT. The Theory of Computation group housed an impressive group of young faculty and students. Silvio Micali and Shafi Goldwasser had just arrived from UCB. David Shmoys, who taught the first class I took, looked like he was 16. My officemate, Miller Maley, I think he actually *was* 16. There was Ron Rivest, Albert Meyer, a secretary named Be Hubbard, … all sorts of people I remember fondly.

In the 1980s, theoretical computer science had acquired the cachet that mathematics and physics had held before. The people coming to MIT's theory group were Putnam Fellows and the like. Lots of prodigies. Compared to them, I was mathematically inept.

The physical setting was weird. We were in an office building across, shared with Polaroid, across from the rest of MIT's campus. The Theory Group was on the third floor. There was a guard in the lobby who'd berate me when I'd bring my housemate's Great Dane to school. I'd glare at him and go on up. I didn't feel that he had any authority over MIT students.

Con Diaz: How did you start sorting through computer science to find your own interests?

Rogaway: Ron Rivest was my first advisor. In the theory lounge one day, a few days after I got to MIT, he described a wonderful puzzle.[3] I made some progress on it that night, which got me an invitation to work with Ron. But I never made any more progress on the thing.

I admired the way Ron worked, thought, and lived. Each day he'd go home to his family for dinner. Later that night I might receive an insightful email. Everything seemed effortless and in balance for him. I found Ron inspiring but intimidating.

I was also drawn to Silvio, who was less intimidating. He wasn't much older than me, and I didn't freeze up as much around him. Silvio had this infectious laugh and easy way of being. I liked the way he allowed philosophical motivations to be the source of the problems he looked at. He became my mentor after Ron.

Actually, I'm not sure mentorship was a formal thing. Faculty pooled their grant money and students were paid from the pot. You didn't need to work on any particular topic or with any particular person. You would TA every other term, which was also a lot of fun.

Silvio would feed me espressos from a secret room in the building. They would have me insanely wired, as I didn't drink coffee at the time. Then we'd go to his office and talk, often for many hours. As I approached graduation, and after, we shifted to outdoor walks, intermingling talk of cryptography and life.

Con Diaz: You mentioned that Silvio had philosophical motivation for the problems he considered, and that you liked that. Could you tell me about that?

Rogaway: At least for Silvio and Shafi, at least in the 1980s, I would say that philosophy was seen as the *only* worthy source of inspiration for problems. The goal was to understand the fundamental mathematical character of privacy and security. You could find that only by pure, insightful thought. I think it's a common viewpoint in the philosophy of mathematics—that key mathematical ideas are "out there," waiting to be discovered. They exist independent of our efforts. The job of the mathematician is to find and investigate the key truths. This was the way of thinking that gave rise to zero knowledge, the formulation of pseudorandom generators, and the definitions of security for public-key encryption. These things were simply *not* motivated by looking at computer security problems. Computer security was seen as a distant and banal domain.

Con Diaz: Would you say then that you started to gravitate towards cryptography because of your relationship with Silvio?

Rogaway: Absolutely, I gravitated to cryptography because of Silvio's personality and his approach to the subject.

When I graduated from UCB I had thought that cryptography was the one area of theoretical computer science that I would *not* do. I was concerned that it sat too close to military and commercial interests. I was already a bit radical by then, by US standards, a gift from the Persian girlfriend. Besides, I was no good at number theory, which I thought to be at cryptography's core.

Con Diaz: How then did you negotiate that impulse to stay away from cryptography with your decision to do it as a graduate student?

Rogaway: One answer is that rationalization can get you anywhere you want to be. But the rationalization here seemed particularly easy, because the entire approach to cryptography that Silvio was engaged in, this cryptography-via-philosophy approach, seemed to ignore or even subvert any connections to state or corporate power. It seemed that any contributions we'd make to governments or corporations would be accidental and minimal. The goal, again, was to find and investigate the *right* cryptographic notions. The goal wasn't to create artifacts, to make money, or to solve practical problems.

Con Diaz: Your thesis was titled *The Round Complexity of Secure Protocols.* Could you tell us about that thesis?

Rogaway: I should clarify first that a thesis was regarded as having almost no value at MIT. You were discouraged from even beginning to write it until shortly before you intended to hand it in. Instead, you were supposed to be writing papers. They'd be published in conferences like STOC, FOCS, and CRYPTO. When you had enough good papers, you could staple them together, slap on an introduction, and that would be your thesis. You assumed that nobody would read it. I believe this was, and remains, the prevailing model for CS Ph.D. students at good schools.

Con Diaz: I see. Could you tell me then about the papers that you stapled together?

Rogaway: The first paper I wrote at MIT was titled "Everything Provable is Provable in Zero Knowledge." It came out of a classroom discussion. There were a bunch of visitors in a cryptography class I attended one day. Silvio presented a proof that everything in IP had a zero-knowledge proof.[4] But the proof had a bug. Several of us contributed to pointing it out and fixing it up. I wrote up the result with lots of coauthors. It might have been borderline folklore: things were moving so fast at the time, and expectations for having writeups were so low, that I think it was hard to say. I used this for my master's thesis.

The MIT librarian assumed that the title was supposed to be funny: for a long time the thesis hung in a display case outside the library with the title "Humor at MIT." But it wasn't intended to be funny, that "everything provable is provable in zero-knowledge." Just a straight-up description of the result.

The work that would wind up in my Ph.D. thesis was unrelated. It was about multiparty computation. A group of *players* would like to collaboratively compute some function. Each goes in with some *input* that only it knows. The players want to compute the function of the jointly held inputs in a way that reveals nothing beyond that which knowing the result implicitly reveals.

Silvio and I described a protocol to solve the problem more efficiently than before: there was a trick to make do with a constant number of communication rounds, regardless of the complexity of the function you wanted to compute. That was one paper.

We also wanted to iron out definitions for this domain. That turned out to be harder. We published a separate paper on this, but we never had a fully satisfactory definition. Ran Canetti is regarded as being the first person to provide one. I can't say that I ever considered the problem to have been properly resolved: the area felt hopelessly informal.

Con Diaz: You graduated from MIT in 1991 and then joined IBM in Austin as a computer security architect. Why did you take the job at IBM?

Rogaway: I didn't intend to; I was sure I belonged in academia. But a couple of things pushed me to go to IBM.

I took a visiting faculty position at Dartmouth during my last year at MIT. I'd return to MIT about once a week, to finish things up with Silvio. I liked Hanover. It turns especially beautiful in winter. I'd walk to campus on its quiet, snowy streets. And there were some cool people in the department. Like this guy Jim Driscoll, who had a rustic cabin in Vermont. He had a student evaluation on his office door that said he was a *God*. Who wouldn't want to befriend a God? I also appreciated Dartmouth hiring me as an ABD (all-but-dissertation). So I wanted to stay at Dartmouth, and interviewed for a permanent position they advertised. I thought it went well, but I didn't get the first offer. When that person declined, I got the next offer up. But by then I felt slighted.

Meanwhile, a group at IBM Austin was bending over backward to recruit me. I had told them I wasn't interested in going to industry *or* Texas. They said they'd send someone over to chat anyway. And the guy they sent was impressive. Bob Blakely knew computer security better than anyone I had ever met. I could learn a lot from him. So I made a trip out to Austin, and the place wasn't so bad. Then IBM agreed to my salary request and even managed to get me two extra weeks of vacation, a

request so unusual for them that their payroll system literally couldn't code it. In the end, I decided it was important to go where you felt wanted.

Con Diaz: Can you tell me about your time at IBM?

Rogaway: There were about 50 new Ph.D. hires sprinkled around the organization. IBM didn't seem to know what to do with them, and I myself didn't know what I *could* do of use.

At MIT I had been trying to solve some problems with fellow grad student Mihir Bellare. He also ended up going to IBM, but in Hawthorne, New York. So we started interacting more than ever. That was the bulk of what I did at IBM: just talk with Mihir.

But at IBM I also became aware of some industry-wide security work. There was this project called DCE, the Distributed Computing Environment, that included Kerberos, the same service UCD still uses. IBM sent me to some standardization meetings, including those for DCE. I found it fascinating that there were people trying to standardize something that was cryptographic, came from MIT, but was totally unknown to me.

At these meetings I would see Bob Blakley's equivalent from other companies, like this guy Joe Pato, from Hewlett Packard. Bob and Joe were so obviously on top of things that if there wasn't any theory-rooted crypto going on, it had to be because my community just hadn't done the kind of stuff practitioners needed. There was this huge gap between the kind of problems that cryptographers from MIT had been thinking about and the kind of problems faced by people who wanted to build secure distributed systems.

Con Diaz: Could you comment a little more on that gap?

Rogaway: Well, the problem that Kerberos aims to solve, the problem of *entity authentication and key distribution*, was unknown to the theoretical cryptographic community. There wasn't a single paper on it from my world.

Similarly, Mihir had identified a basic cryptographic problem needed by folks at IBM Hawthorne. They were interested in message authentication at very high speeds. Messages would stream by at a gigabit per second. This was faster than one could chain DES, which, we learned, was the customary way of authenticating messages. The theory world had ignored message authentication. In fact, theorists had ignored *all* symmetric cryptography.

If you wanted to do something to help these people, you'd need to figure out exactly what problems they wanted to solve. That wasn't easy. It was especially frustrating trying to figure out what problem Kerberos addressed. Papers on it felt like piles of words. I remember this one that was written as a play, a dialogue between two parties.[5] One of them would propose a protocol, the other would attack it, then the first would propose a fix, and so on, for a tiresomely long time. At the end they've arrived at Kerberos. I guess you were supposed to think, *great, they finally got it, and now I see why it works like this.* But from my point of view, each protocol change just proved that they didn't understand what they were doing. I wanted a definition, not a dialog, but the authors were coming from a world that didn't recognize that such a thing was even possible. That was the problem I wanted to solve.

Con Diaz: Defining entity authentication was the first problem you wanted to solve at IBM?

Rogaway: Right. Along with that fast message authentication problem that Mihir had found. I wanted us to do rigorous but practical treatments of these problems.

One time when I was back at MIT to work with Silvio, after a year or so at IBM, I went to see someone at the center of the Kerberos development. Jeff Schiller, I think it was. I told him that the problem that Kerberos was addressing lacked any cryptographic foundations. But whatever those foundations turned out to be, I said, it was clear that a solution *shouldn't* be based on encryption. After all, I explained, even perfect privacy, as with a one-time pad, clearly wasn't enough to guarantee correctness for the Needham-Schroeder protocol, or anything like it.[6] Schiller, if that's who it was, didn't seem to follow or agree with anything I said. Silvio told me that he called him afterward to complain that I had met him, spoken a lot of nonsense, and was disrespectful of their work.

Mihir and I eventually did get entity authentication and key distribution on a proper foundation. But it seemed that nobody was listening. I remember being a little frustrated that our first paper on this, which we published in 1993,[7] was ignored for years. For example, a survey of entity authentication published a couple of years after our paper came out didn't even mention our work.[8] Its author apparently still didn't understand that one could do complexity-theoretic definitions in this domain, and then efficient, provably secure protocols. It took about 10 years until it was widely understood that Mihir and I had done that.

Con Diaz: In 1994 you joined UC Davis as a faculty member. Could you tell me about that transition?

Rogaway: By 1994 I knew how to find problems I liked. Instead of dreaming them up, or answering questions from other people's papers, we could find cryptographic problems that security practitioners had already demonstrated that they cared about, but that cryptographers had ignored. Then, we wouldn't reduce to the conceptually simplest primitive, the usual expectation at the time. We'd reduce to whatever you already had efficiently instantiated. We would look closely at the efficiency and bounds of our reductions, and describe these things with concrete formulas, not asymptotics.

All of this was pretty radical. For example, from our point of view, it made sense to construct a one-way function from a blockcipher. That was just backward for the thinking of the day. Not to mention that a blockcipher wasn't even an asymptotic object, making it seem, to many, as out-of-scope for rigorous cryptography.

There were further wrinkles in how we wanted to work. For example, if you can't efficiently solve a problem in the "standard" model of computation, we thought it fine to work in a richer model. In 1993 we enunciated what we called the "random-oracle paradigm." You pretend you have a public random function at your disposal. Anyone can ask it a value x and get back a random answer $H(x)$. At the end, you instantiate this oracle with something derived from a cryptographic hash function. It was a *thesis* that this would give rise to practical and secure schemes. The approach was already implicit in work by Fiat and Shamir. We wanted to call it out and suggest it as a widely applicable paradigm.

By 1994 we had all these ideas about how to make cryptography more useful, but still rooted in definitions and proofs. I had gotten from IBM all that I could. And UC Davis seemed like a perfect place to jump to. Its CS department was small and collegial, and Davis felt small-town friendly. The Sierras weren't far. I had done my undergrad nearby, and my parents still lived in California. I'm not competitive and didn't see any need to be at a tier-1 school.

Con Diaz: Did you and Mihir keep working in the random-oracle model when you started your research as a faculty member?

Rogaway: We'd work in whatever model felt right for the problem. Neither of us felt attached to any particular model, area, or approach.

Con Diaz: How did the two of you work together?

Rogaway: Mostly by fighting. An external observer would think that we don't get along. Each of us would express strong ideas about how something should look. But neither of us was genuinely attached to our ideas. We would often switch sides, persuaded by others' arguments. Many of our papers underwent a huge amount of flip-flopping in their evolution. And a lot of papers got started but never really got off the ground.

Both of us are obsessive writers. I would write the introduction to a paper, then Mihir would throw it out and redo it. Then I would redo his draft, perhaps keeping some small piece. This could go on for a dozen or more iterations. Many of our papers had introductions, and often other sections, that involved hundreds of hours of rewriting. For a paper like *Random Oracles are Practical*, almost every sentence would have been worked and reworked numerous times.

We would fight about notation, terminology, and definitional choices, changing these things often as each paper evolved, and from one paper to the next. I suppose I could give an entire talk on the evolution of the random-assignment symbol, which I would finally come to write like $r \twoheadleftarrow R$.

Writing and rewriting was, at least for me, a key aspect of working out what I wanted to say. I would work on the abstract and introduction long before we had the results to back it up. That's how I'd figure out what I wanted the results to be. I wanted papers that would read like a story unfolding. You couldn't know if it was a *good* story until you read it, and did so without feeling attached to the work.

I believe that obsessive writing was a key part of our papers' success. Many of the papers were in some way oppositional, and oppositional ideas have to be well written, even beautifully written, to have any chance of buy-in.

Con Diaz: What new projects started to emerge as a faculty member?

Rogaway: Different things would come up. Like collision-resistant hashing, authenticated encryption, garbling schemes, and anonymity. Mostly I did stuff that was conceptual and definitional, not technical. Things that could have been done years before.

I had come to see that there were a lot of problems right in front of us that cryptographers weren't working on because of historical or sociological reasons, like the community split. The cryptographic community had a particular mindset, background, and history, and because of this, there were problems of "obvious"

importance that had effectively gone unnoticed. If you could see the problems that others weren't seeing, you'd have a rich set of things to work on.

Con Diaz: By "community split" do you mean the division between the theory-minded cryptographers and the more applied people?

Rogaway: Yes, but there were other splits, too. For example, there was a universe of people who were studying authentication protocols in unfamiliar ways, like with "BAN logic." It was theory, but theory connected with programming languages and theorem proving, not reductions or complexity theory. I remember discovering that there was this entire community of people working on cryptographic protocols and proofs but who didn't see themselves as cryptographers, and didn't attend conferences like CRYPTO. I thought it was wild that research communities could fracture like that, or could evolve in such a way.

Con Diaz: Could you tell me about your journey with symmetric encryption?

Rogaway: Sure. Shared-key encryption goes way back, but it had gone without a provable security treatment roughly 15 years after public-key encryption had met provable security. It's a clear instance of "cultural issues" keeping something natural from happening for a very long time.

Mihir and I started off with a straightforward adaptation of the public-key notion for IND-CPA security.[9] We did the usual thing of formulating it in several equivalent ways. But, over the next few years, I came to decide that this privacy-only notion was an undesirable target for a real-world scheme. For example, the formulation was probabilistic, and we started to see that that wasn't how security practitioners needed encryption to be. It was too easy to mess up in providing the required randomness. For practical schemes, something based on a "nonce" would be better.

More than that, I came to understand that, when you encrypt something, you usually want the result to be authenticated, too. And further stuff may need to be authenticated, like a message header. So I became a bit of an evangelist for *authenticated* encryption, AEAD,[10] a goal much stronger than IND-CPA security.

When you create a new definition, you create a new abstraction boundary. You hope that people will come to write to it, think to it. And the amazing thing was that, to a large extent, they did. For example, TLS, nowadays, is architected to the AEAD abstraction boundary.

The strengthening continued. After vanilla AEAD I worked on *misuse-resistant* authenticated encryption, then *robust* authenticated encryption. I had come to realize that the stronger the definition, the easier to correctly use a conforming scheme. Or, flipped around, the less likely that your scheme would get misused. Strengthening definitions could be a path to improving security.

To me, the history of symmetric encryption notions undermines the idea that the right cryptographic notions are "out there," waiting to be discovered. Because you could plainly see how, over the years, security notions would evolve based on human concerns. The process was dialectical. There was a community of people who needed to do things with symmetric encryption, and another community who could do definitions, primitives, and proofs. The two sides started to talk. By now my understanding is effectively the opposite of what I had been acculturated to. I see cryptographic definitions and schemes as constructed tools that serve some community, not mathematical basics pulled from the ether.

Con Diaz: A lot of the stories that you've told me about your intellectual journey have been about reconciling different ways of thinking and doing things. Could you tell me more about this process of reconciliation?

Rogaway: I'm not sure I ever thought in those terms, but I think you're right, that I do like to try to reconcile different things. I even have a paper with that word in the title: *Reconciling Two Views of Cryptography*, with Martín Abadi (2000/2002). There we wanted to reconcile these two views of what symmetric encryption *is*: a formal operator that takes a plaintext M and encapsulates it into a ciphertext $\{M\}_K$ about which nothing can be known without possession of the key K; or a probabilistic function E applied to strings M and K to get a string-valued result. The division roughly corresponds to Martín's world and mine. We wanted to show that security in the first sense would follow from security in the second sense if you assumed something strong enough about the encryption scheme E.

A result like that doesn't just bridge two different notions, but two different communities. You hope that insights arise by seeing a thing from another point of view. It can happen. But it can also happen that nobody pays attention to what you've done, because the intersection of people from the two different worlds is empty.

Con Diaz: I understand that you spent much of your career away from UCD, living in Thailand. How did that come about?

Rogaway: After grad school I decided to travel around SE Asia for a bit. I went west from Bali, then up the Malay Peninsula. By the time I reached Bangkok I had been around people quite enough. At the airport, I asked the Thai Airlines agent what was the smallest town they flew to. She sent me to Mae Hong Son. The place was beautiful, magical. But there was no university there, so I took a bus to Chiang Mai. At least in 1991, it too felt magical. I sat at CMU's lake. The hills to the west were layered in fog. There were animal sounds, teak trees. I wanted to make this place my second home.

Arranging that wasn't easy. The CS department had never had a foreign professor, and they didn't want one. Their Chair spoke no English. But a kind professor in the department, Ajarn Darunee, eventually made it happen.

In the 1990s and early 2000s, Chiang Mai was a wonderful place to live. I often stayed in an old teak hotel, where the staff immersed me in Thai language and gossip. I got around on a little Honda motorbike. I kept my possessions few, no more than fit in a canvas bag. I learned to dress politely and speak softly in Thai. I taught the CMU students, with whom I felt a strong bond. Sometimes my colleagues or UCD grad students would come to visit. But I had been collaborating mostly by email and phone for years, so it didn't really matter where I worked, not when it came to doing research. I'd stay in Thailand for 3 to 15 months, then return to UCD to show my face, catch up on teaching, and live in a conventional home. UCD was great about granting my many requests for sabbaticals and leaves.

While I kept returning to Davis and Chiang Mai, I went to lots of other places, too. In the end, I came to work, live, or travel—I didn't see much distinction—in about 70 different countries. Sometimes I'd go alone, but other times my wife, Kot, would join.

I had met Kot on my first extended trip to Chiang Mai, in 1994. We had been neighbors. For decades Kot supported my work by creating a quiet home and an interesting life. Late in life, when I was 47, we had a son. Banlu also felt at home in whatever culture or country we lived. He's currently here in Thailand, too.

Con Diaz: You talk a lot about social and cultural issues. How did you become interested in not just thinking about social issues in technology, but also teaching about them?

Rogaway: A few years after starting to UCD I noticed there was an ethics class in the department, ECS 188. It was offered because it was effectively mandated by ABET.[11] No faculty member wanted to teach it, so we'd hire whomever we could

find to cover it. I went to a class meeting or two. The guy teaching it reeked of cigarettes. He taught a narrow class on Computer Ethics, which was both the title of the course and the lightweight book that it used. The students could not have looked more bored.

It was the early 2000s and the US was busy bombing Afghanistan back into the Stone Age. CS students from departments like mine were helping to make that happen. They did so by taking well-paying jobs at what are euphemistically called "defense contractors." Yet here we were in an "ethics" class talking about whether it was okay to make your friend an unauthorized copy of Microsoft Word. Seriously? And I thought: the students must see this disconnect. If you want to have a meaningful ethics class in a CS Department you better start by owning up to the profound and often negative impacts of CS, and modern technology quite broadly. The fact that that *wasn't* what was being explored felt irresponsible. We were packing the employment pipeline with people we had effectively trained to *not* care about the purpose to which their work is put. So I decided to take over the class, and it became an obsession of mine as strong as my obsession with research.

I felt unprepared to teach a class on ethics and technology. I felt I should have not just a background in CS, but also in philosophy, sociology, history, economics, and STS (Science and Technology Studies), the last being something that I hadn't even known existed. I also felt that, to teach the class without hypocrisy, I should be an activist, a vegan, and someone who lives an exemplary personal life. I felt overwhelmed by what I should know and who I should be.

I wanted a class that would change people, that would help students, and me, to become better human beings.

Con Diaz: Could you like to tell me about your 2015 IACR Distinguished Lecture, *The Moral Character of Cryptographic Work*?

Rogaway: The invitation for that talk came shortly after the first Snowden revelations. I thought the Snowden revelations huge, and that the cryptographic community should be ashamed by what they showed. But my fellow cryptographers didn't feel this way. Most didn't even see the Snowden revelations as professionally relevant. So when the invitation for that lecture came, I knew, in broad strokes, what I wanted to do: to provide a response to the Snowden revelations.

For about a year I just read. I learned about Surveillance Studies, another field I hadn't know of. I read classics like *Discipline and Punish*, and I read about the

history of the NSA, including the books by James Bamford. Then I read historical accounts of scientists' involvement in trying to redirect technology, like Matthew Wisnioski's *Engineers for Change*. Finally, I began to write. I did so daily, for about six months, at UCD's Health Sciences Library. It was emotionally draining. I sometimes said that the essay felt like it was being written with my blood.

It shouldn't have felt that way, in the sense that the essay's main claims ought to be obvious and non-controversial. I said that scientists and engineers have social responsibilities beyond our basic responsibilities as human beings. I said that cryptography is inherently political, as it impacts who has what power. I said that mass surveillance is a threat to democracy and human autonomy. I said that the Snowden revelations demonstrated that cryptography, which is supposed to be about private communication, had failed to help ordinary people maintain even a modicum of privacy when they communicate electronically. I said that cryptographers should care about and rectify this failure. Simple claims, yet few people were saying them.

Con Diaz: Wrapping up, when you reflect on your career, what are some frustrations and some moments of pride that come to mind?

Rogaway: While I like most of my technical work, it is that *Moral Character* essay that I feel happiest about, as well as my ethics class.

When I wrote the *Moral Character* essay, I was keenly aware that most people in my community would think it foolish to spend so much time on something that I didn't even intend to publish. But I never felt more like I was doing that which I was meant to do.

A similar feeling would often arise after teaching a class. I liked that I could teach both technical topics and something more in the humanities. I was proud of the ethics class, which changed the way that many students saw the world. They would tell me so. I especially liked the last group of students whom I taught, in the Spring of 2024. It felt like they represented all the students that came before, over all the years. I think they even intended that, in some way, as a gift.

As for frustrations, there have been too many. I wanted my department to be a place where the faculty cared deeply about our students, the intellectual value of our work, and the social consequences of it. But as time went on, it seemed like faculty cared less and less about those things. Most recent hires seemed like careerists. They want to be seen as "successful," but they don't like to teach, don't care who funds them, and don't interrogate the social value of their work. They

don't even care if their work is intellectually significant. Year after year I would implore my colleagues to stop recruiting in hot areas and to pay more attention to candidates' concern, or lack of concern, for the societal impacts of CS. I lost this fight so many times that I eventually felt I had to withdraw from faculty recruiting.

I was similarly unable to nudge the character of the cryptographic community. Essays and nontechnical talks never moved the needle. In fact, things got steadily worse. Few cryptographers seemed to care if their work helped people, harmed people, or was irrelevant. Lots of colleagues aimed to get rich with fanciful stories about cryptocurrency and blockchains. Did they themselves believe it?

Of course there's a happier point of view: that I accomplished all that was realistic. Often, I expect, it's impossible to shift the culture of an academic department. And shifting the disciplinary culture of an entire research area? Maybe that was never in the cards.

The last few years I came to feel increasingly out of place. It felt as though my values were rooted in another era. Most of what people were publishing felt boring and pointless. What departmental colleagues wrote in emails often seemed entitled, uncollegial, or self-serving. Only teaching remained pleasant. So I focused on that, and stopped working like a maniac. I learned to return to my wife before the crows overflew our Davis home. I spent much of my time taking our son climbing, the thing he loves most in the world. It was okay, I figured, this too may have value. Probably I had worked far too hard for far too long.

Con Diaz: Thank you, Phil.

Rogaway: Thank you, Gerardo. It's been a pleasure talking to you.

---

[1] Human playing X, the player who moves first, and computer playing O. Algorithm: **(1)** If there's an immediate win for O (8x3 cases), take it. (2) If there's a move needed to keep X from immediately winning (8x3 cases), take it. (3) Six special cases: 19 → 2; 37→ 2; 38→ 2; 46→7; 67→ 9; 68→9. (4) Occupy the first empty cell of: 513792468. Cells are numbered 123//456//789 and a rule like 19→ 2 means that if cell 1 is X and cell 9 is X and cell 2 is unoccupied then move to cell 2.

[2] Prof. Chernoff was clearly depressed when my girlfriend and I took his class. I remember him as gentle, lonely, and obese, but speaking with almost super-human clarity. Paul Chernoff died in 2017.

[3] You have a pile of *N* pennies and wish to flip them, one after another, to get *K* of them heads up. You can either flip a new coin or clear away all the flipped coins and start over. Find the optimal strategy.

[4] IP is the class of languages that admit *interactive proofs*. One of these proofs is *zero-knowledge* if proofs of membership convey nothing else of significance.

[5] Bill Bryant: Designing an Authentication System: a Dialogue in Four Scenes. Manuscript, February 8, 1988.
[6] At the time, the only anticipated goal for encryption was privacy, in the sense of indistinguishability / semantic security. From a modern perspective, *authenticated* encryption would have been a fine starting point, but this wasn't yet a notion.
[7] Mihir Bellare and Phillip Rogaway: Entity Authentication and Key Distribution. *CRYPTO 1993.*
[8] Catherine Meadows: Formal Verification of Cryptographic Protocols: A Survey. *ASIACRYPT 1994.*
[9] IND-CPA stands for indistinguishability under an adaptive chosen-plaintext attack.
[10] Authenticated Encryption with Associated Data. The user provides a plaintext, nonce, AD, and key, and gets, deterministically, a ciphertext that guarantees both the privacy and authenticity of the plaintext and AD. Classically, the user would have provided the plaintext and key, and gotten, probabilistically, a ciphertext that would have guaranteed only the privacy of the plaintext.
[11] Accreditation Board for Engineering and Technology (ABET), the often maligned body that accredits undergraduate majors with the word "engineering" in them.