Oral History of Ronald L. Rivest

Interviewed by:
Roy Levin

Recorded December 6, 2016
Mountain View, CA

CHM Reference number: X8019.2017

Levin: My name is Roy Levin, today is December 6th, 2016, and I'm at the Computer History Museum in Mountain View, California, where I will be interviewing Ron Rivest for the ACM's Turing Award Winners Project. Good afternoon Ron, and thanks for taking the time to speak with me today.

Rivest: Good afternoon.

Levin: I want to start by talking about your background, and how you got into computing. When did you first get interested in computing?

Rivest: Interesting story. So I grew up in Schenectady, New York, where my dad was an electrical engineer working at GE Research Labs, and in a suburb called Niskayuna, which is sort of a high-tech suburb of Schenectady. And they had a lot of interesting classes, both within the school and after school, and one of the classes I took, probably as a junior, was a computer programming class, back in about 1964, from a fellow that worked at the GE Research Labs -- I think his name was Marv Allison, who taught it. And he had a programming language he'd invented, and wanted to teach to the local high school students. And so he would come after high school, and we would sit around and hear about computer programming, and get these sheets, you know, with the marked squares for the letters, and write a program. And he would take them aw-- it was once a week, and he would take the sheets away at the end of the class. And a week later he would have them punched up, and get the outputs out, and of course there were bugs. So it took forever just to write a little program that computed the area of a triangle or something, as I remember. But that was my first introduction to computer programming.

Levin: A-ha, that's great. And about how many kids were involved in that?

Rivest: It was probably about a dozen, it was..

Levin: Great, great.

Rivest: I don't remember the name of the language. It was some little funky language he invented.

Levin: I'm sure. And did you guys work together at all on those things, or were they solo projects?

Rivest: They were all individual projects. Everybody had the same assignment, but it was a solo project.

Levin: I see.

Rivest: The one I remember was computing the area of a triangle, given the three sides, or something like that.

Levin: Uh-huh. And you don't remember how many weeks it took to get that done?

Rivest: It was probably about six weeks to get that done, yeah.

<laughter>

Levin: That's funny. So that's your sort of earliest involvement with computing…

Rivest: Yes.

Levin: …as roughly a junior in high school.

Rivest: Yeah, yeah.

Levin: Uh-huh.

Rivest: Yeah, I was sort of interested in math and science at the time. And computers were starting to become.. more known, you know, according to the popular press, but I didn't have any involvement with them personally until then.

Levin: Uh-huh. And were there other things that happened while you were in high school that were…

Rivest: Not that I remember. It was just a strong math program, which is of course good for a computer science background. But in terms of actual involvement with computers, I don't remember anything else.

Levin: Mm-hmm. And then you went off to college, you went to Yale as an undergraduate.

Rivest: Yes. Yeah, so in '65 I graduated from high school, went to Yale as an undergraduate. Wasn't sure at the time what I wanted to major in. So freshman ye-- Yale's a good liberal arts school, and I didn't know whether I wanted to do law, or psychology, or mathematics. I eventually decided on mathematics, it gave me the most freedom to explore other things too. But there was no computer science department in

Yale at the time. And so I took some CS lite classes from the engineering school, but got a degree in mathematics.

Levin: And did you think that you would continue to be involved with computing, or that mathematics was going to be your career at that point? Or maybe you didn't know what you were going to do?

Rivest: I wasn't sure what I wanted to do, although I very much enjoyed the work I did with computers. I supported myself part-time, working for Professor Richard Ruggles of the Economics Department there. And did some programming for the Econometric Society, they were computing price indices in South America and things like this. So I was going to the Yale computer center, and dealing with decks of cards to compute prices indices here and there, and that was sort of fun.

Levin: Oh, so you had to punch your own cards in those days <laughs>.

Rivest: You had to punch your own cards in those days, yeah. And there were also some good computer classes, there was a class on the MAD programming language, or the Michigan Algorithm Decoder, which was fun.

Levin: Interesting. So as an undergraduate, computing was sort of a sideline, you were focused on math. Did you have any particular areas within math that particularly interested you, or turned you on?

Rivest: So I think the math was actually sort of drab, compared to the CS stuff that I was doing. I enjoyed the CS stuff more. There were classes on how to build a computer, how to program computers. The math classes were more analysis, and topology and things like that, which I found.. frankly less interesting. Which is probably why I didn't go into math as a career longer term. I found the tangibility of computers, the ability to program them, to get them to do things, more exciting.

Levin: Going back for a moment to pre-college. Did you like to build things as a kid? Or was it.. maybe not the same kind of building, but..

Rivest: I don't know. I had the usual assortment of things that kids have, you know, chemistry sets and things like that, building electromagnets and whatever. But nothing computer-like. So I wasn't a builder in the sense of building treehouses, my brother specialized in that.

Levin: <laughs> So I guess it would be interesting to-- if you can reflect for a minute on what it was about building computer programs, or in general using computers, that you found intriguing or stimulating at that time.

Rivest: Yeah, I guess it was the logic involved, you know. Just sort of thinking through the-- sort of simulating in your head what the computer would do, thinking about the logic of the control, some of the combinatorics. It was the style of math and thinking that I enjoyed.

Levin: Okay, okay. So after Yale, you went on to get an advanced degree. Was that the next year, or was there a break?

Rivest: Yeah, when I finished at Yale in ’69, then I moved directly to the Stanford graduate program. So I entered the Computer Science Department PhD program there. Which was a new program at the time, it had only been founded I think in ’65, so there’d only been a few graduates through a PhD at the time. It was just getting going, they were still hiring faculty and so on too. But there was a great set of faculty there, Bob Floyd was my PhD advisor, Don Knuth was there, Zohar Manna. I also worked a lot with David Klarner and Vasek Chvátal, who were visiting and doing-- they were visiting mathematicians actually, doing a lot of combinatorics, and I worked with them. And a lot of great students: Bob Tarjan, Vaughan Pratt, a lot of other people were there at the same time I was. So it was an exciting department at the time.

Levin: Kind of a who’s-who of computing in some ways.

Rivest: Yeah, in some ways.

Levin: How was it that Bob Floyd became your advisor? Do you remember?

Rivest: I don’t remember how that evolved actually, that’s a great question. I remember very much liking the algorithms course he taught. He was teaching heapsort and other things, and he just had a beautiful way of explaining things. So I enjoyed his teaching style, and that must have drifted into an advisory thing somehow. But I don’t actually recall how that happened.

Levin: Okay. It was a fairly small department, roughly how many students would you say?

Rivest: At the time there might have been 40 students, I don’t know.

Levin: So you knew everybody, pretty much.

Rivest: Everybody pretty much, yeah.

Levin: Yeah, yeah. And did you actually end up working with Bob, other than on your thesis? That is, were you involved in research work that he was doing?

Rivest: So I did research other than.. with Bob. One of the early pieces I did with him, was the linear time median-finding algorithm, with Bob and Manny Blum, and Vaughan Pratt and Bob Tarjan. That actually arose from a conversation Bob Floyd had had with Manny Blum, at Berkeley. I never met Manny, at the time the paper was published even.

Levin: <laughs>

Rivest: But he said, "Here's the initial idea." And he had some nonlinear ideas, and we sort of refined them and pushed them a little further, made them linear. So that was one piece of research I did at Stanford. I also worked on some enumeration questions with Klarner, enumerating polynominoes, how many polynominoes can you make with n tiles? What are the asymptotics of that? I also worked quite a bit up at the Stanford AI lab, up in the hills behind the campus. In part because it was interesting work, and in part because it was DARPA funded and came with a deferment. And as you know, the Vietnam War was..

Levin: Right.

Rivest: …underway then, and so that's what allowed me to defer thinking about that issue. So I worked on the Stanford Cart Project when I was there, and worked with Bruce Baumgart. And we were trying to get the cart to drive around the parking lot, without hitting anything. So it's interesting resonance with the things that are happening today, with self-driving vehicles and so on too. At the time, the computers were so underpowered and our technology was so feeble, we couldn't succeed at what we were trying at the time. But we tried.

Levin: <laughs> So that was really-- that list of names you rattled off a minute ago was really pretty impressive. I mean, think of how many Turing Award winners there are in that group now. I mean..

Rivest: It was early days, and it was fun times, yeah, some good people.

Levin: Yeah, that's really quite something. Did you have the opportunity to work with Don Knuth?

Rivest: So I talked with Don Knuth off and on. In particular my thesis work on.. search algorithms for associative search. And had had some very helpful remarks there. But he was not my formal advisor, he was just somebody I talked with now and then. We also talked-- I also talked with him about some of his

exercises. I remember I had a solution to one of this exercises, and he said, “Aha, yeah, that’s gonna go in the book.” So it did.

Levin: <laughs> Well those books were kind of all the rage at that time. He was..

Rivest: Yeah. They’re still..

Levin: …actively involved in that.

Rivest: …very much valuable references, yes.

Levin: Yes, indeed. Although it’s still amazing to look back at what his original plan was for those books, and how ambitious it was. And I guess in some ways maybe naïve.

Rivest: I think he-- well yeah, the field was exploding in ways he didn’t expect, I guess.

Levin: Yes.

Rivest: But I think he’s still working on them, right? He’s had some recent folios come out recently. So the plan is still there, and is still being acted on. Although I think he’s probably mellowed a bit in his expectations of..

Levin: I would hope so.

Rivest: …covering everything, yes <laughs>.

Levin: I would hope so. So your work-- your thesis work, as you mentioned, was an analysis of algorithms. And of course Bob had this class, and Don was interested in those things as well. So it sounds like you gravitated to that area relatively early on.

Rivest: It was an area I enjoyed, the analysis of algorithms, it was sort of concrete. You can take a problem-- as you know, you just sort of take a problem, try to devise a good algorithm for it, and figure out what the analysis is. My thesis in particular was on algorithms for associative search. So you’re given a set of binary words, and you’re given a search query, which is a partially-specified binary word. And you efficiently want to find all the words that match.. and so on too.

Levin: Mm-hmm. The whole area of.. I guess what gets labeled “analysis of algorithms” is one that started out very concrete, and then kind of mushroomed out into more theoretical issues, and eventually I guess into complexity theory in some ways. But where would you place yourself sort of on the spectrum there, of different kinds of analysis and so on?

Rivest: So I very much like the concrete analysis. I mean the complexity theory issues I’ve thought about, and did some work with Andy Yao, on k-headed automata and things like that. But my preference is much more for concrete algorithms that work on problems that people care about. And so trying to take a problem that has some real-world impact, and find a good algorithm for it.

Levin: And would you include probabilistic analysis in that general sphere as well?

Rivest: I don’t do so much of that, I do a little bit now. Now more recently I’m thinking more about probabilistic algorithms for statistical approaches, in some of the work I do on voting and so on too. But back then I didn’t do too much of that. Yeah.

Levin: Uh-huh, okay. So.. you graduated from Stanford in ’73?

Rivest: ’73, yeah.

Levin: And then you did a postdoc.

Rivest: Then I did a postdoc. So then I went to Paris for a year. So two of my colleagues, Gilles Kahn and Jean Vuillemin, were at Stanford at the time as well, and they said, “Ron, when you’re finished with your PhD, why don’t you come work with us at INRIA?” Which was just north of Versailles at the time, and outside of Paris. And I said, “Sure, why not?” And I knew a little bit of French, I’d had three years of high school, and a couple of classes in college and so on too. So I figured I could probably cope with the French. And my wife and I said, “Yeah, this’ll be fun for a year, so we’ll go do that.” And I went there-- the one thing that was a big surprise to me was that the working language was in fact French, at the lab. I had expected English for some reason, we never talked about that. So that was totally exhausting and draining for the first three months..

Levin: I’m sure.

Rivest: …to get used to that. But eventually I managed to work there. And we worked on a variety algorithmic questions there, lower bounds. I worked with Jean Vuillemin on the Aanderaa-Rosenberg

conjecture and things like this, so it was a productive time. Also just a great time to have a postdoc in Paris.

Levin: Yeah, I can believe that. That would be a great time. Was the style of work, as it went on at INRIA, similar or different to what you had experienced as a graduate student?

Rivest: It was.. I guess somewhat different. There was no AI component there at all, I meant it was all theory really. And it was more formal ling-- in formal languages, kind of thing. So there was a lot of work that was more theoretical than the sci lite [ph?] scene, it was less algorithmic. But it was.. by and large it was fairly similar. And it was a good team of people to talk to, I learned a lot of things about different styles of computer science there.

Levin: Were there distinct labs or groups that were within the computing aspect of the department there?

Rivest: Yeah, there were different buildings, Bâtiment Huit was the one I was in, building eight. So that was more theory and algorithms. And there were a number of other buildings, which I don't know what went on in all of them.

Levin: Were there-- roughly how many people? I'm just trying to get a sense of the size of the group, and what it was like.

Rivest: The theory group there might have been.. 15 people or so at the time.

Levin: Mm-hmm. Yeah, good bunch, good bunch.

Rivest: Yeah, it was a good bunch.

Levin: And were there other postdocs there at the same time?

Rivest: I don't recall that there were. I think it was mostly local French working there.

Levin: Mm-hmm.

Rivest: Yeah.

Levin: Okay. And then after your postdoc you went back-- came back to the US, and went to MIT, right?

Rivest: Yeah. So it was interesting searching for a job while I was a postdoc in Paris. So I sort of arranged one big tour of the country, looking at a variety of places, trying to figure out what to do next. And I talked to.. Carnegie Mellon and MIT, and a bunch of other places, Sandia Laboratories and so on, and tried to figure out where to go. And decided upon MIT. So when the postdoc was over in the fall of '74, I joined the MIT faculty in the Computer Science Department, well the.. EECS Department, as part of the Lab for Computer Science. Actually, it wasn't called the Lab for Computer Science quite then, it was Project MAC at the time still.

Levin: It was still Project MAC in those days, yes.

Rivest: And I think it was the first year I was there maybe, it changed over to Lab for Computer Science. And now it's the Computer Science and AI Lab, they've changed their name again.

Levin: Every couple of decades, right?

Rivest: Yes, yes.

Levin: And roughly how big was, well, Project MAC at that time?

Rivest: I don't recall how many people there were. It was in a building in Tech Square that had other tenants in it. So it was probably about three or four floors of the building, each of which might have had.. 60 people in it or something. So that's maybe.. a couple hundred people, max. It wasn't my job to know these numbers at the time, I don't.. <laughs>

Levin: No, just a.. sense of what the organization was like.

Rivest: Yeah, I mean we were just..

Levin: It sounds like it was a lot bigger than Stanford was.

Rivest: The CIA was-- yeah, a lot-- well yeah, it was bigger in some ways than Stanford, yeah. The CIA was on one floor of the building, Xerox had some lawyers on another floor of the building, it was a mixed-use building. So it was-- and the theory group occupied the major portion of one of the floors. And so

Albert Meyer, and Mike Fischer, and Vaughan Pratt, and a lot of other people were part of the theory group at the time.

Levin: And when you came to MIT, other than teaching, which I'm sure they immediately put you to work doing, what was your research focus at that point?

Rivest: So I started off writing proposals to NSF on algorithms in general. And in fa-- well I guess originally there was an umbrella grant that Albert Meyer had that covered a lot of the work on theory, a lot of algorithms. And I was intrigued by P = NP, and circuit complexity and things like that. I was wondering whether you could prove that P was different than NP, by proving circuit lower bounds, which is still the dream of lots of computer scientists -- it just hasn't happened yet. And we've learned a lot about that style of argument, but it just hasn't worked yet. We know better why certain kinds of arguments won't work. So that was some of the things I was thinking about. And.. it wasn't too long before I got interested in some of the crypto stuff, that happened a little later but not right away. I was teaching, busy doing that, teaching algorithms courses and learning how to teach the first couple of years.

Levin: Was it mostly individual work, and work perhaps with students? Or were you collaborating with other members of the faculty at that time?

Rivest: So I started supervising theses right away, and talking about string searching algorithms and other things with students, and supervising theses. Collaboration with other faculty? There was a lot of group discussion, theory group had a-- and has always had a great sort of spirit to it. And a lot of people collaborating together over a variety topics. And we've grown, it's a much bigger group now than it was. But even then there was a.. great esprit de corps, and people talking about common problems.

Levin: And that led, I would guess pretty quickly, to the work that you ended up having quite a long time being involved with, namely the crypto work.

Rivest: Yeah, the crypto work started in '76, when the Diffie-Hellman paper was published.

Levin: Right.

Rivest: And so we-- I had a graduate student at the time, Steve Boyack, who showed me this paper by Whit and Marty and said, "Hey Ron, you might find this interesting." And so I did. We had been talking-- Steve and I had been talking about his work, which was on crypto-related things, sort of one-way function-like objects in the matrix space. A sort of, you know, what matrices over GF2 are there that are easier to compete one way than another? And so he sort of had a crypto flavor to it. So he always had an ear out for those kinds of things, and saw the Diffie-Hellman paper and asked me to take a peek at it. And

I did, and said, “Oh, this is very interesting. And there’s a nice, open problem here. It’s really a beautiful paper, and lays out the ideas of public-key cryptography, but doesn’t have implementations.” And at the time Adi and Len were in offices next to mine... they were in the Math Department, Adi was coteaching with me in fact, an algorithms course. So we were-- I was seeing him a lot, and Len was also a friend and nearby in an office. So we collaborated a lot, and I said, “Hey guys, now this is an interesting problem, shall we talk about this a bit?” And Adi and I spent some time devising initial approaches, which Len was quick to show us didn’t work. And so we had quite a back-and-forth for a while, trying to come up with something that might meet the specs of Diffie and Hellman.

Levin: So it was very much motivated by the desire to make real, practical, this idea that had shown up in that famous paper? Now famous paper.

Rivest: Yeah, the paper itself laid out the arguments for wanting to do something like this. Both theoretically, saying, “Here’s the interesting questions, can you do these kinds of things? And practically, if you could, here’s the kinds of things you could do with them. And.. talk to your stockbroker privately with his public key, and things like that.”

Levin: <laughs>

Rivest: So it was a beautifully written paper, it was the motivation for our work. But it really wasn’t clear to us that you could do something like this, it was an open question. And we did actually get frustrated at times and say, “Well, maybe we can prove that this is impossible. That you can’t tell somebody how to encrypt, without thereby telling them how to decrypt as well.” But we failed at that, and ended up..

Levin: <laughs>

Rivest: …coming up with a proposal that still stands today.

Levin: Exactly. So you mentioned just a little while before that you had had some interest in P = NP. And that reminds me of the.. I guess pretty well-known picture of the three of you, Adi and you and Len, standing in an office with a blackboard behind you, on which is sort of prominently written, “P = NP.”

Rivest: Yeah, that was a joke..

Levin: Is there a story there?

Rivest: Yeah, there is a story there, yeah <laughs>. Somebody-- this was after we did the work on RSA, and so somebody wanted a picture for a paper. And so before the picture was taken, I think it was I that wrote the "P = -- therefore P = NP" on the board. And sort of a joke to those that were in the computer science field there.

Levin: <laughs>

Rivest: And the photographer of course didn't know anything. And actually very few people noticed it, and then would look at it and say, "Oh yes."

<laughter>

Levin: And somehow the argument that led to the "therefore" was hidden by your back.

Rivest: Yeah.

<laughter>

Rivest: But it was a joke, we didn't have any proofs then of course.

Levin: That's great. So the RSA paper appeared in early '78, do I recall correctly?

Rivest: So the first-- so the Diffie-Hellman paper appeared in '76, the invention of RSA and sort of the.. figuring out the details was '77. And at the time we were trying to assess, "Is it secure? Does it make sense? Does it work?" And that process continued towards later in '77, in particular with the publication by Martin Gardner of his column in Scientific American. He had this wonderful column called "Mathematical Games" which many computer scientists cut their teeth on. And that was sort of bread-and-butter reading for lots and lots of people. And he just-- we had contacted him to.. see what he knew about factoring. Because factoring was sort of this arcane subspecialty in mathematics, and not too many people actually did research in it. And so we thought he might have known some of the people that worked on these kinds of weird questions. And he got all excited about our proposal, and wanted to write a column about that, so he did. And that was really sort of the first publication of the paper, in a widely accessible manner. There was also a memo that we wrote for the Lab of Computer Science that described it. And that came out.. well, it was published earlier, but wasn't-- there was.. an issue with the distribution of that. And the crypto wars are still going on -- they started back then. -- and the crypto wars were the questions as to whether, you know, working in cryptography.. in academia was in the national interest or not, or should it be done? Or was it illegal? And there were laws that were waved about as

being possibly relevant, like the International Traffic in Arms Regulation, and things like this. So we were told that maybe there would be a violation of some law, if we were to ship this memo around. And so we had the MIT lawyers look that over, and finally in December of '77 they said, "It's okay to mail it out now." So we mailed out lots and lots of copies of this memo, to people had sent in self-addressed stamped envelopes, based on Martin Gardner's column. So that got out. So the publications, there was Martin's column and then the memo, were the first things. And then there was the Communications of the ACM article appeared in 1978.

Levin: Mm-hmm. And since this memo went out widely, was there any immediate impact?

Rivest: There was lots of interest, yeah. People were interested in this, there were some people who got interested in the implementations, the possible applications. It was tough to-- in spite of the wonderful setup that Diffie and Hellman had made, and then technology that seemed to work, it was tough to see how to make the transition to practice. Computers were very slow then. I mean if you think about a VAX back in the day, trying to find a large prime number, it really took quite a long time. Many minutes, maybe half an hour in some cases, depending on how big the prime was. And so the practical aspects of this were daunting at the time. Plus the integration with various systems and, you know, the Internet existed, but the web didn't exist. And so applications were sparse, so it took a while for some of this to settle in. Plus I think it takes a while for people to get their head around a new idea, and just sort of understand its implications, and how it might fit into their existing systems.

Levin: Had the three of you, and maybe some of your colleagues and people that you discussed it with, been thinking about concrete applications of that?

Rivest: Well I think the concrete applications, first of all were to some extent sketched out in the original Diffie-Hellman paper.

Levin: Yes, right.

Rivest: And I think talking to your stockbroker was one of them.

Levin: Mm-hmm.

Rivest: And so I think secure email, secure phones were the kind of things that.. struck us as being the most likely areas of first application.

Levin: But you obviously recognized that the sort of feeble computers you had available at the time, were going to make that a challenge, at least in the short run.

Rivest: Yes, yes. Yeah, implantation efficiency was definitely a concern. And one of the interesting things that happened about that time, was MIT got involved with VLSI. So VLSI at the time had been entirely a black art in industry, there was little academic research in VLSI design. And finally MIT said, "This is something we should do. There's enough interest and interesting questions here, and complexity, that we should take this in and make it a research area at MIT." So they brought in people from industry to teach us how to do VLSI design. And I glommed into this saying, "Yeah, we should learn this. We should figure out how to make an RSA chip." And so the technology at the time was really primitive. I remember taking a class where you would pull out these sheets of red.. cellophane, and you'd take your X-ACTO knife, and you'd cut out the little masks and you'd place them on. And those are used.. the photograph, to make the mask for the chip. It was just a horrible technology for doing anything of scale.

Levin: <laughs>

Rivest: And so we took the class but said, "If we're going to do an RSA chip, we're not going to do it that way." And we continued to explore making an RSA chip, and we did, but instead we wrote Lisp programs that generated the masks. And so we used the Lips machine that was there to produce these masks, and they were all computer-generated and simulated on the computer.

Levin: One forgets..

Rivest: Yeah <laughs>.

Levin: …how primitive the technologies were in those times. So I guess it's natural to ask, when did practical use of RSA actually begin?

Rivest: Practical use, that's a great question. When did-- now the first practic-- I mean we tested it certainly writing messages to each other, and doing things like that. But in terms of commercial applications.. so, I guess the practice followed a path that I guess many companies' applications do. we started off with the idea, and then MIT decided to file for a patent. So they said, "This looks like it might have practical application." And I'm not quite sure how they made that assessment, but they just said, "Yeah, this looks interesting enough to file a patent for." And so there was a patent filed. And that wasn't granted until 1983 or something like that. There was some interference with the Stanford patents, on their working too, that took a little while to resolve. But that all got sorted out eventually. And then in about '83, we decided that this looked like it was commercially viable. We weren't quite sure how, and we didn't know much about business, but we decided to set up a company to explore that. And so we did, we created RSA Data Security. And Len was the first president of the company, even though he, like us, Adi

and I, knew nothing about business really. And we started to explore applications. And the first thing that we had in mind there was secure phones. So that was not.. a market which we knew a whole lot about, but we said, “We can make this work.” And so it involved some.. encoding and decoding of voice, with a crypto and so on too. And that turned out not to be where the market really was, that didn’t work. We had some difficulty with the supplier of the chips that we wanted to make, which were based on the original RSA chip we did. So the chip fabrication route didn’t work. But as that was happening, the whole world of general-purpose computing and software was taking off too. We weren’t quite to the era of the web yet, but the Internet was starting to take off. And the idea of encrypting products, encrypting messages on the Internet, started to seem more and more real. So in ’86 the company about went bust, we weren’t making any money. We got Jim Bidzos to take over the company, and we changed the management structure. And he was a really great, insightful, and smart businessman, who had a lot of technical depth and understood how to close a deal, and started talking with companies that might do it. I think one of their early licensees was IBM-- Lotus Notes was one of the first products that got signed up -- but it was difficult, because there weren’t too many products yet that required distributed security.. in a nonmilitary kind of context. So..

Levin: Let’s go back just for a minute, to the secure phones idea..

Rivest: Yeah.

Levin: …which didn’t work out. But the concept of a secure phone already existed at that time? The military was using..

Rivest: I’m sure they had, and we didn’t have much contact with the military..

Levin: …just conventional encryption, but not public-key.

Rivest: Yes, yes. And key management.. key management got-- I mean the whole point of public-key cryptography of course, is key management..

Levin: Yes.

Rivest: …simplification. And so you could exchange public keys, possibly having them signed by a central authority, and establish keys in a nice way that way. So this would simplify those kinds of things. We did not talk with.. manufacturers of secure phones at the time, that I recall. We were just going our own route, and trying to do that.

Levin: Mm-hmm.

Rivest: And there were other things too. I remember talking with one company -- Delco I think it was -- that wanted to put RSA into door locks for cars. And they said, “Well this is going to work fine for that.” I wasn’t sure the application was a great fit, but they said, “Well if you can make it for under 10 cents, we’ll do it.”

Levin: <laughs>

Rivest: As usual with the car industry, everything has to be really, really cheap. We said, “There’s no way that’s going to happen.”

Levin: That was probably an easy question to answer.

<laughter>

Rivest: Yeah. So that didn’t go anywhere. Yeah.

Levin: And so this is right around the mid-80s, or maybe getting a..

Rivest: The mid-80s, yeah.

Levin: …little later. So we now.. we’re sort of in the era when microprocessors were beginning to come to be readily available, or at least..

Rivest: Yes, yes. So things were getting more and more feasible to do on general-purpose microprocessors. In the early days, again, the implementation expense was large. I remember the prime-finding issue, one letter I got was a guy who was setting up a business to sell prime numbers, because he thought they were hard to generate. So of course, buying your crypto key from somebody else makes no sense whatsoever.

Levin: <laughs>

Rivest: But this is what he was doing. And he had separate prices for two-digit primes, three-digit primes, and four-digit primes. You know..

<laughter>

Levin: Well, all sorts of people out there with strange business ideas.

Rivest: Yes <laughs>.

Levin: That's amazing. But even with the CPUs of the late 80s, you would have needed a dedicated chip to do the encryption, right?

Rivest: Not necessarily. You could do so-- well, I mean.. the encryption.. is basically a singular modular exponentiation. And so that's.. whatever it took, it might have been a minute or so, something like that. Finding the keys was always a longer operation, to finding the prime numbers and so on. But that only needs to be once, so that was not so much of an issue. But computational efficiency, I mean it's amazing how much faster computers have gotten since those days.

Levin: <laughs> Yes, indeed.

Rivest: Like what you can do with your laptop now, and what was possible back then. And so now.. encryption is-- public-key encryption is doable, and it's blazingly fast, in a tiny fraction of a second. But then it was an issue, and so we've seen the evolution of the computing power.. have its impact on public-key cryptography. And I think this will continue, we'll have.. these schemes now that seem complicated, will become easier to implement as we get more tightly integrated computer systems. And maybe Moore's Law is over, but there's still lots of gains that can be made in implementing crypto in special purpose ways, for some of these newer ideas that have come on the scene.

Levin: That's a topic I want to come back to a little later.

Rivest: Sure.

Levin: But at the moment, I'd like to stay with the chronology roughly.

Rivest: Yep.

Levin: Make sure I understand how that unfolded. So you mentioned that MIT had filed for a patent..

Rivest: Yes.

Levin: …around the time of the original work, I guess the end of ’77 or so. And the patent didn’t issue until ’83.

Rivest: ’83, yeah.

Levin: Which was also about the time that you guys started the company. Was that coincidental?

Rivest: No, not entirely. I think the-- I’m not sure there was a causal relationship there, but it seemed it was synergistic. The company was founded, and obtained an exclusive license from MIT for the patent. And..

Levin: I see.

Rivest: …so it was one of the bases for going-- then being able to go out and raise some funding, saying, you know, “We’ve got some assets that we can.. exploit, and try to move the market forward.” So..

Levin: And MIT was okay with the idea of having these three professors go off and, at least part-time, be working at this company as well? Or..

Rivest: MIT has always been very favorably disposed towards entrepreneurial activities on the part of its faculty. I mean it’s a clear.. line, they say, “You know, your primary obligation is to MIT. But if you’re doing some amount of time on the side, doing bus--“ And it definitely didn’t take a lot of time at the early days, because there wasn’t much happening at first. There was no market really, quite. And so it was trying to figure out..

Levin: Right.

Rivest: …what to do. But MIT does have policies of supporting entrepreneurial activities, on the part of the faculty. Faculty typically take time off, and go off and do a startup or something like that, for a year or two. Usually not more-- I think more than two is not allowed, but up to two years. In fact, I didn’t take any time off to work on this, it was.. lightweight enough in the early days that it wasn’t necessary. And after that we had a good team at the company itself.

Levin: It’s very interesting to hear-- I mean we’re talking about more than 30 years ago now, and to hear how that compares with perhaps the way things work today. First of all, five years from patent filing to corporate creation, would be an eternity by today’s standards.

Rivest: Yes, yes <laughs>.

Levin: But that was okay then, because there basically was no way to implement.. for what you were just saying.

Rivest: Yeah, we were way ahead both of the implementation -- good implementations, in terms of the hardware being available -- and the market. The market didn't really happen until the web happened. And so it wasn't until the 90s till things really started opening up.

Levin: So you were exploring—in the context of the company, you were exploring both making concrete products, and licensing technology.

Rivest: Yes, yes.

Levin: And it sounds like..

Rivest: Yeah, the focus ended up being on the licensing primarily. So building up software libraries, implementing the crypto, turned out to be one of the major business sides of RSA then. Rather than the chip, the chip didn't go anywhere.

Levin: Mm-hmm.

Rivest: But the software packages did, they turned out to be essentially what was needed for IBM, or Mozilla or other companies, Netscape, to put into their product, and make these things work. So..

Levin: Mm-hmm. And given that the work had been done in '77, and this memo that you talked about had gotten widely circulated shortly after that, even though the patent didn't issue for a long time. Were you concerned at all about being scooped by somebody else?

Rivest: There's lots of things to be concerned about, I mean there's lots of risks with trying to push out a technology. There's the risk that a) the technology isn't what you thought it was, that it turns out to be insecure. It could be that somebody develops a great factoring algorithm and so the technology just goes poof. It could be that the implementation is such a barrier that it's not going to be there. You can be scooped in a couple of ways. We could be scooped for better algorithms. That turned out not to be the case, at least not for a long time. Nowadays, there's interesting alternatives based on elliptic curves and so on too, that people can use, but I wouldn't necessarily call them better, but they're certainly very interesting, and there's lots of reasons-- lots of risks you can have, plus just the complexity of trying to

explain these things to people with any kind of technology. You've got to describe how does key management work? How do you-- how do keys get from A to B in a trustworthy way? How do you know that you've got the right key?

Levin: Sounds like you were more concerned about the actual technical-- inherent technical risks of it than competitive risk.

Rivest: Yeah, there were no real competitors from a business sense. It was-- but there wasn't much of a market either, so there wasn't much to compete over yet. It was trying-- it was market creation, if anything, that was the most important part of this, trying to explain to people what this technology can do, and how they could use it and to build up a demand for it.

Levin: So you stayed involved with the company for a number of years -- the early days -- but then even after the management got-- maybe we should say -- regularized a little bit, you remained involved; is that right?

Rivest: I remained involved as a consultant and an advisor, board member, for a number-- you know, for a long time and sort of tried to work with-- mostly with Jim Bidzos, who was running the company at the time, until the company was sold in '96, so I was involved with the various aspects of licensing and worked with various customers and working on the technology, trying to make it better.

Levin: And as you reflect back on that, that was your first sort-of firsthand business experience, I take it.

Rivest: Yes, yeah.

Levin: What do you think are the main things you learned from that?

Rivest: Just the complexities that you had to deal with, all the huge array of-- from legal to business to technical to funding to managing employees, to everything -- just the range of aspects that a business involves. It's not just, you know, doing the math.

Levin: Right, that's the easy part, right?

Rivest: Yeah.

Levin: Of those things, other than doing the math, what was sort of the most fun and what part did you not like very much?

Rivest: I tend not to like situations involving conflict between board members and stuff like that, or whatever that-- which we had some of, but, you know, I think the-- I actually enjoyed the variety of issues that we had to deal with, like-- as I said, I went to a liberal arts school. You get to study a broad variety of things, I like situations that have a diversity of aspects to them. I guess some of my more recent work on voting is that flavor too, where you got a variety of things you have to think about, and doing a startup is sort of like that. You have to think about a lot of-- many different things and try to make them all synergize well, so that was part of it. If you have good people-- I mean, working with good people is always a lot of fun, and if the market's just taking off, that's a lot of fun. Got some fun technology, that's a lot of fun, so...

Levin: How big did the company end up being in people?

Rivest: Well, it grew and grew. It still exists. At the time-- it was small until the Web happened. It was only a handful of people. We would have, I remember in the early days, people come to our offices in Redwood City and was a small office with, you know, 10 or 20 people in it and they say, "Oh, this is a nice branch office. Where's your main office?" <chuckles> This is it, so it was small for quite a while. Was really-- I mean, there was no-- not much revenue for a while and we were living on our investments and the small amount of revenue we got until we had some real market penetration later on.

Levin: I guess one thing that-- I wanna move on to talking about some other topics, but the last one I want to talk about in this area was that in the late '90s, I think '97, if I got that right, it became known that work on a very similar algorithm had been done by Clifford Cox in the UK, but working for GCHQ and it was of course all classified.

Rivest: That's correct, yeah, yeah, so I'm not sure when they announced-- might've been '97 or '99. Trying to remember the exact...

Levin: But anyway, a long time after your work.

Rivest: Yeah, so the announcement came long after-- they announced that they had invented, in secret, in their work in GCHQ, the idea of public key cryptography. They called it nonsecret encryption, something close to RSA. It wasn't quite the same. They had the modulus both being used as the modulus and the exponent or something. Clifford Cox-- but it's essentially the same idea, and also something like the Diffie-Hellman key exchange idea as well, so a number of things. As far as I know, they never did anything with any of those things. They wrote them up, so these are interesting, put them back in the drawer, and never worked on exploiting them either commercially or in the military space, as

far as I know. I don't have a security clearance, so I'm not sure what I know is the full story, but they, as far as I know, didn't do that, and it might've been because the military needs are different than commercial needs. I mean, in a more hierarchical situation, a need for public key is perhaps less than in a more free-floating kind of commercial scene. Might've been the implementation difficulties they found daunting. I don't know.

Levin: Well, if it was earlier than you and you found it daunting, imagine that probably was the case. Did you ever have the opportunity to talk with Clifford Cox?

Rivest: Recently, yes. I've met him a couple of times, yeah.

Levin: Tell me about that.

Rivest: Was a brief conversation. At the time, I think he was talking about his new ideas in identity-based encryption, so he's done some additional work since, but didn't talk in any great detail about the inventions that they had made too and I'm not sure to what extent he was able to talk at the time.

Levin: No hard feelings, though, as far as you could tell?

Rivest: Interesting work, yeah.

Levin: Yeah, it's a strange thing when, you know, priority in inventions somehow isn't discovered as a result of things like security issues.

Rivest: If they'd published at the time, it's not clear what would've happened. I think really having a company and things like-- doing things like that really made a difference. I mean, it took a lot of work to take these ideas and get them out there. The founding of the RSA conferences, for example, was a big step forward in trying to get these out, so a lot of things that had to happen in order for these ideas of public key cryptography and RSA algorithm to sort of take root and just having the ideas is just the first step, yeah.

Levin: So of course cryptography has a lot of applications and it's not necessarily an end in itself, but it's an enabler for a great number of other things and you've done quite a lot of work in computer security, including most recently, I guess, the work in voting, which we're gonna talk about in more detail later, but excluding that for the moment, what do you-- thinking back on the work that you've done in computer security that maybe took advantage of cryptography or maybe didn't, what do you think of as your significant work there?

Rivest: So most of my work in computer security stuff is-- there's things I've done that don't have any crypto in them at all or very little. For example, there's this work on the game called Flip It that I did with RSA Labs, which is sort of a stealth game. So you wanna know when you should change your passwords, so you model that and you say, "Well, if somebody's got the possibility of stealing my password, that's his action and I've got another action I can take asynchronously, where I can reset my password or change it," and so maybe when you're changing, you find out that it was stolen, say, and there's a question as to how you model this and how you mathematically-- so the methods of non-cryptographic kind of scenario, where you're modeling a game between an adversary like (in security, there's always an adversary of some sort) and the person with the password and you're trying to figure out how to model that and that's a-- again, a mathematically modeled situation. Game theory's the core of it, new kind of game theory, sort of continuous time, partial knowledge, and so it's one of the things that I think is one of the more interesting computer security kinds of aspects of it. I did a number of pieces of work, many of it-- much of it crypto based, with the folks at RSA Labs too, 'cause I continued to work with them on the security issues. Most of the other work I did was sort of more pure crypto or something at MIT, so work on-- some of it derives from the policy issues. For example-- and this continues today with the crypto wars 2.0. You know, should law enforcement have access to plaintext? And so on too. And we went through this debate back in the '90s and earlier, and there was the whole Clipper chip episode too, where the FBI, law enforcement, suggested that everybody should have their chip implementing the crypto and so some of the work I did on crypto was then-- I don't know if you call it-- I guess it's more crypto than computer security, say with Mihir Bellare, was this notion of "translucent" crypto, so you can have a situation where the law enforcement gets access to a certain fraction of the plaintext. Turn a dial, it controls the fraction, so you're trying to resolve a hard policy debate. Say, well, do they get 60 percent, 20 percent, 90 percent, 10 percent? Another dial you can play with, so some of these questions of hard policy questions you can try to turn into technical questions and say, "Is there another dimension at which you can play this game?"

Levin: Has any of that caught on?

Rivest: No.

Levin: Should it?

Rivest: The debate continues, but those particular ideas haven't caught on anywhere.

Levin: Coming back to FlipIt for a minute, 'cause I read that paper and I thought it was very intriguing, and it seemed to me that maybe a little ahead of its time, given that that work-- I mean, today, passwords are a major problem, right, for people?

Rivest: Yeah, yeah, it has application. We've had people talk to us about trying to apply it here and there. There's a lot more theory that could be done too. It's an interesting framework, sort of a new area terms of game theory, I think, that hasn't been well-explored yet, and so we've taken some initial steps at doing this and I think it's a fun problem and could have some impact down the road.

Levin: Anything else in computer security that you'd like to highlight?

Rivest: Probably not, probably not. I teach computer security and crypto both, but the computer security side of my work is probably smaller than the crypto side. We had this dichotomy -- sort of almost a two-cultures kind of situation -- with cryptography and computer security. Cryptographers like to live in this ideal world where Alice has a key, Bob has a key, they can keep their keys secret, they can work with those keys and encrypt mail and so on, and whatever. And so the fundamental assumptions that are made by the cryptographers tends to be this ideal world where people can generate secrets perfectly well, they can keep their secrets perfectly well, and they can use them without disclosing them through leakage and so on too. The real world is much messier than that in trying to actually generate good secrets on a computer, to keep them secret on the computer, and to use them without disclosing them inadvertently somehow is much harder than you think, so there's-- we're learning how to do that.

Levin: Some of those problems, of course, are nontechnical, right?

Rivest: Some of them are nontechnical. Many of them are technical. I mean, leakage questions. How do you implement implementation of RSA so that it doesn't leak information by the time that it takes to encrypt or the power that it uses during the encryption? Things like this are aspects of leakage we hadn't thought of at the time and now learning how to do better, so the instantiation of this ideal world into the real world, where you got real computers with real power usage in real time, things like this, is a-- and real questions of protection -- isolation of our processes so you can keep secrets -- is a hard one and so computer security has a lot to do yet to make crypto as usable as it should be.

Levin: We have seen some progress, though, right? I mean, we actually had machines that ship with secure computing base of some sort...

Rivest: Yes, we've gotten better.

Levin: ...become more the norm. How well it's used in the wild is perhaps open to question, but at least there's a flat rock to stand on.

Rivest: We have made progress, absolutely.

Levin: As I look through your publication list, I was struck by the fact that while there were obvious focuses, foci, of activity and technical area, there's a sort of continuous background of other topics in research that don't necessarily relate to crypto or security or, later, voting, and I was intrigued by the breadth of those. I found that you'd published papers over several decades in computer-aided design algorithms, for routing, computer architecture itself, a lot of work in machine learning, which goes back, I think, into the '80s, e-commerce. Talk about that space of things and how you're...

Rivest: I think to a algorithms person or a theory person, these are all sort of applied computer science, right, so you have situations where a little bit of mathematics and algorithm thinking can be useful. The work on machine learning was a big piece of work that I did. After I got tenure, I decided that maybe I should do something rather different and take some risks and do something quite a bit different than what I did before, so I decided that machine learning would be the thing that I would try. I had done, as I had said earlier, some work with the AI lab at Stanford as a graduate student. I had some interest in AI then. It's certainly one of the big challenges of the century to try to figure out how to make smart machines and so I said, "Well, I'll spend some time looking at that," and so those are driven by these fundamental philosophical questions, but technically, they end up being instantiated by questions involving particular classes of functions to be learned and so on too. Les Valiant had some seminal work on probably approximately correct learning little bit earlier, and that was a nice framework to work on, and learning about automata and stuff was some of the work that I'd done with some students, particularly Rob Schapire and so there's a lot of fascinating questions about taking a machine-learning scenario and trying to model it formally, trying to come up with good algorithms, and to prove they're effective that was-- that were of interest to me. I really much enjoyed working on those kinds of questions. I must say, though, that after a number of years of working on the machine-learning things, I found it to be somewhat less satisfying than the work on crypto because crypto has this wonderful flavor of bridging theory and practice so nicely. You can take a crypto application. You can design an algorithm for it, get the security pieces together, say, for an auction or something else, and then it immediately can be applied. Machine learning at the time was much more theoretical. It was much less applied, trying to find good data sets, trying to measure how well you were doing in any kind of application was tough at the time. It's changed a bit since then, but at the time, it was tough, and so I was a bit dissatisfied with it being sort of more theoretical work without any suitable practical foundation and trying to assess how well you were doing at the time was tough. So I drifted more back, and besides, the work on crypto and security never stopped quite either. I sort of always had my foot in that bit of work and then so I got pulled back into that more full-time as time went on, but machine learning work, I still have an interest in. It's interesting.

Levin: Is there a-- there's sort of a fundamental difference there too, also, that inherently in the world of crypto and what I might call more conventional algorithms, you actually get to prove things and it's a little harder to do that in the machine-learning world, where things are inherently probabilistic.

Rivest: The probabilistic, the metrics as to how well you're doing. The metrics that really matter are those-- you know, how does it work in the real world on these real data sets, and those real-world data

sets may not have good theoretical models, so results you prove about how well an algorithm works in a theoretical model may not have any relevance to what happens in practice, so yeah, it's tougher.

Levin: So maybe there's a sort of fundamental difference there in fact. Maybe we can digress for a minute on that topic, the notion that algorithms work as it was when you were, you know, a graduate student and a starting-out professor has sort of changed in the modern world where there's so much more probabilistic stuff; there's so much more...

Rivest: A lot more probabilistic stuff, that's correct, yeah.

Levin: ...so much more work in machine learning. How do we deal with that when we're trying to, in particular, teach students that getting things right-- well, what does it mean to get things right?

Rivest: Yeah, that's an issue that I'm struggling with actually right now. I'm planning to write a new chapter for our algorithms textbook on machine learning and it's exactly those questions that you've raised are-- it doesn't even fit within the classical algorithms textbook. Getting it right is not like it is for finding the shortest path. It's meaning finding a good hypothesis out of a hypothesis class that fits the data well and so on, so I think it fits, but it's a different level of abstraction.

Levin: Yeah, I think the evolution of what it means to be an algorithm has been rather interesting. I remember hearing a talk by John Hopcroft in which he basically said, "I started out my career doing graph algorithms and in those days, graph algorithms meant something with 10 nodes," and he said, "None of that is interesting anymore. The only things that are interesting are graphs with large numbers of nodes, millions of nodes, and the algorithms that you look at are fundamentally different, the way you think about them is fundamentally different. I wonder if the same might be true when it comes to the machine learning in the probabilistic world.

Rivest: Yeah, absolutely. I think that what we're seeing is the success of probabilistic methods and the success of large-scale computation machines to do this. I think that the stochastic gradient descent and algorithms like that are really carrying the day in terms of being effective at machine learning and these are kinds of algorithms that you couldn't have implemented a decade ago probably, efficiently, and the ambition with which the machine-learning folks are applying these ideas to modern practices is quite incredible. I was very much impressed with the work that Google did on the Go playing program. I played a lot of Go as a graduate student, so I can appreciate some of the subtleties of the game and the Go playing program that evolved through these machine-learning techniques very, very impressive and it's hard to explain what they're doing in some sense. I mean, again, you're not designing an algorithm because you know how to solve the problem and you're trying to capture how to play well. You're designing an algorithm that's capable of learning how to play well by playing against itself in this case.

Levin: Back to the movie "War Games," right?

Rivest: Yes. <chuckles>

Levin: You mentioned the textbook in passing and I wanted to come back to that. In its third edition. It sounds like a fourth edition might be on the way.

Rivest: Fourth edition's on the way, we hope. Planning it.

Levin: How did you get involved with Cormen and Leiserson on that?

Rivest: So at MIT, we have taught a variety of algorithms courses, so Charles and I and the faculty were teaching 6.046 at the time, which is our basic algorithms course. That course has now splintered into 6.046 and 6.006, which is a sophomore-level course, and Tom was a graduate student at the time helping to teach the course and we-- as usual for courses at the time, we had course notes that were evolving. We had-- every lecture has, you know, 10 pages of course notes that were written up by either a TA or the faculty and it sort of evolved towards a book and at some point, you say, "Well, this should be a book," and so Tom and Charles and I said, "Let's turn this into a book." Little did we realize how much effort that would be. It took a long time to get the first edition out and ready to go, but it eventually happened.

Levin: So there were other algorithms textbooks around at the time. Aside from just wanting to have your own, was there another thinking about a different slant on it, perhaps, than existing books?

Rivest: I think the level of presentation, I mean, I think the care with which you go through the proofs and presentation of the algorithms. It's a fatter book. We go to-- we explain things a little more detail than some of the other textbooks do. If you want something that's for a much more advanced student that just gives some of the intuition, there are other textbooks that might be better, but for an introductory textbook, I think we wanted to hit the nail on the head for a student who's never thought about algorithms before explaining all the details, and so I think we succeeded at that, yeah.

Levin: Was there any sense that just the level of preparation that an undergraduate coming into MIT would have with respect to computing was changing, or was that not a factor?

Rivest: It keeps changing. It's hard to tell what to expect out of students coming into an algorithms class or coming into MIT now. MIT is struggling with the question at this time as we speak, even with the questions to whether we should have a introductory computer science course for all undergraduates,

whether they're CS majors or not. But many students do come in with a background in CS and some programming or some familiarity with computers and so on too. In terms of how it impacts our textbook design, we're presupposing "not much". We're presupposing maybe an ability to program a little bit. I think the idea of looking at a piece of pseudo code and understanding what that meant in terms of an algorithm had to be covered in parallel with teaching the algorithms or preparatory to that, so-- but we wanted to stick at the level of pseudo-code rather than writing a book about, say, Java.

Levin: So you made a couple of passing remarks previously about your work in voting, and I'd like to spend some time talking about that now.

Rivest: Sure.

Levin: I went back and looked at the papers that you've written and the earliest one that has the word "voting" in the title is-- I was intrigued to notice is entitled The Business of Electronic Voting, and that made me wonder, how did you get interesting in the topic in the first place?

Rivest: So I think that the interest in voting arose out of general crypto protocols, right, so cryptography has evolved from just being about algorithms to being about applications that have interesting challenges technically to them, and so there's lots of things you can say, "Well, can you do this with crypto? Can you do that with crypto?" so payments, for example, is one of those that come up all the time. We see a resurgence of that recently with Bitcoin and things like that, but even payments had been around the literature for a long time. And there's lots of other protocols, maybe some more abstract like oblivious transfer and some of them more applied, like payments or voting. So voting has been a interesting application area for cryptographers for a long time and so I taught a crypto class where that was of interest way back when, back in the late '80s maybe, so that-- "Can we vote somehow?" and I know Josh Benaloh was interested in these things and got involved in some of that way back then, did a thesis on that. So voting has been an area that-- can you-- voting's got challenging aspects to it. Voting is-- because of the secret ballot, and that's the one thing that's really unique about voting and makes it hard is protecting the privacy of the vote, and that's important. Talk about that more if you'd like, but because of that, the difficulty of implementing a voting scheme is large. You have to think about how to approach it in a way that doesn't allow somebody to sell their vote. Selling votes has got a long history in the United States. Great book you can read called Steal This Vote, I think it is, but it's about selling and buying the votes, and if you don't design your voting system right, you can have all kinds of corruption.

Levin: So your interest in this goes back further than just this paper, which I...

Rivest: I don't actually remember that particular paper. I'm not sure what's in that particular-- so we'd have to go back and look at that one, but I have had an interest in voting for a long time and early on, it was both the design of cryptographic voting schemes -- how do you implement a scheme that would work

for voting based on cryptography? -- and also, from a fairly early stage, the auditing of elections -- how do you audit an election to ensure that the outcome is correct? -- and they sort of go hand in hand.

Levin: Well, as you just said, voting is a unique problem in some ways, partly because of the secrecy of the ballot. I think that people understand that computer security and cryptography are gonna be tangled up in electronic voting, whatever that means, but they don't really understand maybe what the core issues are. Could you talk about...

Rivest: Sure, I think-- just to reflect on what you said a little bit too, I think voting is sort of particular in our democracy in the sense that you want to have everybody understand what's going on. You don't wanna have to trust the experts or trust the technology any more than you need to, and so to the extent which you bring in complicated technology or complicated crypto, something like that, it may be inappropriate, at least for certain populations, because people don't understand what's happening. But the core issues for voting, I mean, you need to have some control over who's voting. You wanna make sure every person can vote at most once. You wanna make sure that their vote is private -- and that's the key issue with the secret ballot -- and then you wanna have some verifiability that their vote actually counts the way they intended. So the phrases I like to use are "cast as intended, collected as cast, and counted as collected," and you want all three of those steps to be verifiable, so paper ballots turn out to be a marvelous technology for voting systems. People often are surprised that I come from the Massachusetts Institute of Technology and I'm a fan of paper ballots <laughter>. It's really the best technology out there for voting. You can mark the paper or fill in the ovals and you can see how you voted. It's verifiable right there, whereas seeing what bits are recorded inside of a computer is really tough. The computer can tell you it's recorded X, but it's actually recorded Y, and so much harder to evaluate. This is why we're having problems right now in Pennsylvania, which is a-- there's a recount going on in Pennsylvania as we speak, I think it is, and parts of Pennsylvania have paper ballots and parts of Pennsylvania don't have paper ballots and those that don't, it's really hard to figure out what a recount should mean.

Levin: So there are a number of technical issues, obviously, involved and there're also a bunch of nontechnical issues, which I think you alluded to by saying that people really ought to be able to understand these systems, even if they're not technical. Can you kind of elaborate on that a little bit?

Rivest: Yeah, voting is interesting because it's a nice-- it's what some people like to call sociotechnical systems or whatever it is. There's all kinds of considerations that come into play. The technical ones are having to do with, how do you represent the ballot; how do you tally the ballots; how do you confirm that the ballot tally is correct and so on, but there's lots of other issues that come into play too: cost, usability, understandability. Things like this are certainly-- accessibility-- are part of the picture too, and so voting turns out to be one of the more difficult technical systems to build and the secret ballot is the one that really makes it the toughest because you can't use a lot of the same auditing techniques that you use with other systems, because you can't keep a tight coupling between who put in this vote and what the vote is. It's gotta be somehow disconnected a bit.

Levin: Are there any analogues in other societal areas that we can draw on?

Rivest: Not very much. People often say, “Well, I can bank online. Why can’t I vote online?” for example, or things like this, which is-- doesn’t follow because of the difference between banking and voting. Banking doesn’t have a secrecy requirement. Or people can say, “If I can put a man on the moon, why can’t I vote online?” or something like this, but again, putting a man on the moon-- there’s not people shooting at the rocket while it’s going up. The voting system is, as we’ve seen recently, maybe subject to attacks, so it’s gotta be a highly secure system, an understandable system. It’s gotta have all kinds of verification capabilities designed into it, so a paper ballot system that's-- you fill out the paper ballot; you scan it with electronic scanner; you have the paper ballots; you can recount them if you need to and you can randomly sample those to audit them if you need to, is probably in today’s world the best choice.

Levin: But there are some other areas of society where we try to legislate and enforce privacy and in some cases with technology involved. I’m thinking of medical information. Is there anything we can learn from that?

Rivest: It’s hard. I mean, I think that medical-- they’re rather different. Medical systems have different notions of privacy than do the voting system. Voting system is just, how did Person X vote, which is all you care about. Medical system, you may care about a variety of different systems. You may want it private to the outside world, but not private to the doctors and so on they are various kinds of disclosure, you might wanna have disclosure rules. There’s nothing quite like voting in terms that we can leverage off of. We have the tools that we can-- confidentiality per se is a grand goal of security and of cryptography and we can use some of those tools to achieve a level of voter privacy in the voting system, and there is a realm of what’s called end-to-end voting systems, which I’ve worked on and other people have as well, where you have a voting system where you protect privacy cryptographically, the voting privacy cryptographically, so when you’re casting your vote, you’re casting your crypto-- cipher text, and so you have a process of casting a vote which involves creating the cipher text and knowing that the cipher text actually represents your vote, and then that cipher text, being cipher text, could be posted on the website next to your name so you can see, you know, so-and-so voted this way, but as the cipher-- and ciphered this way, and so there’s a collected as cast check you can do that you can say, “Did my cipher text make it to the pile of votes to be counted?” so you can see “My name is there and my vote is there,” and then there’s a tally that's produced and you need to know that the tally is the proper tally for that collection of cipher text without decrypting everybody’s cipher text. This can be done too with a little crypto magic, say using homomorphic encryption of some sort, and so there’s some technology there for doing-- so some nice tools for applying cryptography to voting, as well as taking this off in a different direction, and some of these schemes are in fact being deployed and used as we speak. Tacoma Park, Maryland, has had some elections using some of these andad schemes past few years and Travis County, Texas, is implementing a scheme called Starvote, which implements some of these kinds of ideas as well, both paper ballots and the cryptography, so we’re learning as we go along. We’re learning how to take this cryptographic ideas, these ideas of verifiability, and deploy them in real world systems

that are both secure and usable in a nice way, so I think we're making progress, but slowly in the voting arena, but people keep pushing for things which are, I think at this stage, not really possible security, like Internet voting, and so one of the things I do is, I spend a bit of time talking with people about the risks of Internet voting, and I think we're just not there yet to be able to do that securely.

Levin: So it'd probably be helpful to contrast electronic voting from Internet voting specifically.

Rivest: Yeah, electronic voting is a sort of a vague term that sometimes means Internet voting as well, but electronic voting often refers to just a machine where you're recording your vote electronically. It's called a DRE in the voting world, for Direct Recording by Electronics. You touch a touch screen, your vote gets recorded somewhere in the innards of the machine, and you have no idea if it was recorded properly or not. There's no verifiability of that vote. So electronic voting has its own problems, and so it doesn't have a tangible of record of how you voted, but Internet voting is even worse in the sense that you've got people voting in all kinds of machines, lap-- their own laptops, which may have malware on them. You got the Internet to deal with. You got denial of service attacks to deal with, lots of other things.

Levin: So it's a-- again, as you said, when people say, "Well, we can send a man on the moon" or "I can do electronic banking. Why can't I-- why can't I do voting online?" it's these verifiability issues _____.

Rivest: Verifiability is some of it, yeah, yeah, and having-- some of these things are scalability and having robustness under attack too. I think it's really-- if you've got a server that's collecting votes on Election Day and the server goes down because of a DDOS attack, you're sort of hosed for that election and so there's no reason to do that. One of the questions you should always ask about a proposal like that is, why do you wanna do it, and people often propose answers like, "well, it would increase turnout" or "people will like it" or something like that. And the first of which seems to be false: the evidence we have is that putting something-- a voting system online does not increase turnout. In fact, it may increase-- may decrease turnout, that people are unfamiliar with the technology -- they have an awkward time. There's lots of people who don't use the Internet much yet and things like this, so putting a voting system online may not increase turnout. And once people start talking about the security issues, they're wary of it and they often prefer to go vote on paper, so you'll have political parties saying, "Don't vote online if you have the option. Go to the polling site. Go early, vote on paper," so there's reasons, both security-wise and ease of use and so on too that I think people find for voting by paper rather than voting online, so it'll be along -- maybe someday. Maybe we'll have secure enough phones down the road that we can do that, but you don't wanna be in a situation where you're trusting that particular vendor or manufacturer to count your votes.

Levin: Nevertheless, there are vendors and manufacturers out there purveying equipment that people...

Rivest: Yes.

Levin: ...in authority buy, right?

Rivest: If you're in the voting business and wanna make a buck, you may take on some responsibilities you shouldn't be taking on, some responsibility that you're not actually ready for.

Levin: Are there any good examples that we can point to, perhaps outside the United States?

Rivest: There are examples of people attempting things like Internet voting outside the United States, in Estonia for example. The security evaluations I've seen of those by people like Alex Holderman and so on do say these systems are really not ready for prime time yet either.

Levin: Are there systems that have been demonstrated to work in any reasonable way, perhaps on smaller scale than something like a United States national election?

Rivest: There are systems that use some of the principles I talked about. In Israel, for example, there's the so-called Wombat system, developed by Alon Rosen and others, which involves paper ballots, but they have this end-to-end cryptographic flavor to them, so you get a paper receipt that's got two parts, one of which has the candidate's name you're voting for and you put that in the ballot box and the other part has a QR code that you can take home and validate on the web, so on too, so there's things like this that I think are very promising and have been used a bit. But these are not online voting systems. These are poll-site voting systems still.

Levin: Mm-hm. I want to get more into the question of future directions for voting, but before we do that, I want to come back and ask for your perspective more broadly in the area of computer security. I found an interview that you gave almost a decade ago, in 2008, with Dr. Dobbs. And had an interesting exchange in it talking about credit cards. They asked you the following: "A Nielsen survey released March 12th of 1997 showed that people fear buying things on the Internet because they don't think their credit card numbers are safe." I'm a little puzzled as to why in 2008 they asked you about what a study 10 years before said about buying habits on the Internet, but we'll ignore that. It's your answer that I found interesting. You said, "They're probably safer using their credit card for Internet transactions than they are using it at a local restaurant. It's a question as to what the risk is. The risk in a restaurant is if the waiter or waitress you give your credit card number to copies it down, keeps a copy of the carbon, whatever, sells it to a friend. My personal guess is that this is more likely to happen in a restaurant than on the Internet. Not that this situation couldn't turn around at some point, but currently, the number of credit card thefts carried out over the Internet is probably very small compared to the number carried out in ordinary institutions like a restaurant or shopping center." So now it's nearly a decade later. The world has changed quite a bit. I'm not sure that the situation has improved, at least not in U.S. restaurants. What are your thoughts on credit card security and the time with computers today?

Rivest: We've gotten better. You've seen all the chip cards come out now, so we have, instead of the magstripe cards, we have the chip cards and they promise a step up in terms of security. It's, provides, a proof of possession that you don't have with the magstripe card quite so much, because the magstripe card you can copy and easily you can take the old iron out and take a mag tape and put it over your card, just copy the bits easily. So I think we're getting there. We haven't moved to the chip-and-PIN level of technology that we have to type in the PIN as well. But, you know, by and large, the security-- the fraud departments and credit card companies, have gotten very good. And I think a lot of the security we have now comes less from the technology of the card as it does from the ability of the fraud departments to say, "You've never shopped at this store before. And moreover, you shopped at another store hundred miles away two seconds ago," or something. You know, or whatever. And they understand weird buying patterns. Much more readily than they used to. They pay attention to those things. Those matter. So it may be that the-- it's not the technology of the card as much as it is the ability of the fraud department to very quickly see that this is a irregular purchase and should be denied. And I suspect that has a lot to do with it. Plus, they make a lot of money these days and they're willing to take some losses, as always. And that makes the system work.

Levin: So that raises a couple of other questions. I think maybe this is an application in machine learning that is helping them to do this. I don't know. But it wouldn't surprise me.

Rivest: Probably it is. Yeah.

Levin: But there's an economic calculation, obviously, for the credit card company, which has to factor in some, I would say, somewhat intangible issues related to how much of the risk is borne by the merchant versus the credit card company, which varies in different parts of the world. Different laws apply. But the other thing is, the inconvenience for the customer when a card is compromised. I mean, it's all very well and good to say, "Well, we'll send you one express mail overnight." But if you're traveling inwestern China or something, that might not be quite as convenient as you had in mind. You might prefer then to--

Rivest: Because I was in Ireland and had a card stolen at the time.

Levin: I'm sorry, say again?

Rivest: I was in Ireland-- or no, it was Iceland-- once and I had a card stolen. And very inconvenient. It was--

Levin: Exactly. Exactly.

Rivest: --a big deal.

Levin: So you probably weren't so happy with whatever decision they made that allowed the fraud to occur.

Rivest: This was-- yeah.

Levin: So there are tradeoffs there, but it's not immediately obvious, to me, at least, that they're factoring in considerations that are of paramount importance to the customer.

Rivest: Well, I think they don't want to lose customers. So they understand the actuarial-- This is why they have, you know, so many tiers of risk. You know, 22 tiers or something like that, of different kinds of discount rates for the merchants and so on too, with different levels of risk. And customer service relationships, customer service departments that are well-staffed and ready to help out quickly if they can. So they don't want to lose you as a customer. It's an actuarial game. I mean, it's the kind of thing where you've got remedies for situations you can handle. And again, back to voting. This is one of the things that makes a difference in voting in a sense that, you know, once the vote is cast and it's broken its tie with you, you know, there's no remedy if something goes wrong. You don't know what's happened to your vote quite necessarily. And, you know, there's no way to fix up a broken election. Whereas with a broken credit card transaction, it's… And maybe they can do something.

Levin: So I think that actually leads pretty nicely to the next thing you said in the Dr. Dobb's interview, which was, "The security's a cat-and-mouse game. You can't afford perfect security. Companies make investments to ameliorate the risks. Over time the technologies improve. It gets harder for the bad guys. Breaking into bank vault isn't done very much today. It's gotten really hard to do. You went through a series of stages, and now you've got vaults that are really elaborate. We'll see the same evolution with Internet security." Have we?

Rivest: I think we have. I think we have. I mean, I think now we've gotten to the point, for example, with iPhones, where they're really quite secure and in fact so much so that law enforcement is complaining about the security. And that they can't get into them when they feel they need to.

Levin: Right. Which raises another interesting topic that I hope we'll talk about a little bit more.

Rivest: Yes.

Levin: So it seems that on electronic voting, coming back to sort of where that's going, if we look at the history of introduction of computer-based or computing-based technologies into mainstream society, we often see that they start out as fairly inferior solutions that work well for some portion of whatever problem they're trying to solve, but don't address the other. And then over time they get better and adoption

increases, and eventually the old solution maybe doesn't go away but gets down to some relatively low level of use as the good stuff is made better and the bad stuff is more or less dealt with. Do you think electronic voting will follow a path like that?

Rivest: It's harder, I think, because you don't want to be in a sort of a trust-me situation with the voting vendor and the voting technology. So the question's, "Who are you having to trust in order to trust the election outcome?" You know, so if somebody says, "So-and-So won the election," well, maybe it's believable, maybe it's not. But if you believe it, who do you have to believe in order to do that? Do you have to believe the manufacturer of your smartphone that they've actually implemented something secure and haven't tampered with a-- like, if your smartphone was manufactured in China or the Chinese, you know, did they put something into the phone that might've interfered with your vote? It's not beyond the pale. We've seen Android phones recently which ship a lot of the call data off to China, you know, unbeknownst to the owner. So that having a manufacturer of a phone put in some malware that could affect the vote is entirely plausible. So we've got those issues to deal with. But things are getting better. But technology's very complicated. I mean, having a voting system where the number of lines of relevant code is larger than the number of voters seems, you know, a dubious kind of thing in some ways, right?

Levin: <laughs>

Rivest: You know, you've got-- typical commercial code is three bugs per thousand lines or something like that, too. And these, each of these, maybe a security vulnerability. It allows you to change the vote. So you've got to be very careful. That's one of the reasons we introduced the notion of software independence. John Wack and I wrote a paper number of years ago which defined software independence as: a system, a voting system that's software independent if, you know, an undetected change in the software can't cause non-detectible change in the outcome. And basically says you're not dependent upon the software being correct in order to confirm the outcome. So things with paper ballots have that property because you can, even if the software and the scanners were manipulated with, you've always got the ability to go back and look at the paper ballots and see what the correct outcome is. So I think that software independence is a key notion you'll want to have for a voting system. And that's hard to do with an all-electronic system, because the evidence that you've got is in bits. It's tamperable, it's manipulable, without the voter being able to verify them directly. So it's really pretty tough.

Levin: Right. Well, that's where this end-to-end argument that you were talking about before--

Rivest: Yeah, yeah. The end-to-end argument--

Levin: --has a place.

Rivest: --can get around some of these things, yes.

Levin: On the need for improvement in voting, which may not necessarily mean electronic technology. In 2012 you and a number of other distinguished scientists sent a letter to President Obama on the need for voting improvement. Did you get a response to that letter?

Rivest: We didn't get a response to that letter, no. No, but--

Levin: How disappointing.

Rivest: I think the current election, the 2016 presidential election, maybe will have more long-term impact on voting as we move forward. People will understand that to the extent that we have systems that are online or electronic, they may be vulnerable to foreign actors trying to manipulate the vote. The Russians may have interfered with not only e-mail but possibly voting systems. You know, there's no evidence that I know of yet that says they did, although they did interfere with the people's e-mail systems. But we see that the vulnerabilities that we see in all the other realms of commerce, you know, Target and other companies being, or OPM being hacked, could easily happen to a voting system vendor or a voting jurisdiction. And so the sensitivity, the vulnerability, of our systems to hackers of various sorts has become much clearer in this current election round. The need for, in my mind, paper ballots and good auditing techniques has become even clearer.

Levin: Mm-hm. We always have irregularities in elections, right?

Rivest: Yes.

Levin: Just thinking back over the U.S. elections that I recall, there's always something that comes up, and it usually seems to me that the noise dies down relatively quickly after the election about whatever the irregularity was, because either it wasn't really an irregularity or it didn't actually make much difference. And so this whole question of making much difference is in some sense related to what I know you've been talking about recently, and that's the auditing of elections. Could you say a few words about that?

Rivest: Yeah. So I think that one of the things the elections should do is not only produce the correct outcome but produce evidence that that is the correct outcome. And so you want to check, you know, does the evidence trail support the outcome that was announced? Is the reported outcome the correct outcome? And so that's a great question. And you can, turns out, you can do such things efficiently with risk-limiting audits and things like this have been evolved by Philip Stark and others. So the idea is that you want to be able to take a statistical sample of the paper ballots and do some statistics on them, on a sample, to see whether the outcome is consistent with that sample. Maybe the result is that you need to enlarge the sample because the election's close, but typically they'll say, "Yes, it's right," or, "It's wrong," or maybe take more data. And so that's a area of interesting research as we speak and one where I'd

like to see more legislation supporting of audits afterwards. I distinguish an audit here from a recount whereas an audit typically means a statistical sample of some variable size, perhaps, whereas a recount is sort of, by definition, recounting all of the ballots and it's quite expensive. But an audit can be very simple. The last presidential election that we just had, Philip Stark and I made some estimates of how much work it would take to audit, say, the entire country, state by state. And a state that's got a reasonable margin of victory for one candidate or the other is very cheap to audit. For example, California looking at 70 randomly chosen ballots out of the whole state as enough to sort of, should be enough, to confirm the outcome.

Levin: Seventy?

Rivest: Seventy.

Levin: Seven zero.

Rivest: Seven zero, yeah. Yeah.

Levin: And how many votes are there in California, roughly?

Rivest: Millions, right. Yeah.

Levin: Yeah.

Rivest: So it's a _____--

Levin: That's amazing.

Rivest: It's amazing. Right. It's a very simple. You just sort of say-- you know, if somebody tampered with the ballots enough to cause the outcome to be wrong by that much, they would've had to tamper enough that 70 ballots would've discovered the discrepancy somehow, so…

Levin: That's amazing.

Rivest: Yeah. So statistical thinking's going to be very powerful. As we see not only in AI but also in voting here.

Levin: Yes.

Rivest: And so getting an efficient audit, I think, on the books as part of the law -- required -- is part of the basic practice, the good hygiene that you want to do with every election, would be something I'd like to see happen.

Levin: So what do you think are the prospects for having auditing become a-- sort of a standard part of the election process?

Rivest: They're not too bad. It's certainly the right approach to take. So there's just only a technological compelling argument to make that, "This is what you should be doing." And I think part of the difficulty's just effecting that change, convincing people that it is-- that this does work, it can be done fast. And, you know, getting it on the books to say that it's legally required. Change is hard in voting systems. People who are elected don't like to change the system they were elected by. You know, there are constraints, calendar constraints and so on too. People want to know the result right away, so putting in something that happens after Election Day that still tries to confirm the output-- confirm the outcome can be controversial or difficult. But I think it could be-- we're starting to see that, things starting to happen, in various states now. A number of states have various kinds of auditing requirements. They're not always as mathematically tough as you'd like to have. But they're examining a number of randomly chosen ballots. I think we'll get there. So I tend to be optimistic that this is the right direction to go in and that over, you know, another decade maybe we'll see it all fall into place.

Levin: Mm-hm. That's interesting. I wonder if there were to be principled opposition to auditing, what it would be.

Rivest: I think the arguments that are made are typically things like, you know, "It doesn't fit the calendar," or something like that. It's, you know, there isn't enough time between the time the polls close and between the time you want to certify the elections to do the audit. And I think those things could be worked with. I think that's sort of the only principle that I can see that's valid. The fact that if you don't have paper ballots, I mean, you've got other issues to deal with. I mean, you've got to have a foundation of something to audit that has sort of a ground truth in it that the voters have looked at. So the security fundamentals that, you know, you start with the evidence trail that the voter has looked at and, you know, the voters verify. And that gives you the ground truth. And then you just want to compare that ground truth against the machine interpretation of that ground truth, what the scanner said. And that can be done efficiently. So I think there isn't any basis for a principled opposition to this. Unless you have already, for example, doing hand-count everywhere. So I see that some, a few jurisdictions, count all of the ballots by hand for the first count, and that's a pretty good process and that may not-- that may be a principled reason not to bother doing a sampled recount again. Perhaps.

Levin: Yeah. It's interesting. We're perfectly willing to have elections drag on for weeks when they're so close that we need to have a very careful recount.

Rivest: Yeah.

Levin: But when you might have an election be a few days or a week longer just so you could be confident that you didn't need a recount, that seems a little odd that you could object.

Rivest: Yeah. Right. There's lots of things that need to be improved in our voting system. Hopefully we'll get there. I mean, be nice to have a national holiday for Election Day. It'd be nice to have nonpartisan districting rules set up and so on. There's just lots of ways to improve voting still, so make your democracy work as best as it can, is what…

Levin: That's obviously a long and complex topic and one that will be under discussion probably forever.

Rivest: Yeah.

Levin: I want to move on to another thing, but since the voting stuff is such a recent area of interest and one that has obviously got a lot going on, I have to ask you, just for in terms of your own personal research, where do you see that going in the voting area or maybe in other areas over the next few years?

Rivest: Well, the voting arena, the two things I'm working on at the moment, include, well, some pure policy things. So the idea of Internet voting keeps bowling up and I keep trying to push back against that and explain to people that, "This is-- we're not ready for that." So that's not research, per se. That's sort of just advocacy. In terms of the research, there are interesting ideas that I'm exploring with Phil Stark and others on auditing elections. And there's lots of interesting deep technical questions there, particularly as you get into non-plurality elections. So if we have instant runoff voting, for example, where you list your preferences and then there's a complicated algorithm that says, you know, how those ballots are combined to come up with a winner, that may be the kind of election system you want. But then you can ask the question, "Well, how do you audit that, such an election?" Right. And if I give you a sample, how do you tell what to do with a sample. And so there-- we got ideas on how to handle that, but it's real research there that needs to be dealt with.

Levin: Mm-hm.

Rivest: Also in the cryptographic realm, the end-to-end cryptographic systems, I think there's a lot more to be done there to try to simplify and make those work as well as we can. And I'm optimistic about the STAR-Vote project in Travis County, which sort of combines some of the auditing ideas, some of the cryptographic ideas and so on too. So we've got some practical experience to gain with those and then maybe some redesign to make those kinds of systems work better. But-- so applying cryptography, applying statistics, to make the integrity of elections be really what it should be is sort of the high-level goal.

Levin: Well, and there's lots of work to do, as you said.

Rivest: Lots-- yeah. <laughs>

Levin: I want to come back to what really was the motivation for this interview in some sense, from the beginning, which was your-- the fact, that you won the Turing Award with your colleagues, Len and Adi. That was in 2002, if I remember correctly, right?

Rivest: Mm-hm.

Levin: And a certain amount of time has gone by since then. But that's time to sort of reflect on the impact that that might've had on your work and your career and the way you interact with the rest of the field and the world. What are your thoughts on that?

Rivest: So the Turing, you're talking about the Turing Award, per se, as-- yeah.

Levin: Yes.

Rivest: As opposed to the RSA algorithm itself. Yeah. The Turing Award is a nice recognition of work done. It certainly highlights things that are impactful and have importance to the field. Cryptography has certainly, as a field, blossomed and had a lot of… both theoretically and practically, in computer science, since the early days. I mean, it's really turned out to be a nice bridge between a lot of the practical aspects of the field and some of the theoretical aspects. It's had connections with complexity theory and so on too. So I think in some sense the award is in part a recognition of just the importance of cryptography to the field and being an early contributor to that. We've seen the Diffie-Hellman paper also get a Turing Award more recently, which is nice too. So I think there's a lot of conceptual and theoretical things happening in the crypto field that are just very important and I think it's good to see, get recognition. I think it helps emphasize that security and cryptography are issues and technologies that people need to understand and we talk about educating next generation of computer scientists and so on. It's nice to have examples of technologies that are not too difficult to understand. I think RSA is an

example of that, actually. That give real impact in terms of security and so on too. So I think just the recognition, the highlighting of both the field and the particular technologies, is nice to see. I'm not sure that's responsive to your question or not. I--

Levin: Well, that's certainly relevant. I was wondering if it may have had any more personal impact on you in terms of what you've done in the last decade and a half or so?

Rivest: I think my work has sort of proceeded along the same kind of trajectory it would have anyway. I mean, I think the kinds of questions that my research has followed hasn't changed because of the Turing Award. I think it's a nice recognition to have made, but it doesn't raise new technical directions for me. So I've probably given more talks on the older work because of the Turing Award. But the new directions I think remain with voting and then some of the cryptographic applications to voting and things like that are still the same.

Levin: Do you find it opens any doors or has opened any doors that might not have been opened otherwise?

Rivest: I don't know about doors. But it's recognition that's nice and it's… There is the new Heidelberg Laureate Forum that ACM sponsors.

Levin: Yes.

Rivest: And it's very, very nice. And so I attended those once, and that was nice to meet other award win-ees there, either ACM Turing Award winners or Nobel laureates and so on too – it was just a fantastic event, so it's a wonderful way to get together with people who have also won similar awards. And also the young students who come to that. It's a very nice event for students and post-docs who come to that.

Levin: Yes, absolutely. I think it's a great thing.

Rivest: Yeah, yeah, yeah, yeah.

Levin: I think sometimes the benefit of these awards comes in the fact that not so much people in the field who already know about and appreciate your work, but people outside the field-- it's kind of a funny, for want of a better term, certification.

Rivest: Yeah, yeah, yeah.

Levin: That says, “These are people who know what they’re doing. You should pay attention to what they say.”

Rivest: Yeah, yeah.

Levin: Maybe you can leverage that in the voting area. I hope so.

Rivest: I think some of that helps, yes.

Levin: <laughs>

Rivest: It does. Definitely. Yeah.

Levin: I thought maybe we could wrap up with a couple of maybe speculations about some aspects of computer security that are with us today and likely to lead us in interesting directions. You’ve already alluded to one of them, which I’ll hold off for a moment. I’ll talk about the other one. The notion of digital certificates, which grew out of the cryptography work, and is very important: Certificates are everywhere. You know, every time you use a web browser there are a zillion certificates running around. But I suspect it’s the case that most people who <laughs> spend lots of hours every day in a web browser have no clue about certificates, what they mean, what they’re being used for or any of that. First of all, would you agree, and is that a problem?

Rivest: I do agree. I’m not sure it’s a problem. I think that, you know, the questions are what are the aspects of security and cryptography that need to be in your face when you’re using these systems, which were the ones that are better under the hood? And so when,\-- you know, certificates sort of play a role of the glue of putting these things together. So you want to have a-- and I think we’re still a long ways from having a PKI that really works for everybody. I mean, it’s a-- the binding of names to keys, the questions to who has authority to do that binding. What happens when keys go bad and how you revoke certificates and so on. There’s a lot of detail that has to do with digital certificates, that we’re still not handling as well as we should. And, you know, I’ve spent some time over the last decade thinking about some of these issues and trying to come up with better systems. And it’s hard. I don’t have any proposal to put on the table yet, but there’s certainly things I’ve got ideas about and I think it’s challenging because it deals with many aspects of security that are sort of fuzzy almost: I mean, who has the authority to do a certain, to make a certain assertion? I mean, do I have the authority to assert that your key is such-and-such. If so, what strength should somebody give to that assertion? And, you know, maybe if it’s a corporate relation, we’ve got other certain kinds of rules that affect us. If it’s a personal situation, something else. So there’s lots of aspects of this that make it complicated. And then when things start going bad, you know, how you trace out the consequences of an attack and all of a sudden you think that it’s a certificate that’s gone bad, somebody’s discovered the private key, and then they signed off on that

on other keys, how do you find all the consequences of that? So it's a complicated situation. So that complexity: much of it should be hidden from, I think, the typical user. You want something that's sort of simple, you know, little lock icon that goes on. User interfaces for security is a area which I haven't studied deeply, but there's a lot of surprising things that happen there where people just don't pay attention to things. They click on banners that come up just to make them go away without reading them and so on and so forth. So I think it's designing a security system that minimizes the amount of complexity, the amount of understanding that a person needs to have and does succeed at getting their attention when it's important to get their attention. It is still-- I, you know, things, surprising things still happen. I was at a website just a couple days ago making a donation or I thought I wanted to make a donation to this organization and then I realized their site was totally insecure. They had no https on their site or anything like that, so…

Levin: Huh.

Rivest: You know, see, I had questions, what do you pay attention to? And when does it catch your eye?

Levin: Most people would not notice, I imagine.

Rivest: Yeah, probably most people would not notice. But I said, "I'm not making a donation this way." You know, so I wrote a check.

<laughter>

Levin: Paper again.

Rivest: Paper check. Yeah, yeah. Back to paper, so…

Levin: It does seem, you know, you said maybe this isn't a problem, but the whole complex of things. Maybe the underlying certificate technology is nice and solid, but the way in which it's deployed and the way in which people interact with it, it sounds to me, is less than ideal.

Rivest: No. I think the underlying technology is not philosophically well-founded yet. I mean, the whole name-- the question, "What's the name space for a certificate?" --

Levin: Yes.

Rivest: -- is not well thought through. I mean, a hierarchical name space doesn't make sense for the world, so I think I've got to work on that. I did some work way back when on this thing called SPKI/SDSI (spooky/sudsy), which sort of tried to do a different take on some of this, and I think there's some good ideas there that should play a role in a future system. But, you know, the current framework is-- works okay for e-commerce and it sort of falls flat anything when you try to go push beyond that at all. You know, put in certificates for you and for me in a personal sense, you know, we're just not there.

Levin: Mm-hm. So e-commerce here has the advantage that you can put a quantitative value on--

Rivest: Put a value on. <Inaudible>--

Levin: --on the loss that you might sustain as a result of something like that.

Rivest: Yeah. Well-defined companies and-- yeah.

Levin: Yeah.

Rivest: So business units you could--

Levin: Yeah.

Rivest: You can certify. Yeah.

Levin: So the other topic that I wanted to come back to is one that you've already mentioned, which is this question of key escrow, by supposedly--

Rivest: Yes.

Levin: --trusted authorities. And, you know, the tussle, for example, between the government and Apple over the iPhone--

Rivest: Yes.

Levin: --iPhone security. I think you recently published, or maybe it was last year, an article in the communications about the issues involved in key escrow and about balancing legal and individual rights.

Rivest: Yeah. So the--

Levin: So say some things about that.

Rivest: So this is an issue that's persisted throughout the development of cryptography. Law enforcement in particular has determined that-- perceived that cryptography makes its job difficult or impossible in certain situations, that they have a perceived need or authorization to access certain information and the cryptography just doesn't make it possible for them to do that. And so they get frustrated and they push for laws or help somehow to do this. Back in the '90s they had a proposed solution. They said everybody should have the Clipper chip and the Clipper chip should have a escrow key in it and every time you encrypted anything a copy would be shipped off to law enforcement as well. And that was blown apart by Matt Blaze and others who discovered serious technical flaws with the particular proposal. More recently, they sort of say, "Well, we're not going to propose technical solutions. You guys are smart. You should figure out what should be done here." But they don't even define the problem. I mean, if you ask them for solutions, the problem is not well-defined. Who should have access when? What kinds of-- what's plaintext? What's ciphertext? You know, for example there's a classic technique for erasing data. You encrypt the data. The data could be backed up and encrypted for them nicely. You know, this is a nice way of handling the erasure data, because when you want to erase that data you just erase the encryption key. Okay. So that's a nice way of making the accessibility of the data go away. But now you're in a situation where you've got ciphertext on your machine, which itself hasn't been erased. Only they key has been erased. And you don't have a way of decrypting it. So now you're probably in violation of whatever laws that the FBI wants to have or something like that, because there's no way that they can get access to it either. So is it going to be illegal to erase data? I don't know. You know, what's the right way to think? So there's lots of different angles to this. You know, sort of like saying, you know, ball bearings are problematic for law enforcement, because they allow bad guys to go fast. You want to have some regulation of ball bearings. Well, that's a kind of complicated thing to ask for, you know. And crypto is like, is in the middle of everywhere, just like ball bearings are in the middle of everywhere. And trying to figure out all the different places that crypto might apply and how you might regulate them -- there's a lot of detailed thought that would have to go into that. So I think that regulation crypto is very, very complex, much more complex than law enforcement probably realizes. And it's probably unworkable in the end and having other approaches to help law enforcement get the information it wants is perhaps the better approach. But it's a difficult situation. There's certainly societal tradeoffs to be made here. And other societies that made different judgments. Britain, for example, recently has passed their Snoopers Charter, which basically says that, you know, government has to have access to all encrypted communications and so on too. And I doubt that's going to work for them, but we'll see.

Levin: Seems like-- you said, you know, the problem's not necessarily well-defined. Might that be because the problem stems from a pile of assumptions about how the world ought to work that may not be universally shared. Law enforcement has certain assumptions about what data ought to be accessible to them that might not be shared by the general population. And that means that, you know, a solution, when you come from different assumptions, is not a solution at all.

Rivest: No. I think even defining what a ciphertext is and what the plaintext is, is hard. I mean, you've got--

Levin: I see.

Rivest: --encrypting keys, encrypting other keys. You've got keys encrypting maybe random data. You've got situations where a secret is split among a number of parties, any several of which could decrypt it. You know, there's lots of situations where the cryptographic complexity and the layering of levels of abstraction really comes into play. It makes a situation very complicated that, such that you're trying to figure out, "Well, which applications of cryptography are those that should be accessible to law enforcement?" I mean, it's not clear. You got places where you're using things called encryption algorithms but they're not used for confidentiality. They're used for authentication and things like this. So maybe that shouldn't be covered. I-- you know, you've got to be very careful about defining all these. Maybe it's approachable, but I think it's, you know, require a massive amount of work on the part of every sort of tech company that's doing anything with security. So it's… Makes me nervous that it's going to throw a lot of sand in the gears of technology.

Levin: Indeed. Indeed. Let me ask you one last question which is really more question of perspective for the future computer scientists, if you will.

Rivest: Sure.

Levin: Now, where do you think that the opportunities are most likely to lie for people who are thinking about going into computing today?

Rivest: It's a great question. I mean, the field is surprising. And like most research, I guess, there's surprising consequences to research that you may not think is what, you know-- maybe something that seems like it's a small problem may turn out to have a lot of interesting consequences down the road. Adi Shamir and I did this work on rewriting write-once memories once. It sort of seemed like a cute little problem and showed that, you know, if you've got a memory unit where you can only change zeroes to ones and you can't change them back, you can in fact store a lot more data on such a memory unit than you might imagine. And so we wrote a paper and that was a nice theoretical result. It turns out much later we learned that this has had quite an impact on flash memories and how they're used. You know, and so there's a lot of surprising consequences over it. So I think that, you know, just sort of following, being persistent about taking a problem and following it through, trying to see where it goes and, you know, sometimes these things turn out to be-- have more impact. I think that, to my taste, looking for the intersection between applications and theory is a very fruitful area. The applications that come along, that keep coming along, are always posing new problems in security or theoretical computer science somehow. You want to say, "What's the-- how do you take this problem that's a problem in practice?"

You know, maybe something new. Something to do with social media or something that's relatively new. And you say, "Well, how can I formalize that? How can I ask a theoretical question about that and then try to develop a theory around that?" So that's the kind of work that appeals to me and I think that can be recommended to new computer scientists. Sort of saying, "You know, let's try to look for the new applications," but then try to step back, abstract them, and say, "What's really being done here? What are really the hard questions? What's the hard part of this? What's the easy part of this? Can we come up with good algorithms or good approaches to the security of this?" So I think that deriving your motivation from things that people care about in the real world, but then stepping back and saying, "Can we do a theoretical approach, an analysis of the situation?" is a nice style. And that's what we see. A theory side of computer science too, a lot of. And it works pretty well.

Levin: Great. Thank you very much, Ron.

Rivest: Thanks.

END OF THE INTERVIEW

