An Interview with

THOMAS A. BERSON

OH 506

Conducted by Rebecca Slayton

on

18 April 2014

Palo Alto, California
Stanford, California

Charles Babbage Institute
Center for the History of Information Technology
University of Minnesota, Minneapolis

Thomas A. Berson Interview

18 April 2014

Oral History 506

Abstract

This interview with computer security pioneer Tom Berson discusses his early interest in computers, formal training in physics and computer science, and career in computer and network security industry. Berson earned a bachelor's degree in physics before going to IBM Yorktown Heights in the late 1960s. He worked as a consultant while earning a Ph.D. in computer science from University College London, which he completed in 1977. After completing the Ph.D. he went to work for Ford Aerospace and Communications Corporation in California, where he worked on the Kernelized Secure Operating System (KSOS). In 1979 he and five others from Ford started a computer networking company, Sytek, where Berson was involved in several innovations related to network security. In 1986 he founded a new start-up, Anagram. Berson also discusses his involvement in IEEE Symposium on Security and Privacy, the International Association for Cryptologic Research (IACR), the influence of the Orange Book, and the future of the field of computer security.

This interview is part of a project conducted by Rebecca Slayton and funded by an ACM History Committee fellowship on "Measuring Security: ACM and the History of Computer Security Metrics."

Slayton: Thank you very much for taking the time to talk to me today. We'd like to start with some basic biographical information, since it's not publicly available. If you could just say a little bit about where and when you were born, where you grew up.

Berson: Sure. I was born in New York City; I grew up in the Bronx. I was a nerdy kid. I went to public school. Public schools were laned at the time, and I wound up in a Dash 1 class, which was the top class. So in sixth grade, amongst my classmates was somebody you know, Marty Hellman.

Slayton: Oh really?

Berson: Right. So we go back to fourth, fifth, and sixth grades, we sat in the same classroom; and I think seventh and eighth, as well.

Slayton: That's amazing.

Berson: Isn't it? And other people wound up there, as well. So I grew up in New York.

Slayton: So where did you go to school after you got out of high school?

Berson: I went through high school, junior high school and high school; and everybody; I had been in one of these accelerated programs so I was 16 when I was graduated from high school; and to everybody's surprise, I got into M.I.T.; including my own. But I didn't last there because of 5.01.

Slayton: What is that?

Berson: I think I remember it right, it was chemistry.

Slayton: That's right; I was in course 5.

Berson: So there was no way I was going to pass 5.01, and it was a required course. In those days, talking about 1962, courses were marked on the curve. I had a great time at M.I.T. but it was the first time that I was with other people who were really smart. I kind of drifted through high school just by being me. So M.I.T. asked me to go away, and I did, and I went to SUNY in Oswego, where I did a bachelor's in physics.

Slayton: What year did you go to SUNY?

Berson: I think I got there in 1963 or 1964. I worked for a year; I worked for a year and took night courses in New York.

Slayton: Where did you work?

Berson: I worked at the Albert Einstein College of Medicine building Skinner boxes; Skinner boxes for operant conditioning of animals. You know, you teach them to bar press and they get a reward. In this case, the experiment was exploring how hypothermia affects learning, so these Skinner boxes that I built were air tight so you could measure oxygen uptake and so on; and you could put them in refrigerators. And the reward that the poor rats would get when they did the right thing was a shot with a heat lamp. It's a terrible, terrible thing, now that I think about it, but that's what I did.

Anyhow, I tried to go back to M.I.T.; they said take two courses; get two Bs. I got a B and a C; C in chemistry. And I thought it was good enough and they thought no. So I called up Oswego — do I really want this on the record? — I called up Oswego and I said you may not remember me but two years ago you had admitted me — because you always applied to a "safe" school. I told the story and they said orientation starts Monday, better get up here.

Slayton: That's quick.

Berson: Right, just like that.

Slayton: Wow. That's amazing.

Berson: So I did, and I did physics, and I was on the dean's list.

Slayton: That was 1964?

Berson: Yes. It was 1967 when I got out.

Slayton: So did you have much exposure to computers while you were in college?

Berson: Oh, yes; a lot. A lot. It goes back, actually, to when I was in elementary school. My dad had a friend called Norman Friedman who worked at Bell Laboratories. Bell Labs at that time was on West Street in Manhattan, before they moved to New Jersey. Norman kind of took me under his wing, and he saw that I was interested in science. And he raided the Bell Labs junk box and came up with a whole bunch of relays, stepping relays, switches, lamps, and a power supply, and a bunch of interconnects. And so while I was still in elementary school, I was building adders; adder circuits out of telephone relays.

Slayton: That's awesome.

Berson: Right. And arguably, that's a computer.

Slayton: Yes. What did your dad do, by the way?

Berson: My dad was a civil servant; he worked for the New York Housing Authority.

Slayton: I see. But he had friends at IBM.

Berson: No, this was Bell Labs.

Slayton: Bell Labs; I'm sorry.

Berson: It was actually a friend from summer camp, where he had been a counselor and Norm had been one of his kids, one of his campers at summer camp.

Slayton: So that was your first exposure, and then you used computers quite a lot in college, as well?

Berson: Well, they ran all the computers in college. When I was at M.I.T., I think the TX0 was there. I didn't get any time on it. When I went to Oswego, there was a 1401, I want to say; and that's probably the first program I wrote, was in FORTRAN. The course was called computer programming, and what you had to do was matrix inversion, and matrix multiplication. It was a statistics course and it was in FORTRAN. I almost flunked that course because you had to write flow charts. And I could see the structure of the program and everything was perfectly clear to me and so I just wrote the program, which worked, but I wouldn't draw the flow chart.

Slayton: Wouldn't waste your time.

Berson: That's right.

Slayton: That’s interesting.

Berson: I remember taking an incomplete in it and then had to go back and draw the flow charts.

Slayton: Wow, that’s hard core. So when you were at M.I.T., you probably; did you even know about MULTICS going on at that time?

Berson: No. I’m not sure; I mean, I’m talking about 1962; I got to M.I.T. in 1962. I think it’s before MULTICS, right?

Slayton: That’s right. It was barely started in 1964. I was thinking 1967, but you were gone by then, right?

Berson: Right.

Slayton: So what happened next? Can you sort of walk me through your professional trajectory? You graduated in 1967?

Berson: I graduated in 1967, right. So one of the recruiters who came to campus was IBM, and they said okay, take this test, called the DPAT, Data Processing Aptitude Test. So I took the DPAT and the guy calls me up and says wow, you’ve scored off scale on the DPAT. My minor at Oswego was education, so I said, off scale in which direction? And he said, hunh?

[LAUGHTER.]

Okay, and then he said oh, in the high direction; and he says we want you to be a customer engineer; systems engineer, yeah, systems engineer. I said, does that have to do with; do I have to meet with customers? He said yes. I said I'm not interested; find something else. He called back the next week and said we'd like you to go to the IBM Research Center in Yorktown Heights for an interview. Okay, so I went to Yorktown Heights and got a job there as a programmer; I was an assembly language programmer, and that was the first real computing job I had. Yorktown Heights is ***fantastic***. So I'm talking about 1967, I got there; I left there in 1969; but there were so many smart people around, such interesting things going on. There was the work that would become DES; there was APL, this was Ken Iverson; do you know APL? It's a programming language, right, that was described modestly by Iverson as an improvement upon mathematics. And there was CP67 CMS. Okay, so IBM at the time had this huge programming project called TSS, it was one of these 10,000 man-hour — that's what we called them then — 10,000 man-hour software engineering projects to try to make a time sharing system so you could interact with the machines.

Slayton: Was this as part of OS/360?

Berson: OS/360 existed but this was separate from 360.

Slayton: Totally separate, okay.

Berson: OS/360 was a batch processing.

Slayton: Right, but then I thought that they tried to make it time sharing, sort of at the end, because they saw the demand. I mean, they lost a contract to M.I.T. for [pause]

Berson: I don’t know about that. This was the research division; we didn’t know about contracts.

Slayton: That’s interesting; so it was totally separate.

Berson: But the IBM Cambridge Scientific Center had done a hack where they made something called CMS, the Cambridge Monitor System. So some 360s — not all — but in particular the 360/67, three-sixty-slash-sixty-seven, had paging registers. And so Cambridge Scientific Center made CMS, Cambridge Monitor System, which served as a program on the real 360/67, and would create virtual 360s above that. And then CMS was a single user interactive computing environment that ran on a standalone 360, so you slide in CP67, Control Program 67, right? And on top of it you build a bunch of CMSs, and each CMS you boot, I mean, you serve one user. And bravo, I mean, it was time sharing. Many people could interact with one 360.

Slayton: How many?

Berson: I don’t know but tens, perhaps hundreds. I mean, 360/67 was a honking machine for its day, it really was. That was no low end 360.

Slayton: Interesting.

Berson: So I was involved a bit in that program; particularly, I was interested in the security properties of these virtual machines, and I was exploring how to get from one virtual machine into the other. This was Bootleg Project, it wasn’t my assigned duty.

Slayton: But it was interesting.

Berson: Right. And I recall I discovered that by reading a tape backwards across a page boundary, you could bust out from the virtual address space to the real address space because of the way the IBM 360 channels work. And so all you needed was to read the tape backwards. It didn’t have to be a real tape, it could be a virtual tape and you got a virtual machine escape. That was maybe my first computer security hack. Although I must say, I need to tell you that I’d been hacking stuff for a long time. I mean, as a kid, I grew up in a house full of secrets, and I was always interested in technologies for breaking secrets and for making secrets. So by the time I was in sixth grade, I knew about locksmithing, I knew about telephone tapping, I knew about burglar alarms, I knew about secret hiding places, I knew about invisible inks. So I’d always been interested in that stuff.

Slayton: Why were you living in a house full of secrets? Was it because of your dad's work?

Berson: Because of my family; no, just family secrets. Every family has them. But for a kid, you don't know that any other family has them, they were just mysteries around the house and I was going to solve them.

Slayton: That's awesome; I mean, some kids don't even know there are secrets. I didn't discover there were secrets until I was in my 20s.

Berson: Really? That's very interesting.

Slayton: I assumed there were things I didn't know, but I didn't realize just what I didn't know.

Berson: Right. So I was into that. Actually, another person I met through the camp my dad worked in the summers was an amateur radio operator. I got interested in amateur radio and when I studying the Morse code, I came across encrypted fleet broadcasts. The Navy used to send its fleet broadcast in CW, Morse Code, and some of it would be encrypted and some of it would be plain text, in those days. So I was using that for practice. I could easily copy the clear stuff. The cypher text was very curious to me so I of course set out to break the code, which I didn't do. But I can recall doing frequency analyses on Navy cypher texts, when I must've been 11 or 12 years old.

Slayton: So you must have had a really strong math program.

Berson: I don't think that I had a particularly strong math program at all. It just; some aspects of it just came easily to me, particularly what we now call discrete math.

Slayton: So did you continue hacking when you got into college?

Berson: No, it kinda went away; so, for the moment, did the radio. I mean, I discovered physics; I discovered sex; either of which is arguably more interesting. [Laughs.]

Slayton: Right, when you're 18, sex is a lot more interesting.

Berson: So I kind of let that drift away for a while. I worked at IBM for two years. I met a guy at a cocktail party; says what do you do? I said I'm tuning operating systems for performance, and he says what are they paying you? I told him, and he says I'll pay you double and you can do the same thing for me.

Slayton: Did he know anything about your hacking?

Berson: No, I don't think we discussed that.

Slayton: Was there anybody else at IBM who was hacking with you, or was it just your own thing?

Berson: No, it was just my own thing. I would read the manuals cover to cover, and IBM had wonderful documentation in those days; and I would read them cover to cover. The places where it said don't do this, don't do that; the parameter has to be like this. Those edges were always curious to me; I was always spotting, you know, what's the edge of the envelope here? And then writing little programs to see what happens if you step over the edge of the envelope.

Slayton: That's fascinating. And did anybody ever clue into what you were doing at IBM?

Berson: Oh sure, my manager. I wasn't keeping it a secret or anything. I said oh, this is interesting; why does that work? Let's fix it or write in the manual, don't read tapes backwards over page boundaries. [Laughs.]

Slayton: So, did your manager see that as a contribution to IBM or just something to be tolerated?

Berson: My manager, I think, was pleased by my performance and was disappointed when I said I was leaving.

Slayton: Okay. But he didn’t really care; the hacking, he was indifferent to.

Berson: Yes, the hacking, he was indifferent to, yes.

Slayton: Didn’t really think security was a problem?

Berson: No, and perhaps it wasn’t a problem in those days. I went to work for that guy who hired me; he was bankrupt within the year.

Slayton: What year was that?

Berson: He turned out to be a con man. It was 1969. We had two clients; a couple of clients; First National City Bank, AMEX, and Loeb Rhoades.

Slayton: And what was the name of the company, such as it was?

Berson: Never mind. It was a small consulting thing. It was my first model of how to be a consultant.

Slayton: Or not to do.

Berson: Right. I called up City Bank, and I said you know that project we're doing for you — which had to do with traveler's checks, by the way — they said yes. I said it's not going to finish because this guy has just gone bankrupt but if you'll pay me a third of what you were paying him, I'll finish it for you. They said sure.

Slayton: Wow.

Berson: So I worked for them for a while and that was the first time I was a consultant in the consulting business. I called my business Computerniks. C-O-M-P-U-T-E-R-N-I-K-S, Computerniks.

Slayton: How interesting, that term went around quite a lot. I don't think I ever saw it used before; I mean, I don't think I saw your company.

Berson: It was just me.

Slayton: Were you working for somebody other than City Bank?

Berson: I was trying to.

Berson: But they were your main customer?

Berson: They were my main customer, right. But then I got another customer, again through this guy for whom I had worked, and for whom I didn't realize the depths of his depravity, at the time. He had left the United States and gotten a job doing; he had gotten a job for ICL. ICL was at the time, the largest computer company in the world not controlled from North America, International Computers Limited. And his job for them was, they needed a guess what? A real time time-shared operating system for a particular customer, who was the Royal Air Force.

Slayton: Interesting. You said a real time shared operating system?

Berson: Real time time-sharing…right; the vision was that people globally would be able to dial into this system and make a query. It was a logistics system, so the canonical query was somebody in Cyprus wants 20 pairs of airman's socks size 10. So you could query the system and the system would find out where those socks were and what transport was available to get them to him.

Slayton: Yes.

Berson: I knew about operating systems; I knew about interaction; I knew about all this and so I went to work for this guy. Again, I spent a year, as Computerniks, I spent a year commuting between the U.S. and U.K. I'd spend a month in the U.S. at home, and a month in the U.K.

Slayton: What year was that?

Berson: It was 1970-71. Dorothy said to me, this is crazy, we have two kids. Decide where we're going to live.

Slayton: You already had two kids.

Berson: Yes, my kids were born in 1966 and 1969. So I said well, we lived here, let's go live there. So I got a contract from ICL and we moved over there. It was a two-year contract, and we did that work. While I was in the U.K, I discovered that one could do a Ph.D. for £27 a year in tuition and fees. You had to register as a part time student and all you needed to do to get admitted was to have a professor say yes, he/she can be my student. So I had lined that up; we finished the work with ICL; and it kind of came to a; it wasn't a success and it wasn't a failure, it was just a program. Oh, because this was for the ICL Spectra 7; no, ICL System 4; whatever they called it, which was a rip-off of the RCA Spectra 70, which was a rip-off of the 360, okay? So this should interest you as a software engineer kind of person. So when it went from the 360 to the Spectra 70 — I may have these numbers wrong — and it went from that to the ICL series whatever-it-was; at each point, people looked at the specs and said I don't understand what that is for, I'll just leave it out. So subtleties of how to program the thing just kinda went away because the people who imported it couldn't figure out what it was used for and it added cost and complexity.

Slayton: So the code actually got reduced during that?

Berson: Yes, the semantics of operations got reduced. And because I was one of these performance-tune-operating systems kind of guys, I knew all the subtleties of the semantics of instructions and when to take advantage of them. This is working far below the floorboards, doing things with bit hammers that probably shouldn't be done because they're not sustainable; they're not portable. But yet, again, coloring slightly outside the margins.

Slayton: Right. So you were doing this at…okay, I have two questions. You were just sort of doing this on the side out of interest to see what could be done with this system? This is basically hacking the system.

Berson: No, that was my main [pause]

Slayton: Or that was your work.

Berson: That was my work; that was a piece of my work at IBM Research.

Slayton: Okay, I see.

Berson: I wasn’t a researcher at IBM Research, I was an assistant programmer. I mean, I had a bachelor’s degree in physics. They didn’t hire me to be a researcher; they hired me to support their work.

Slayton: Support the researchers, yes. That’s interesting. So when you say that you were working beneath the floorboards, you’re talking about sort of interacting very closely between the hardware and software?

Berson: Absolutely.

Slayton: So you’d had a lot of experience of tinkering with hardware, was that something that helped sort of get you to that point?

Berson: It’s not really hardware. I mean, the instructions set exported by a computer, even as old of a computer as a 360, it is just an expression of the micro code. I mean, the real hardware, the gates, are below the micro code, which is below the assembler language interface. So I wasn’t beyond altering micro code to make my instructions work better.

Slayton: Okay. Interesting.

Berson: I mean, it’s turtles all the way down. I mean, it’s abstraction, abstraction, abstraction, abstraction; and somewhere at the bottom are the laws of physics.

Slayton: Really far down on the bottom. Okay, so backing up, this work that you were doing for ICL; it was after you had left IBM then that you were working [pause]

Berson: I left IBM, I went to work for this guy in New York City; and then as Computerniks; and then as Computerniks I took a contract with ICL. And then when that was over, we bought a Volkswagen van in Amsterdam; we had it converted into a camper; we put both kids in it and we spent eight months driving across the face of Western Europe, rarely getting past the first vineyard in the morning.

Slayton: That's fantastic; nice.

Berson: Alright, so we finished that; we were in Rome, I guess; we wanted to go to Greece, but it's the colonels or something bad is happening in Greece, and travelers' tales, coming out of there, were poor and this wasn't the year to go there. So we go back to London; it took us six months to drive to Rome and maybe two, three days to drive back; stop off at Queen Mary College, where I had befriended this professor named George Coulouris, and I said hey George, will you still have me as a Ph.D. student? He said sure, and then I became a graduate student in London.

Slayton: So what did George, what is [interrupted]

Berson: Coulouris, C-O-U-L-O-U-R-I-S. His father, an actor by the same name, you may know him; he had many roles on the West End stage; but you may have seen him in the movie of Agatha Christie's "*Murder on the Orient Express*."

Slayton: Oh, I haven't seen that, but yeah, that's fascinating. So he was in computer science?

Berson: Yes. He had; this was 1971-72, alright? In George's lab; George's lab was very far forward-looking. William Newman, N-E-W-M-A-N, was a member of it, and William spent time at Xerox PARC. George was running UNIX in his lab; he had written his own little interactive time sharing system; he had mice; he was interested in tablet input and computer graphics . . .

Slayton: Wow.

Berson: . . . amongst many other things, including the beginning of what we now call distributed computing.

Slayton: Was his work all funded by the government?

Berson: I don't think so; I don't know where he got [pause]

Slayton: In the U.S., you know, IPTO was funded most of that; or I shouldn't say most of…a lot of it, the very important work.

Berson: Right. So in 1972, whatever it was, I began to be a UNIX user. No, I don't remember the version. I mean, it's what there was. So I got interested — see if I can get this straight — I got interested in pattern recognition and I wanted to do signature verification stuff, because he had these tablets so I wanted to do real time signature verification. You write your signature on the tablet and the system says yes or no. George says no, that's not interesting; I want you to do handwritten character recognition. So handwritten character recognition's a much tougher problem because it's 36 or 40-class classifier that you have to separate each character from all the others, right?

Slayton: I see what you're saying, yes.

Berson: In signature verification, you have a hypothesis, this Rebecca's signature, yea or nay. But in the other thing, you say oh, this is Rebecca, what do we know about her handwriting; what's she writing now? So I worked on that; I built programs; I did a lot of Bayesian statistics, what we call now "machine learning." And finally struggled my way through to a Ph.D.

Slayton: And did you maintain your interest in security and hacking, at that point?

Berson: Well, yes and no. I wasn't working on; no, I didn't do any hacking. I was interested in cryptography. There was an issue of custody of recordings, such as you're making now, by the police, to see whether the recording had been altered or not. So somehow, this problem got presented in our lab and the solution I had was to actually digitize the thing. Take a check sum over the whole file and sign it with a secret key. DES was just coming out; I mean, I finished my Ph.D. in 1977; DES, I think was 1976; was just coming out. I was aware of it; I had gone in 1975 or 1976, NIST had a workshop. I came from England to NIST, to the DES workshop. And I proposed the solution to Scotland Yard, who didn't buy it. In retrospect, they didn't understand it; I mean, it was like gobbledygook to them.

Slayton: The solution being DES and even checks [pause]

Berson: Digital; it's actually a secret key digital signature over the file, which they can then demonstrate to some judge that it hadn't been altered. Can you imagine walking up to some judge and saying, your honor, because of quadratic residue, this can't…you'd get kicked right out of court.

Slayton: That's interesting. Now, were you in touch with Marty Hellman at this time?

Berson: No, but I had been in touch with Marty. How? I'll tell you. One day in the IBM Yorktown cafeteria, I'm having my lunch and there's Marty Hellman. What are you doing here? What are you doing here? So we both independently were working; we overlapped slightly at IBM Yorktown Heights [pause]

Slayton: Oh, I didn't know that.

Berson: Okay, so that was the second time that we overlapped. The third time is coming up. Because when I finished, my daughter was 11 and was asking me questions that I didn't know how to answer. At 11, you take; British schoolchildren then took this test called the 11 Plus, which decided where they were going to go to school, which essentially decided whether they were going to university, who they were going to marry, what kind of families they were going to have. And I just didn't know what to do here, and I decided that if someday she wanted to be an immigrant she could be, but we weren't compelled to be immigrants, so let's go back. So we came back and looked around for a job. There were two jobs I looked at. I had been to a NATO summer school on pattern recognition, during the course of my Ph.D. And while I was there, I met a fellow whose name I forget now, who worked for Ford Aerospace and Communications Corp., which is here in Palo Alto, and he had left me his card or I had collected his card. So when I was looking for a job, he's one of the people I called up because I thought I'd get maybe a pattern recognition job. So came back and looked for jobs, and quickly got two offers. One from IBM, who would be very happy to have me back; and the other from Ford Aerospace out here in Palo Alto.

Slayton: IBM was back in New York?

Berson: Yes, exactly. I could've gone back to research as a researcher.

Slayton: Imagine that?

Berson: But I looked around the Hudson Valley and the Silicon Valley. I had never been to Silicon Valley, it was like a third country. Again, I mean, there was New York, where I grew up; there was England where I had lived for seven years; and Silicon Valley! I looked at Silicon Valley and said wow, what a wonderful place for a nerdy guy! What a place to be in business. What a place to raise kids. People were saying, don't worry if this job with Ford that you haven't taken yet doesn't work out because you'll find another job by the time you reach the corner. And you look at all the spec sheets for gadgets, and instead of writing away for more information, which is what we did in those days — the internet had not yet; well, the Web had not yet been invented — you just hop in your car, drive five minutes, and go speak to the engineers who wrote the spec sheet. So it was kind of a no-brainer, but Dorothy and I did this thing, like an A/B test. We left the kids with my folks and Dorothy and I went to IBM land, and we went to here. Like a flip flop, and it was perfectly clear where we should live so we moved here. On the flight out when I was going to my interview at Ford, I pick up *Scientific American* on the airplane, and there's a Martin Gardner article describing public key cryptography, and saying Marty Hellman's at Stanford.

[Laughter.]

Okay, that was the third connection. So I reconnected with Marty while I was out here, because I had no idea where he'd gone.

Slayton: Can I just back up and ask you a few questions?

Berson: Sure.

Slayton: We're backing way up; when you were working on ICL for the Royal Air Force, was security a big concern?

Berson: No. It may have been a concern but it wasn't part of my job; nobody ever spoke to me about security. It may well have been; don't forget from their point of view, I was a foreigner working on their program. So I may not have been exposed to those security oriented issues.

Slayton: Got it. The second one was, when you were in grad school, did you ever actually take any classes on cryptography or was it just self education?

Berson: No, no classes. No classes in British education, then, for a Ph.D. I don't know how it is now, but then there were no classes. You were supposed to learn stuff, but this is a research degree and you're supposed to figure out what it is you need to learn, figure out where you need to go to learn it, and learn it.

Slayton: So did you work with any professors in your department doing cryptography?

Berson: No. Not at all. They were doing; first of all, the department was called Computer Science and Statistics. The only thing you use computers for, of course, was to convert matrices. George's lab, with a computer, which he called the Computer Systems Lab, I think; was a refreshing change from what everybody else was doing, which frankly, was computer science and statistics. At QMC and in George's lab there was very interesting programming language work going on; very interesting architecture work going on. I'll tell you a piece of each, if you're interested.

Slayton: Sure.

Berson: Alright. The programming language work; they had this guy called Peter Landin, who was an ALGOL guy, and he had just written a paper called, *"The Next 700 Programming Languages."*

Slayton: That's ambitious.

Berson: Well, it's ambitious, but it's actually what he's saying is based on what we know now, you can make 700 languages and it's not going to make a difference at all.

Slayton: Got it.

Berson: So he was an interesting fellow. In the architecture domain, there was a man called John Iliffe, I-L-I-double F-E, I think — may be wrong — and John had two architectures that interested me greatly. One was a content addressable memory and the other was what they called a Basic Language Machine, but it had nothing to do with the language BASIC, it had nothing to do with Dartmouth BASIC. What he meant was basic, in this sense. And this was a hardware tagged architecture, so every word in this architecture would have hidden bits that would describe the type of thing that was in the word. This is fascinating from a security point of view. You can prevent, for instance, counters being overcast as pointers into memory.

Slayton: Right.

Berson: And we saw that later; there were some tagged architectures made. I think Burroughs made a tagged architecture. And Bob Fabry may have had one.
And SRI had this PSOS, provably secure operating system. Do you know the famous typo about PSOS?

Slayton: I don't know about a typo; I know you said something about "potentially provable."

Berson: No, so if you're typing "provably" and the "v", which is an uncommon letter, down there right? You could hit the "b" next to it. And so . . .

Slayton: Probably. [Laughs.]

Berson: That’s right. So it appeared in some biography as a “Probably Secure Operating System,” which was copied over and over for some time. Anyhow, so it required a tag; in one implementation it could’ve used a tag architecture. These ideas were out there in the world. Where was I; other guys in the department; no, nobody was teaching about cryptography. Of course, there was a lot of cryptography that was being done in the U.K., but it had not yet seen the light of day. The ultra secret of Bletchley Park was still being kept. There are people with whom I would’ve loved to speak, but even if I had known of them, I don’t think they would’ve spoken with me in that era.

Slayton: So you just picked up cryptography just by reading what you could, pretty much?

Berson: Yes. I read Helen Fouché Gaines, and I read David Kahn.

Slayton: Kahn inspired a lot of people.

Berson: I may not have Gaines’ book here anymore; it was essentially hand ciphers; you know, paper/pen kind of ciphers.

Slayton: So I presume at Ford, that’s where you got involved with the kernelized secure operating system?

Berson: Yes. At Ford, I mean, it was the Cold War; Ford was in the defense industrial complex; it was an aerospace contractor; we were building lots of systems for the military and the intelligence community. My job at Ford; Ford had this contract from NSA, from Dan Edwards, who was one of their guys, Dan was; the contract wasn't from him but he was the guy in charge of technical; the CTO, Contract Technical Officer I think they were calling it. There was somebody to run the business and somebody to run the technology. Dan was running the technology. And they had this contract to build KSOS, the Kernelized Secure Operating System. This was supposed to be built on a PDP, what, 11? I think. Yes, on a PDP-11, DEC PDP-11. We were supposed to build this kernel, and it was supposed to be small, and complete, and correct. You'll recognize Roger Schell in here, right? Small, complete, correct kernel, which was supposed to be totally proven to be correct against the Bell-LaPadula security model. And then on the kernel we were supposed to write a UNIX emulator; and then on the UNIX emulator, you're supposed to be able to run UNIX programs. There were a bunch of good system programmers at Ford but there was nobody who was willing to take the challenge of proving a program of considerable size. So I was the new guy on the block, and the rumor was you couldn't do this, so that kinda had my name all over it. So I undertook to get the formal specification written of the kernel, and to get the proofs to go through. And we had a subcontractor for this, SRI.

Slayton: Oh, SRI was a subcontractor.

Berson: SRI was a subcontractor to us on this, and Peter Neumann and Rich Feiertag were the guys who were over there on the SRI side. And they had a set of tools called HDM, Hierarchical Development Methodology, some of which worked very well and others which turned out to be [pause] incomplete. [Laughs.]

Slayton: Okay. Now, originally you went to Ford, that would've been in 1976?

Berson: Let me check this out. 1977. I remember I moved to California in August of 1977, and I thought; and everybody was smiling; and I thought what a happy place because everybody's smiling. It turns out everybody's smiling because the sun is always shining. I had lived in England where the sun never shone — almost never — and I come to California it was like a total different world.

Slayton: I bet. And very different from New York, as well.

Berson: That's right.

Slayton: That's awesome.

Berson: 1977.

Slayton: And did you go to specifically work on KSOS?

Berson: That's right. I was hired by a man who had KSOS; that's right, at Ford, I mean, you have to be sold. And KSOS was paying my money. That's right, I was working on KSOS. I got involved in other projects while I was there, but KSOS was what I was hired to do.

Slayton: Was that the first you had known, then, of Bell-LaPadula?

Berson: David Bell and Len LaPadula.

Slayton: Was that the first you'd heard of that model?

Berson: Sure. I mean, it's a multi-level security model; I had never thought about multi-level security before. To me, things were either secret or they weren't secret.

Slayton: Right. Did you look at all at James Anderson's work at this time?

Berson: Well, Jim Anderson was; that's why I first; I first met Jim Anderson on that project; I first met Willis Ware on that project because they were, of course, very interested. I first met Roger Schell on that project; I first met Brian Snow on that project; it was a nexus of computer security research. It really was a leading edge of computer security research. In this nation, we had marvelous computer security research up until the time the Orange Book was written.

Slayton: Interesting.

Berson: Maybe we'll come to that, or are we there?

Slayton: Well, yeah, we may as well; I mean we can go back but I was going to ask you about the Orange Book.

Berson: So we were learning about how to build secure, high assurance operating systems. And what happened is, in the attempt to codify in Orange Book is what we'd learned, the nation instead ossified it. I mean, I was there or nearby when the Orange Book was written, and I know for sure it wasn't delivered on top of Mount Sinai on tablets. It was written by guys who struggled to hack it out but yet it became accepted as a DoD standard and going forward, things that supported the Orange Book model were funded and things that didn't weren't. I think a lot of new and wonderful computer security research withered because of the Orange Book.

Slayton: Can you give me an example of something?

Berson: PSOS.

Slayton: Okay. So PSOS didn't [pause]

Berson: Non-interference.

Slayton: Interesting. I would've thought that PSOS would have been seen as something in support of the Orange Book standards, but maybe not.

Berson: Maybe it was, maybe PSOS just died on its own. But anyway, we forgot a lot of what we knew about computer security and instead we changed it from being a research area into being a standards compliance area.

Slayton: That's really fascinating.

Berson: I had the occasion in the 1980s to go to a conference in China that reminded me of sort of pre-Orange Book U.S. papers and I thought to myself, these guys are picking up where we dropped; picking up the ball we dropped.

Slayton: Fascinating. When you say that it ossified it, are you also referring to…one of the criticisms that's been made, is certainly, I think, the Orange Book that there's been a lot of emphasis — and research that came from that — lot of emphasis on the Bell-LaPadula model, but not on the Biba model.

Berson: Okay, Biba. Ken Biba, was at MITRE and then we hired him at Ford Aerospace [pause]

Slayton: Oh, you did; okay.

Berson: Yes. And then a bunch of us left Ford Aerospace; five of us left Ford Aerospace and we formed a company called Sytek. Biba was one of them, I was another.

Slayton: Ah, I didn't know that. Who else?

Berson: You know those two; the other three are Mike Pliner , who had been the manager of the group at Ford Aerospace, he was the man who hired me at Ford Aerospace. P-L-I-N-E-R. A guy named Jack Goldsmith, dead now, alas. And a guy named Sammy Kroll, K-R-O-double L; although Sammy was just his nickname. I think his name might've been Robert, or Bob; but everybody just called him Sammy.

Slayton: So the five of you formed Sytek, was that 1979?

Berson: That was 1979.

Slayton: What prompted you to start that?

Berson: A couple of things. Working at Ford Aerospace was working for Ford, and Ford Aerospace was a wonderful company while we were there; our department of it was great. But it was like working on an oasis in a desert, and every now and then, a wind would blow up out of Dearborn and you'd wind up with sand in your teeth because of personnel policies or [pause.]

Slayton: I see. Was it related to the fact that you were doing research and not sort of Ford's more production [interrupted]

Berson: No, I don't think so. I mean, we could've; we were early into networking, we really were. We had a proposal to put a local area network on an automotive frame.

Slayton: Oh, interesting.

Berson: Alright. I mean [pause]

Slayton: That's fascinating.

Berson: And we had an opportunity to do a piece of work that Ford wasn't going to do, and that we wanted to do.

Slayton: And that was a networking project?

Berson: It was a networking project. So we went off to do that, and we finished KSOS.

Slayton: After you started Sytek.

Berson: No, no. We had finished KSOS, at least to the point where you could see that you could make the proof but the system wasn't going to work. It worked logically, it just didn't work from a performance point of view because the way it was, there was one set of registers, I think, in the PDP-11 and when you cross a domain from user space to emulator space to kernel space, it was two calls, two register swaps going in; two coming back out; and it was glacial. Very unsatisfactorily glacial and I think; and that was only part of the problems. I mean, we also made what is called Error 33. Do you know Error 33?

Slayton: No.

Berson: It's relying on one too many miracles. [Laughter.]

Slayton: Why Error 33 and not 34 [pause]

Berson: I don't know; and not Error 1. Right? So we tried to write it in Euclid. Why Euclidl? Because it had proof rules and we had to have proof, right? But it was a new language with a new compiler; so trying to write a new operating system in a new language and using a new compiler . . .

Slayton: Right. Too much new…

Berson: . . . wait, and I'm not sure it was Euclid. It might've been modula, M-O-D-U-L-A. Anyhow, both of those, I think, are Klaus Wirth, Niklaus Wirth. You know of him? Niklaus Wirth languages.

Slayton: Okay, right.

Berson: It was a research project and we learned a lot from it; and I think a lot of what we learned wound up being in the Orange Book.

Slayton: Okay, that was my next question; what do you think the impact of it was?

Berson: I think it was tremendously impactful.

Slayton: On things other than the Orange Book, or primarily on that?

Berson: Well, primarily on the Orange Book but we learned, again — because I think people had learned early — how difficult it is to prove even a very simple property. And Bell-LaPadula was a simple property, about a complex object. Dan Edwards used to talk about a Coke machine, and the difficulty of proving the correctness of a Coke machine. Not one of these Coke machines that has 105 flavors, that hadn't been invented yet. His model of a Coke machine was you put your nickel in, and you press the lever, and the Coke comes out. And he used to say how difficult it would be to prove the correctness of even that.

Slayton: That’s interesting. Yes.

Berson: So one of the things we were grappling with, that the Orange Book ducked, was what happens when you hook this thing up to a network, because we knew it was going to be hooked up to networks and Orange Book is absolutely mute on that. So hookup security, composition of components, all that stuff was just beyond the reach of the Orange Book. By the time that the Orange Book came out, it was unresponsive to the requirements of modern systems.

Slayton: Was it something you were talking about when you were working on KSOS, because I mean, networking was already starting to happen?

Berson: Yes.

Slayton: So you talked about it and, in fact, you didn’t think it was sufficient to deal with this new context?

Berson: Absolutely. One of the projects we got involved in, and didn’t win — luckily, I think — was something called BLACKER. BLACKER was a network encrypter. So yeah, we knew about networks; and we were interested in BLACKER because to our minds, you know, a network protected by BLACKER would be connected to KSOS.

Slayton: I see. So you sort of thought BLACKER will handle the network and we'll handle KSOS?

Berson: No, BLACKER will handle the communication security, which all BLACKER was meant to do was handle communication security; but somehow, you know, you have this KSOS machine here and it's going to be talking to some other KSOS machine; a multi-level machine here, a multi-level machine there. Well, how do they talk to one other about classifications, and security markings, and how does data flow? Even if you can work that out, how do you say anything at all about the correctness of that composition?

Slayton: So do you think if the Orange Book hadn't come along that you would've sort of continued on in that line of questioning?

Berson: Perhaps. And, in fact, I think parts of NSA knew that it made a mistake. I think they wanted to get the Orange Book out, but I think to be fair to them, their research program probably said right, we'll deal with networking next. So in 1985 — that's actually a long time later — in 1985, there was this Invitational Workshop on Network Security in New Orleans. Have you seen this picture?

Slayton: I haven't seen it, no.

Berson: Alright. So to talk about network security is Vint Cerf, it's Steve Kent, Brian Snow, Sheila Brand, Dorothy Denning, Steve Lipner, Morrie Gasser, John Rushby, Dan Edwards, and other people who I can't see at the moment — got together and spent a long — David Bailey, did you hear his name yet?

Slayton: No.

Berson: From Los Alamos.

Slayton: Okay.

Berson: Ruth Nelson. Terry Benzel, Jon Millen's in here somewhere, Clark Weisman — I think he wound up winning BLACKER — I mean, BLACKER is important. BLACKER is going to become important in a second.

Slayton: Okay. I think that's most of my questions about KSOS. You want to go on and talk about BLACKER, then?

Berson: Yes. So we went off to do this networking project at Sytek and soon discovered a need for local area networking, because we had VAXes, and the way to do things in those days was to pull serial cables from the VAX to glass teletypes, but we knew about networking and we decided we'd design and build a local area network to hook up our glass teletypes to our VAX. We decided to do this over cable TV-plant kind of transmission line; over coaxial cable because it was cheap, it's commodity thing. In order to do that, we built modems and we built frequency agile radios, so we'd build different frequencies in the coaxial cable. The way that cable TV works is channels go out, there's a head end that repeats them, translates them to another frequency, comes back down, so the whole area is covered after passing through the head end. MITRE had actually tried to do this, and they built kind of a military-grade thing with military-grade radios; very expensive and it worked, but uneconomically. We, instead, got citizens band radios and improved the filters in them. This is back to my amateur radio experience. We improved the filters and we wrote NetBIOS, and so we became a local network company. Out product, LocalNet, operated on multiple frequency pairs, and shared each of them using CDMA/CD. But I was interested in security. So one of the things I designed for this; also when you're broadcasting over these cable TV plants, anybody with a radio that plugs into the cable anywhere can copy all the traffic and that wasn't good enough for me. So I decided that we would build an end-to-end cryptosystem at the session layer, into this local area network. And I designed such a system, and I had to be very careful to avoid using the BLACKER protocols because I knew the BLACKER protocols from when we had bid on the system. They weren't ours, and they may or may not have been classified. So I had reinvent different key management protocols, and we put it in there; worked like a charm, but we made the business mistake of making it an extra cost option; $80 extra. I really think that security features should not be optional.

Slayton: Did you talk about whether you should make it optional or did you just assume it should be optional at that time?

Berson: I wasn’t a business guy yet, when we went to Sytek. I became a business guy while I was at Sytek, that’s when I really learned about business. When we started Sytek the only one of us who had a suit was Jack Goldsmith, and we let him negotiate all the contracts to decide all the money stuff.

Slayton: What was the business model that you were going with? Who did you expect to be your clients or customers?

Berson: At Sytek?

Slayton: Yes.

Berson: We built this equipment and sold it. We sold it direct.

Slayton: To anybody, just any person, not companies or [pause]

Berson: To companies, to campuses; many campuses were already wired with cable TV plant. So we had a large installations, for instance, at Brown University. We had a large installation in the FBI headquarters.

Slayton: So you saw a number of organizations that were starting to network and you knew that you’d have market there?

Berson: That's right. And I was also running consulting on the side, as cash cow for Sytek. So this end-to-end crypto system was one of the interesting things that I did at Sytek. It was DES based, secret key, KDC, Key Distribution Center, and the real mistake we made with it was to make it an extra cost option.

Slayton: And did you expect, originally, that organizations would buy that?

Berson: I thought some would and some wouldn't. It wasn't because of market pull, it was my idea to put it in. In retrospect, we should have put it inn standard, which would have required all of our competitors to also come up with a crypto system and been a service to customers everywhere.

Slayton: Would it have made a critical difference in terms of pricing? Would it have made [interrupted]

Berson: We had to buy an extra chip at that time; we had to buy the Western Digital DES chip, and so it would've driven up our costs slightly.

Slayton: More than your competitors?

Berson: Well, our competitors were just getting organized. They had names like Ungermann-Bass, 3Com, and we were all competing for; we had different technologies and we were competing for the local area network customer. If we had made this security, the end-to-end security built in, we would've required all of our competitors to do that. It would've been a checklist item.

Slayton: You think the people would've been willing to pay for it? Even if your competitors could've underbid you.

Berson: If they had no choice; if we said look, this comes with it. I don't know; I think other people would've said we don't want this one, it's not secure.

Slayton: You think they would've been willing to pay more, if your competitors underbid you without it?

Berson: Yes, but we'll never know.

Slayton: That's the thing that people always say is that we can't force people to buy it because then we'll miss out because our products will be too expensive and we'll lose out to competitors.

Berson: It wasn't much more $80 more a box. That was the charge; that was the price, okay? The cost difference must have been $20-30 to us.

Slayton: And what fraction of the total, is that like a one percent difference or is that a 10 percent difference?

Berson: I don't recall.

Slayton: But not large, probably. It doesn't sound large today, but of course that was 1980.

Berson: Consider Skype. Skype has a crypto system built in, also at the session layer. Most people don't even know it's there. I mean, it's just part of it.

Slayton: Yeah. It's cheaper now than it was then, right?

Berson: For sure, there were no extra chips.

Slayton: I'm not trying to argue with you at all, I'm just trying to understand what the market was like at the time.

Berson: There's two other things I want to tell you about at Sytek. One was we had a research contract to build a guard. Sometimes these are now called cross domain solutions; there's a high side and a low side, and traffic wants to flow from the high side to the low side and needs to be inspected to be sure it meets certain characteristics. And the research contract was to build a guard which we could prove was correct. Alright.

Slayton: And who was the customer?

Berson: An intelligence agency. So based upon what we knew about how difficult it is to prove things, we decided to throw processors at the problem. So instead of having one processor talking to the high side, making the decision, and then talking to the low side, we bought three single-board computers, which were just coming out; connected them together on a bus, and cut lines on the bus so that the talking to the high side guy could only talk for the high side and he had a memory. The guy in the middle could reach into the high side's memory and pull stuff out of it. The guy talking to the low side, he could only talk to the low side but he couldn't reach anywhere. The guy in the middle could push stuff into the low side's memory, and we knew that because we could look on the PCB traces on the bus and see that we cut them. Now, all that we have to do is prove correctness of the very limited program running on the middle guy. He doesn't have any communications protocols, or any human interaction kind of stuff. He just has to worry about whether the information is releasable; goes to the high pot, picks it up, is it releasable? Yes. Put it into the low pot. So by modifying the hardware architecture, we made the proof ages easier.

Slayton: Interesting. That's fascinating. Presumably, your previous work on program proving [pause]

Berson: On KSOS informed this; oh, yes. Absolutely.

Slayton: Had everybody who started Sytek been working on KSOS?

Berson: Let me think about that. Biba, me, and Pliner, yes. Sammy and Jack were working on a different program.

Slayton: Ok.

Berson: They were kind of in our friendship circle. So that's number two; okay; and I think we actually published a paper about that called "*Processor Per Domain Guard Architecture*." It may be in IEEE Security and Privacy.

Slayton: I think I did see that; yes, I did see that. I thought that was paper number one.

Berson: The first was the end-to-end encryption system at the session level.

Slayton: Okay, right, got it.

Berson: And the other thing we did is we built this challenge response authenticator. Years earlier, in a conversation with Dan Edwards, we got to talking about the trouble with passwords.

[Laughter.]

I mean, this was back in KSOS days, and I'd been thinking about that for a long time. What you'd really like is to have one-time passwords, a big list of passwords, you use them once. That way, anybody who hears you over the wire can't log in as you later because both ends knock it off. The trouble is generating, maintaining, storing such lists and keeping them in synch is tedious and fraught with errors. So one obvious way to get around that is to generate the next password on the list on demand, and you can do that with a block cipher. I decided to instantiate that; you could just about program DES in a four-function calculator chip, and I did. We built and shipped this thing — there's one right there — called the Sytek Passport. The thing was this; you turn it on and you press the red button, which puts you in a special mode, authentication mode, and you put in a PIN. So it's something you have, the keyed calculator, and something you know, which is the PIN. Then you go to try to log in, system says user name? You say Rebecca. System comes back to you with a seven-digit challenge. You put the seven digit challenge — why seven digits? — because we all; at the time, telephone numbers had seven digits and people could remember that. You put seven digits in the calculator, press another button, and up comes a seven-digit response; you type the seven-digit response back in to the system and you're good to go.

Slayton: These are common nowadays.

Berson: Nowadays they're very common. I think this was the first one actually built. In the market at the same time as we were, was a guy called Ken Weiss, who used a similar idea except he didn't use a challenge response, he used a clock running the thing. He eventually sold that to RSA, and now that's the RSA Secure ID. Sytek didn't find a market for this; we were selling local area networks, not authenticators…this is 1983. So we sold this to Racal, which sold it for a while under the name of Racal WatchWord ™.

Slayton: Interesting. And Racal? I haven't run across it.

Berson: R-A-C-A-L.

Slayton: Are they still around?

Berson: I don't know, but they were a big British military electronics company. They had Racal-Milgo was an early modem company; Racal-Comsec; maybe Comsec. R-A-C-A-L.

Slayton: Interesting.

Berson: So we sold it to them. It's a slick idea; it was wa-a-ay ahead of its time. The UI wasn't good but it's very, very high tech. DES wasn't proven yet, in 1983; people weren't content that DES was secure. So in addition to DES, I put 15 other algorithms in here as well, and when you enter the key, which you did before you distributed it to users, you choose which of the algorithms you want to use.

Slayton: Right. You originally did this just because you were interested in doing it, it wasn't because somebody had asked you for it?

Berson: That's right.

Slayton: That's interesting. And did you actually try to market it to any company [interrupted]

Berson: Sytek did try to sell it, but the main business of the company was local area networks. The sales guys were selling local area networks. They were the wrong sales guys; they were calling on the wrong people.

Slayton: So your sales guys may not have helped to actually sell security at that point.

Berson: That's right. So that's the three main security things, I think I did at Sytek; The crypto system, the guard — now they call them cross domain solutions, those things are generally called cross domain solutions — and the challenge response authenticator.

Slayton: So what was the culture like at Sytek? You started out with just five people; did you grow really quickly?

Berson: Yeah, we had 500 people by the time I left in 1986.

Slayton: That’s fast. Early on, I imagine, everybody knew what everybody else was doing.

Berson: That’s right; and eventually not. I mean, Sytek, not my department at all, but Sytek, got the contract from IBM PC division to build the IBM PC LAN. So we had this thing we were selling, it was about the size of a desk dictionary; and IBM came to us and said could you put this on a PC card? We said yes, but it’s going to have more processing power on it than the PC itself. They said that’s okay, we’re interested. We did that; we solved a lot of mechanical problems, heat problems, all kind of problems to get that card built. We built them; we built a factory in Mountain View; we went to very expensive school, I mean it was expensive for us, with IBM, about quality manufacturing; and eventually got to the point where IBM had confidence in us to build these things and to put them into IBM boxes in our factory in Mountain View, and ship them into the IBM supply chain. That came crashing down when Philip Estridge, known as Don, who was the head of the IBM PC division, tragically died in a plane crash. The concept of local area networking was heavily competed in IBM. People in Raleigh in particular, I think, were sponsoring a technology called token ring and they said if it’s going to be an IBM product, it has to have token ring. And the IBM business went from a huge part of Sytek’s revenue to zero in the space of a quarter.

Slayton: Wow, that’s kind of scary. What year was that?

Berson: I don’t recall. Anyhow, I had kind of had it with Sytek. We were doing good work but it was very frustrating for me, partly because I was a founder and I felt responsible for everything that happened, and there was just too much happening to be responsible for. So I decided that for my 40th birthday I would retire from Sytek; leave Sytek, and carry it on my mental balance sheet as zero. And I did that.

Slayton: Nice. So that was the point at which you started Anagram, 1986?

Berson: Anagram, yes; 1986, right.

Slayton: Let me back up just a minute and ask you a little more about a couple things at Sytek.

Berson: Okay.

Slayton: Teresa Lunt, you knew her?

Berson: I hired her. And we also hired Bill Wilson, who Teresa had worked for at MITRE.

Slayton: And she started working on intrusion detection?

Berson: Right. We were interested in intrusion detection and I think it was Jim Anderson who was mostly interested, on the customer side.

Slayton: So he was one of your customers already?

Berson: That's right. There were two ways to do intrusion detection. One was to spot anomalous behavior, and the other was to spot behavior that correlated highly with known intrusion behavior. And there still are two ways. And Teresa was working on them.

Slayton: Did you get closely involved in all that or did you kind of keep your distance from it?

Berson: Neither. I mean, it was one of the projects that I managed, but I wasn't actually working on it. I had visited that whole space in my Ph.D. I mean, intrusion detection is an application of pattern recognition and machine learning.

Slayton: Right, it's statistics.

Berson: And I was kind of through with that.

Slayton: Do you remember some of what your threat model was at that point? What were you thinking of in terms of; one of the things that she mentioned in her interview was that she was frustrated at SRI and, I presume, also at Sytek, that she didn't have a lot of examples of sort of real world threats with which to test these systems.

Berson: No, I don't recall much about the threat…

Slayton: Didn't think about it.

Berson: …model. I think we came to it in the abstract. Certainly we were interested in insiders.

Slayton: Insiders, primarily; more interested in insiders than outsiders.

Berson: I think we always have to be interested in insiders because there are insiders.

Slayton: Right.

Berson: And also, the goal of any outsider is to become an insider. And so if you if you are blind to insiders, and can't catch people at the perimeter—then you can't catch them all, right? You have to be looking for them once they penetrate the perimeters and they're behaving as insiders.

Slayton: This may be jumping a little ahead but I'm just curious; at what point — and maybe it was in the very beginning — at what point were you really thinking about the threat of being sort of an international one, as opposed to; I mean, did you think about motive? Insider is one way to describe a threat, but then there's also sort of the question of what is this entity after? Were you thinking about [interrupted] Cold War context? Were you thinking about espionage?

Berson: To start with, we were thinking about espionage.

Slayton: In the international sense, not just like corporate espionage?

Berson: We were thinking about international nation-state espionage. The Cold War ended and the Berlin Wall came down in 1986-ish?

Slayton: It might've been 1989 or 1990.

Berson: Anyhow, we were; Teresa, Bill, the five of us who founded Sytek, and everybody interested in security there; and all of our customers who were interested in security, were all veterans of the Cold War. I mean, we talked about KremVAX. It was a mythical machine, on the usenet; and we had bang notation where you route e-mail around, you had to do explicit routing at the source. You familiar with that?

Slayton: No.

Berson: So if I wanted to do it, I'd have to say first go to here, then go to there, then go to there.

Slayton: As opposed to being decided at the node.

Berson: That's right. There were no routers, no routing tables, you had to know a path from you to your destination. You'd write that out in the address. And so, I mean, we always talked about KremVAX, and routing stuff through KremVAX. Suppose the traffic is routed through KremVAX, which would be our mythical VAX in the Kremlin. There were many names like this, there was MenloVAX, there was SytekVAX; and we also talked about the mythical company, Red Star Software. Red Star Software is going to give you a Lotus 1-2-3, which became, you know, which was the first spreadsheet program and which was sold for money. Now everybody uses Excel, right? But Red Star software is going to have a spreadsheet program, they're gonna have it for free. In fact, I used to have a poster somewhere that I had made up — I haven't seen it in years — it said Red Star software, free good.

[Laughter.]

So, yes, we were nation-state oriented; very much.

Slayton: Certainly thinking much more about that threat than; well, you probably also though about computer crime, as well, by domestic criminals.

Berson: For commercial customers; the internet didn’t get criminalized yet. There wasn’t a lot of stuff of financial value on the internet; it wasn’t until money came on the internet, I think that criminals became interested.

Slayton: So one other question relating to Teresa; she mentioned that you got interested; you worked with her on expert systems.

Berson: Yes.

Slayton: She said that you had gotten interested in studying expert systems from a security point of view because there was somebody living close to you who had some startup involving expert systems?

Berson: Yes, that was a company called, I think, Teknowledge. T-E-K-N-O-W-L-E-D-G-E, or something like that. There was a guy called Lee Hecht; he had a company and a language to make expert systems. I decided I would learn something about it and think about the security implications of that. And I think Teresa and I have a paper about that.

Slayton: “*Multi-Level Security in Knowledge-Based Systems*”?

Berson: Yes, that would be it.

Slayton: In 1987 IEEE Symposium on Security and Privacy.

Berson: Yes. We could talk about the Symposium on security, I hope we'll talk about it more, and IEEE Technical Committee on Security.

Slayton: Yes, that's a few questions down.

Berson: Okay.

Slayton: I was just going to ask, when you were thinking about…so this neighbor got you interested in studying expert systems from a security point of view, were you thinking of using expert systems for intrusion detection at that time?

Berson: It was just something new and I was interested in exploring the space, but it never, for me, amounted to much.

Slayton: Another Sytek project I'm interested in; so Lawrence Halme — you call him Larry — and John van Horn; so I know Dick did some work on automated analysis of computer system audit trails. I can't find any other information about either of them. I just wondered if you could give me any context about who they were. And you mentioned that Halme went on to Arca Systems.

Berson: Right. That sounds a lot like intrusion detection to me.

Slayton: It was, yes.

Berson: I think John lives in Palo Alto, I sometimes see him walking around the streets.

Slayton: Oh really?

Berson: I don't know where Larry is, but Bill Wilson will; or Ken Bauer will. I don't remember that, sorry.

Slayton: No, no, that's okay. Just thought I'd ask. I think these questions you already answered, so that's great. Back to what I was going to ask about Sytek. In 1986, you started Anagram Laboratories. Do you need a break?

Berson: I'm okay.

Slayton: We'll keep going then.

Berson: It's pretty intense.

Slayton: I know. I'm concerned whether we'll scale. We'll try to get through most of the questions. If I need to; if you need to go, I can come back another time, I hope.
So you started Anagram in 1986, you said you were just tired of the situation at Sytek.

Berson: Yes.

Slayton: Was Anagram one of the first startup computer security companies, since Sytek wasn't explicitly focused on security?

Berson: I don't know. To be honest, I wanted to go into the consulting business, and I know from my life experience to that point that some people would rather do business with a business, than business with an individual, so I made a business.

Slayton: Got it.

Berson: Excuse me for eating on tape.

Slayton: No worries.

Berson: It was always my intention that I'd have no employees. It's the ideal number of employees to have.

Slayton: You didn't like management much, is that part of the issue?

Berson: Right. I think the people who worked for me, for the most part, will say that I was a good manager. I mean, they found it useful, and I found it useful, but I didn't find it fulfilling.

Slayton: Fair enough. So who were your main clients or customers at Anagram? Private sector, universities, government [pause]

Berson: It would be industry and government.

Slayton: And so there were companies at that time that were interested in security?

Berson: I'm trying to think of who my earliest clients were. One of them, for example — also, I don't speak much about my clients — I mean, one of the reasons people hire cryptographers is they have secrets to keep.

Slayton: Understood. I'm not asking for names, just [pause]

Berson: No, but I'm trying to give you an example. I'll give you a couple of examples. Pitney Bowes has postage meters, and they wanted to get out of the business of people taking their postage meter to the post office to have more postage put in. They had this proposal for postage by phone. The postage meter calls home and gets postage put in, and they had a crypto system around that. I helped them refine that crypto system and make it correct. They also had an issue with traceability; so when you print postage on an envelope, how can the post office be sure that it's not just a photocopy of some other postage that you printed on the envelope?

Slayton: It's kind of authentication.

Berson: That's right. There are many different techniques used for that, physical and logical. And I helped Pitney Bowes refine and develop one of the logical techniques that's used in modern postage systems.

Slayton: Interesting. So that situation where; part of why I'm asking is because my impression is that security has traditionally been kind of a hard sell. But you didn't find that to be the case.

Berson: No, I've never been out selling. I wait for the phone to ring and it never stops. It never stops.

Slayton: That's interesting. Starting in the 1980s.

Berson: A company making pulse oximeters, you know what that is? It's when you go in the hospital—and I hope it doesn't happen to you—they stick a thing on your finger and it says what the PO_2, how much oxygen is dissolved in your blood.

Slayton: Okay.

Berson: Less is bad, alright? And they do this by shining a red light through your finger and seeing how much comes through. So they make the instruments and they sell the sensors because they get contaminated and are supposed to be thrown away, and you're supposed to have new sensors for new patients. One of the ways these companies make business is the razor blade model; you give away the razors and sell the razor blades. So there were competitors selling cheap razor blades, cheap sensors, and the problem with these is that red LEDs come in different colors and different intensities so there's a whole bunch of calibration curves in the instrument. The genuine sensor tells the instrument about the parameters of that LED and the instrument adjusts its readings. But the counterfeit ones don't accurately tell the instrument…so it had bad patient outcomes, not to mention bad business outcomes. So for them, we developed, at Anagram, developed a system, which actually proves the authenticity of the sensor as being manufactured by the manufacturer of the instrument, and the authenticity of the parameters that the sensor reports to the instrument. And the instrument may, if you plug in a counterfeit sensor, say 'uncalibrated' on it, which of course, reduces patient confidence in the care they're getting. So it's an example of; it's a security solution.

Another one was…I came up with an early scheme for prepaid cell phones. Another early customer was the cell phone industry itself, which wanted to move from old credentials to the new credentials because of tremendous fraud, people were cloning phones. It used to be like if a phone had the equivalent of a user name and password, and they would stay the same for the life of the phone so people would just clone them and run up a lot of bills against the poor person who had been cloned. And the situation was such that criminals would hang outside of airports, which is where a lot of people turn their cell phones on, and capture the registration messages and clone them right there. Another situation, very clever criminals would build a package that would do this and they would ship it cross country, and it would just go across the country picking up things as it went along, and then in the end, it would be delivered to the criminal associate. I mean, you would have to address it to yourself and like if you were in New York, you'd have to address it yourself or an associate in Los Angeles, and you'll eventually come out of the system with a whole…like a trawling net. So those were examples.

Slayton: That's really interesting. There were so many companies that were interested in seeking out help with this, so that there was actually quite an early demand for it.

Berson: Yes.

Slayton: So now I want to switch gears and ask a little bit about professional groups, and a little bit about ACM because ACM doesn't necessarily come up a lot, quite often, in the world of computer security. But I know that you were a founding member of ACM's Special Interest Group on Security Audit and Control.

Berson: But that never went anywhere. So I have nothing to say about it.

Slayton: Oh, okay.

Berson: They said oh, we're going to have a SIG so I joined it and then nothing ever happened.

Slayton: That answers a couple of my questions. Did any of the other ACM special interest groups, like SIGSOFT, or SIGOPS, did you get the sense that they were playing an important role in the field of computer security?

Berson: Well, SIGOPS — whoever runs SOSP —

Slayton: Yes, that's SIGOPS.

Berson: Right. SIGOPS, certainly, because of the operating system thing and I often went to SOSP meetings, especially when they were at Asilomar because that was convenient for me. Operating system people are more concerned about computer security, but I never played a professional role there, unlike two other organizations where I played a huge professional role; which is the IEEE S&P, and IACR.

Slayton: Right. So do you want to say a little bit more about how you got involved in each of those? I couldn't find, by the way, I looked for IACR…when were they founded?

Berson: 1982.

Slayton: Okay, so you were involved from the beginning.

Berson: It was actually some guys — not me — had a workshop on cryptography in Santa Barbara. I could be getting this slightly wrong, but not the place; but the year I could get wrong. And then they did it again; they made a small amount of money; they wanted to have an organization to hold onto the money from year to year. They founded IACR for that purpose. I'm not one of the charter founders but I've been involved with it since the very beginning. I was the general chair of that meeting in 1984; I was the secretary treasurer, which were then combined, of the organization, back then; I've been the president of that organization; I'm still a director of that organization. We now have, I want to say, a thousand members; something on the order of a thousand members from 40, or 60, or 70 — something like that — countries. And it's very active; extremely active; has three flagship conferences a year; Crypto, Eurocrypt, Asiacrypt; sponsors workshops; TCC, which is Theory of Cryptography Conference; FSE, Fast Software Encryption; PKC, Public Key Cryptography, and CHES, Computer Hardware and Embedded Systems. IACR publishes the *Journal of Cryptology*, *Transactions on Symmetric Cryptology*, a cryptographic eprint service, and a newsletter. and that's IACR.

Slayton: Has fellows.

Berson: IACR has fellows.

Slayton: You being one of the inaugural fellows.

Berson: I'm ***THE*** inaugural fellow.

Slayton: Oh really? I thought there were six that year.

Berson: Yes, but they did it alphabetically by last name, so I came first.

[LAUGHTER.]

Slayton: I'm an "S" and I don't think that's fair.

Berson: I agree.

[Laughs.]

Slayton: Fair enough; AN inaugural fellow. Were you at that very first 1982 workshop?

Berson: I was.

Slayton: And how did you know the founders; just professionally? How did you get hooked up with them?

Berson: Well, it was Dorothy Denning and; I'm pretty sure it's Dorothy Denning, and Whitfield, and Marty, and Steve Kent. I mean, they were my colleagues from all the work we had been doing over the years on secure operating systems.

Slayton: That's another really interesting thing, there was this connection between secure operating systems and cryptography, which; I can see how there might be a connection, but the whole formal proof [pause]

Berson: Well, there are people interested in security and then there are people who are interested in operating systems, people who are interested in communications, people interested in program proving; like I always through the program proving people were interested in security only because they could get money for program proving from the security customers. Frankly, I was only interested in program proving because it had an application to security. So the people who are interested in security, it doesn't surprise me, show up in both the operating systems venues and the crypto venues.

Slayton: That's interesting. And there were a fair number of those, then; I mean, certainly Whit and Marty, and Dorothy. That's interesting. Great. Thanks. Just a couple of more questions.

Berson: There are others, like David Chaum, who were IACR founders. I have a list somewhere.

Slayton: Okay, then we can follow up with that later, if you want.

Berson: Okay.

Slayton: So you were also very involved with the SSP?

Berson: Right. The IEEE Symposium on Security and Privacy, also known as Oakland.

Slayton: So you were there at its founding in 1980?

Berson: I don't know if I went to the first [pause]

Slayton: I know you presented a paper in 1983.

Berson: Oh, by then, yes. I put a lot of papers there. I never much liked writing papers; I usually got other guys to write them because I was never on an academic track. I prefer to build systems than write papers, although I think writing papers is important to document what you did.

Slayton: You've got a lot of papers, though; that's one of the things that makes this interview, actually, challenging but also interesting is because you've covered a lot of territory, in terms of both doing it; academically publishing your work, and, you know, industry application.

Berson: Absolutely unpublishable. [Laughs.]

Slayton: [Laughs.] I guess it's unpublishable if it's proprietary or classified.

Berson: This paper fills a much-needed gap in the literature.

[Laughter.]

Slayton: Been there. So there's SSP.

Berson: Right. So I began to go to that, and then; and I played several roles with that. For one thing, I became program chair. The way that conference works is first you serve as the junior program chair, then you serve as the senior program chair, so there's some continuity; I was probably the program chair for two years. I worked with Teresa; I think she came just after me but I don't recall who came before, maybe Dick Kemmerer. Then I became chair of the Technical Committee, which sponsors the workship, and that's a two-year co-chair and then two-year chair thing, so that's a four-year total commitment.

Slayton: That's a big commitment.

Berson: And that got me to see closely into the IEEE Computer Society, and I wasn't thrilled by what I saw there; to the point that when I finished being the chair — after four years of co-chair and chair and what— I stopped going. And in fact, I stopped my membership in the IEEE Computer Society.

Slayton: So what didn't you like?

Berson: I never liked it while I was involved with it because of the tax that IEEE put on the activity. We were always profitable and most of our profits were sucked up to the headquarters, and who provided us no service and no support and only, to my mind, interference.

Slayton: That’s interesting. What are some examples of interference?

Berson: For example, they wanted to be the people who would negotiate the details of the hotel contract.

Slayton: I see. And not always very helpful in the way they did that?

Berson: That’s right. So that’s the third way; another thing I did in S&P was we were meeting in the Claremont Hotel in Oakland. And the Claremont went through — have you been in the Claremont?

Slayton: No.

Berson: It’s a marvelous all-white, wooden confection in the hills, just on the Berkeley-Oakland border, and when we started going there is was quite run down and very cheap. They underwent a change of management, and remodeling, and prices were going to go up. It looked like we’d be unable to keep the meeting there and it was sort of our home, so I took on — because I’d learned a lot about business at Sytek a (from Jack Goldsmith) and by now I had a suit, too — I took on the business of negotiating the deal with the hotel. I kept us there far beyond what made sense for the Claremont.

Slayton: And IEEE let you do that, though, initially?

Berson: Oh yeah.

Slayton: Okay. But you said they wanted to [pause]

Berson: We, having done all the work; they wanted to charge their tax for, amongst other things, negotiating the hotel.

Slayton: Oh, I see. So they didn't insist on doing it but they charged you a tax [interrupted]

Berson: Right, then they would call up and say well, what we really meant to say was this, that, or the other thing. I forget the details. I was un-thrilled by them; I was un-thrilled by their professional staff; and I was un-thrilled by people who made their career out of being professional IEEE volunteers. Now I'm really going to get in trouble.

Slayton: You can edit it, if you want; but yeah.

Berson: It was just not me.

Slayton: Got it. What was the culture like, there?

Berson: It was great in the S&P. Oh, it was great; it was a party every year; it was the same people you see. We had a 30-year event; and the picture is someplace, yeah.

Slayton: I'm not sure I saw the picture; I saw there was a publication on how S&P had changed over the years.

Berson: Right. In that same *Proceedings*, we took a photograph, or we took a photograph the year before. You can find everybody who's on your list; just about everybody who is on your list should be in that photo.

Slayton: Interesting. Okay, I'll look for it. Thanks. And that was a pretty small meeting, right? Relatively speaking.

Berson: I think it started with a couple hundred people, but then it got very large and one year we had to appoint an attendance czar or registration czar who would decide who could come and who couldn't. What happened is that some organizations decided to use the conference as training ground, so everybody they had hired in the past year who had to work in security, they'd send off to Oakland, so you'd find like 30 people registered from one place. Not a bad thing, necessarily, but it was bad because we were capacity-limited at the Claremont. So Dick Kemmerer had to be the registration czar and decide who could come and who couldn't. There were some years where we asked people who wanted to register to write a little essay, just a few lines, about what was their work in information security.

Slayton: Interesting; that's fascinating. How many did you turn away then at the Claremont, 10 percent? Half? Seventy percent?

Berson: I don't know because I wasn't the czar.

Slayton: Okay, right. That's interesting. Now you were probably also active at RSA?

Berson: I attended RSA. I attended, I think, the first one in a basement of the; no, I don't know where the first one was. I remember an early one in the basement of the Sofitel in Redwood Shores. But I was a regular attendee at RSA. Of course, I knew R,S&A; I still know R,S&A; and in return for some business advice early on, I had early shares in that company.

Slayton: Oh that's nice.

Berson: Which was very nice, except my shares were worth maybe 15 cents, is what I paid for them. I sold them when they reached $1.50, which was way too early. I should've held onto them for longer. But I figured 10 to one wasn't a bad return; 10 to one in a few years.

Slayton: Not bad at all. So what was the culture like at the RSA Conference? How would that have compared to Symposium on Security and Privacy?

Berson: RSA was a company with products, and was selling something.

Slayton: Right. I mean the conference. Right, it was a trade show?

Berson: A trade show/user group. It wasn't what it's become now, which is sort of a convention with booths and babes of all possible gender, and bands, and swag, and stuff like that. Early RSA was nerds sitting in a basement talking about technical details. But there were actually some real technical issues at early RSA conferences. There was development of the PKCS, and so a lot of discussions about PKCS series, Public Key Cryptography Standards series. And so there were very detailed technical questions about those things.

Slayton: Right. And so there was a lot of; you found a lot of interesting, important technical work being done, at those conferences.

Berson: At those early conferences. Now, I'm not sure there's much technical work at all done.

Slayton: Probably more of an applied nature than what was going on at SSP?

Berson: SSP was about operating systems and program correctness, and Crypto was still of a very applied nature. Early Cryptos were tremendously engineering-oriented and application-oriented. Later, they've been more oriented toward theoretical cryptography; taken over by the computer scientists, which is not necessarily a bad thing. They've brought a certain rigor and large view to Crypto. But early Crypto conferences were much more engineering-oriented. What you heard in the early Cryptos that you don't hear much now were papers on coding theory, information theory, even more number theory than we have now.

Slayton: So the theory has moved, as well.

Berson: The theory has moved, certainly.

Slayton: That's interesting.

Berson: I think so.

Slayton: Were you also active at the National Computer Security Conferences?

Berson: No. Just one conference too many. I went to one or two, but [pause]

Slayton: Were there any ACM conferences that you attended? I can list of several of them; I made a list. The IEEE ones tend to be more cited. The Computer Security and Privacy Conference?

Berson: No.

Slayton: No, okay. The New Security Paradigms Workshop?

Berson: No.

Slayton: Symposium on Access Control Models and Technologies?

Berson: No.

Slayton: The ACM Conference on Computers and Communication Security, CCS?

Berson: No.

Slayton: And then the Computer Security Applications Workshop?

Berson: No.

Slayton: Would you characterize SIGSAC—because a lot of these were associated with the Special Interest Group on Security, Audit and Control. Were those sort of people who were just doing work that wasn't as good or [pause]?

Berson: It was just different.

Slayton: It was just different.

Berson: I didn't go to the conference; I haven't read the proceedings; I can't say their work isn't as good, it may be spectacular. The reason I didn't go to them is just lack of bandwidth. I'm totally committed to IACR; totally committed to IEEE; and also making a living. When you work for yourself, nobody pays your salary but you.

Slayton: It's a lot of pressure, for sure.

Berson: Right. So you have to [pause]

Slayton: It's just really interesting because I've looked at SIGSAC a number of times, thinking why isn't there more going on; like all the people that we're looking at are not in that group; and all the people that I see in that group — with the exception of you — I hear about nowhere else. So, it just makes me wonder what is that microcosm about?

Berson: I'm not a social scientist so I don't know; maybe tribalism? Which I think is a scale-free property of humans. Any layer you look, you can find tribes.

Slayton: [Laughs.] That's nice. Okay, another conference-y thing; a workshop thing. In 1988, you led the Identity Verification Working Group, or Authentication Working Group, at the Workshop on Integrity Policy for Computing Information Systems.

Berson: I did?

Slayton: According to a publication in the SIG SAC Review, yes. Clark Weissman was on that panel; it was Peter Kapec and Jim Schweitzer.

Berson: Where was it?

Slayton: Gee, where was it? I could go back and find [interrupted]

Berson: What's it called again?

Slayton: The Workshop on Integrity Policy for Computing Information Systems; so what I have found is that it was a workshop that was apparently in response to a paper by David Clark and David Wilson . . .

Berson: Oh, okay.

Slayton: . . . basically saying that formal security models are giving too much attention to confidentiality, so Bell-LaPadula, and not enough attention to integrity. You know, the confidentiality reflecting Defense Department interest, but not thinking enough about commercial interests.

Berson: I remember that paper.

Slayton: And so I was just wondering if you could say anything more about the context of it; if you know who sponsored that workshop; how you got involved in it; those sorts of questions.

Berson: As you can tell, it's gone out from my mind.

Slayton: It's okay if you don't remember, we'll just move on.

Berson: I mean, I still collaborate with David Clark.

Slayton: You know him from your work in networking, originally? Where did you first meet him?

Berson: I think at Oakland. I would recommend to anybody, to everybody, to go; to choose a conference in their field and to go there regularly because you get a network of — I don't want to say friends, necessarily — but of collaborators, some of whom may become friends and you will work together throughout your lifetime.

Slayton: I think we're almost done.

Berson: Okay, you're doing well. Do you need a break?

Slayton: No, I'm fine. I just don't want to; I know you're very busy and I don't want to take too much of your time. So when you were nominated, or awarded the IACR Fellow Award, the citation was for "visionary and essential service to the IACR and for numerous valuable contributions to the technical, social, and commercial development of cryptology and security." So I'm curious to know what you would view as your most contribution; what do you think mattered most?

Berson: What I think mattered the most to IACR?

Slayton: To the field.

Berson: To society, in general?

Slayton: To the field of computer science and security.

Berson: I think for society in general, we — and I mean a very large we — won the Cold War. And I'm not taking credit for that . . .

Slayton: Yes, I hear you.

Berson: . . . but that was what we went to work for, every day, for a period of time.

Slayton: And you felt that drive more than, say, keeping the U.S. government out of peoples' business. Some people really saw cryptography; so Marty Hellman got on the keep-the-powers-of-the-government-limited kind of thing.

Berson: No, that's not my thing at all. I, you know; so my most important contribution. I've liked a lot of them. I think trying to prove an operating system is spectacular; I think building crypto systems; I mean, nothing pleases me more than to see my cryptography in use by people who don't even know they're using cryptography. As an individual consultant I can't do that; so I try to choose clients who dominate, or are likely to dominate their markets. So for instance, Pitney Bowes, who I mentioned, every time I get a letter with one of those meter stamped postage indicia on it; I'm kind of thrilled a little bit.

Slayton: I could see that.

Berson: Skype. I reviewed the Skype crypto system and helped them perfect it, and it pleases me that everybody who uses Skype is using a crypto system I had a hand in. It's a huge impact on the world. Is it intellectually hard? No. Neither of those things were intellectually hard, just good engineering, but yet engineering into products that make it into peoples' hands and are used. Let's see; I think a huge impact that I made was shepherding IACR from its early stages. When I took over IACR, it was running on a shoestring. During my run there we turned it into a substantial organization that's not going to go away. It's hard to imagine; when we started IACR, there was no literature in cryptography, no open literature. We created a bookshelf that, had it all been printed out, would probably run 20 feet long. Now, of course, most of this stuff is just PDFs and you can't see how much it fills up bookshelves. We created a substantial amount of literature. The field is thriving. A quarter of the people that register for conferences register at the student rate and so if even most of them are students . . .

Slayton: That's pretty big.

Berson: . . . right? It's great, because there was a time when the average age of the attendees was going up one year for every year; for every year.

Slayton: It's a little frightening.

Berson: It was a little frightening. But now it seems that crisis is past. There may be other crises coming, but that crisis is past, so it has life and it has legs. So for me, that's very fulfilling a thing to have done.

Slayton: Yes, fantastic. So then another question — almost an ultimate question — it seems like your work is somewhat unusual in that you've spanned both theory and practice and I wonder if you've seen sort of a big gap in the field, between the theoretical work that you've done and the more practical work? Is there sort of a gap in the field and how do you navigate with that divide?

Berson: For years, I thought about my work as somebody who could understand the theory and reduce some of it to practice, alright? I don't think I create new theoretical breakthroughs but I like to be there when they're being created, and to understand them early, and to visualize, and if possible, implement practical applications of them.

Slayton: Got it.

Berson: Lots of theory these days, it seems to me that the theory has increasingly diverged from what's practical. I think the gap has grown over the years, as theoreticians build upon earlier theories, although there are some spectacular crossovers. Phil Rogaway and Mihir Bellare in the crypto world, and some of their students have taken theoretical insights and reduced them to practice, which I admire very much.

Slayton: That's good. So this last one is a question that Lynn Eden said you really have to ask; if you could tell a social scientist or historian what key question or problem they should be working on, related to computer security, what would you tell them?

Berson: These days, computing systems are so deeply integrated into peoples' lives that they're often closer physically and emotionally, even, to their computers than they are to their family members, colleagues, neighbors, fellow citizens. And yet, a computer doesn't feel responsible, doesn't have a personality, has no ethics, it's not a person. It's less loyal; it's certainly less loyal than dog, right? And less loyal even than a cat.

Slayton: That’s saying something.

Berson: I mean, a computer just doesn’t care. I think — I never thought about this before — it’s a great question; brava to Lynn.

Slayton: She always has great questions.

Berson: Brava to Lynn. So I think from a social science point of view, it might be interesting to study the relationship between people and these computers, which are being treated like people; being treated as though they deserve a seat at the table.

Slayton: Yes. Sherry Turkle has done some work in that area. Do you know Sherry Turkle?

Berson: Is she in Georgia?

Slayton: She’s at M.I.T.

Berson: She’s at M.I.T. What’s the name of the woman in Georgia? Because Sherry Turkle came to PARC; I worked at PARC for a short period of time.

Slayton: Oh, right. Lucy Suchman was another person at PARC, although I’m not sure if she’s there anymore.

Berson: She wasn't there when I was there. Right. But anyhow, tell me about Sherry's work.

Slayton: She's recently started, I think, with robotics, but she's generally looked at how people and social groups form relationships with computers. So she did some early work on hackers at M.I.T., she wrote a book called *The Second Self*, looking at virtual communities; and now she started looking at robotics and she has a great TED talk, where she's talking about watching older people interact with robotics, because they're being pushed into the direction of well, these could be good therapies [interrupted]

Berson: For elder care. Absolutely, right?

Slayton: She was sort of saying that isn't it depressing that we're kind of pushing robots onto our elders so we can go do something else.

Berson: Instead of caring for our elders ourselves.

Slayton: Or learning from them. They have so much experience and wisdom that we could benefit from and instead we say oh, take this stuffed seal. So anyway, I'll send you a link to some of her work, you might find it interesting.

Berson: Please do.

Slayton: She's great.

Berson: But anyhow, I think that's one social question. Another social question, of course, is for the value of about a dime, we give up a lot of information about ourselves. And I'm talking about advertising networks now, not talking about government surveillance. I'm talking about advertising and I suppose it's a fair trade; I mean, if people didn't think it was a fair trade would they continue to do it? I mean, is the cost of the trade palpable? People certainly see some of what they get out of it, but do they see; it is salient; is what they're giving away salient to them? I think the answer's probably no.

Slayton: That's interesting. So the question of why people; or maybe, how can people be made to appreciate what they're giving away, because I don't think people really get it.

Berson: You know Walter Mischel marshmallow experiment at Stanford, where you give the kid a marshmallow. Take these four, five, six-year-olds and you say okay, here's a marshmallow, put it on a plate in front of you, and you can have it now, but if it's still here when I come back, I'm going to give you two marshmallows.

Slayton: I heard about this with the Oreo; it was an Oreo in my version, but yes.

Berson: It has to do with being able to defer gratification for a bigger payoff later.

Slayton: Right.

Berson: Smashing, smashing experiment; and it turns out that the ability to defer gratification is highly correlated with positive outcomes later in life; or the inability to do it is correlated with negative outcomes later in life. Of course, the experiment has been designed by someone who values the ability to defer gratification, but never mind that experimental bias. What will be; what is the experiment of similar sort of leverage that can be done with respect to people and their interactions with computers?

Slayton: That's interesting. I should talk to Jonathan Mayer about that.

Berson: I don't think we've seen that yet. What experiment about computers will speak and inspire generations of social scientists, and will illuminate human/computer interaction is the wrong word because HCI is just sort of at the level of the user interface.

Slayton: Yes, it's GUI.

Berson: But at the Sherry Turkle kind of thing, you know.

Slayton: Relationships to computers, something like that; more than just interactions.

Berson: Right. Or society; what do you call a society that includes computers? I mean, one of these robots?

Slayton: You know what we call people, we call them cyborgs. A cyborg society; I don't know. That's a good question.

Berson: Yes. Okay.

Slayton: And the last question is always are there any other questions I should've asked?

Berson: I'm sure there are. [Laughs.]

Slayton: Okay. Well, if you think of them; if you want me to ask them now, or do you want to talk about them now? Or we can wait.

Berson: No, no; because it's a close-ended question. The last question should be, what other question should I have asked?

Slayton: Oh, that's what I meant. Thank you, I need to reword it. You got me. I meant to word it that way.

Berson: [Hearty laugh.]

Slayton: Okay, so now that you've said yes, the real final question is what other questions should I have asked?

Berson: I don't know.

[Laughter.]

Slayton: Fair enough. Well, you're local, so . . .

Berson: Yes, you can come back.

Slayton: That sounds good. Thank you so much for your time. This has been amazing.

Berson: You're welcome. It's rather flattering to be the focus of such intense curiosity.

Slayton: It's so fun talking about this stuff; I learn so much every time.