An Interview with

RICHARD Y. KAIN

OH 459

Conducted by Thomas J. Misa

on

27 May 2015

Computer Security History Project

Minneapolis, Minnesota

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

Richard Y. Kain Interview

27 May 2015

Oral History 459

Abstract

Richard Y. Kain discusses his graduate education in computing and his work as a professor at MIT, including consulting with Bolt Beranek and Newman (BBN). The interview focuses on his consulting work with Earl Boebert at Honeywell on several iterations of the Provably Secure Operating System including the spin-off Secure Computing Corporation. It also describes his career as an electrical engineering professor at the University of Minnesota and interactions with the local computer industry.

This material is based upon work supported by the National Science Foundation under Grant No. 1116862, “Building an Infrastructure for Computer Security History.”

Misa: My name is Tom Misa and I'm here with Richard Kain in his house in Minneapolis on the 27th of May 2015. We're going to do an oral history looking at Dick's technical career, with a special focus for our NSF-funded project on the history of computer security. Just to begin with, Dick, could you outline your background and education, and how you became interested in computing before you went to MIT?

Kain: As a child, I was always interested in science, what they now call STEM types of things, and especially the science and math parts of that. I was a very interested person — interested in astronomical things. When I went off to Andover in 1950, for my birthday in January of 1950, my parents gave me a subscription to *Scientific American* magazine. That October — I think it was October, it was either October or November of 1950 — there was an article on the cover of *Scientific American* about Simple Simon, which was a very simple relay computer that had been built at Columbia because computers were too big to let students have access to them. They were too big and too expensive so they built this little toy, which sat in a box on the bench, which was just a relay machine, trying to show basic principles of computing and logic, design using relay circuits. So this was written up; it was called Simple Simon, and it was written up by Edmund C. Berkeley, who wrote a book called *Giant Brains*, which I own and still have. So I got interested in that and decided I really want to do that, so astronomy dropped out of the picture at that point. I still read astronomical things and look at astronomical pictures quite frequently, but astronomy dropped out of the picture then and computing, logical design and stuff got my interest.

Misa: Did you actually build something that was like Simple Simon or were you just playing around with some of the ideas about how to take relays and make them into logic circuits, which is what Berkeley was intent on doing?

Kain: Actually, what I spent a lot of time doing on scrap paper was trying to build a relay circuit to play tic tac toe. I think I sort of figured that out but I don't remember enough details about that anymore.

Misa: So the tic tac toe would actually involve memory as well as some kind of logical thinking.

Kain: No, I don't think I had any memory in there; I think I was just doing the logic of stuff.

Misa: About when? How old would you have been in 1950?

Kain: Fourteen.

Misa: Fourteen. So this is prior to your going to MIT. Could you tell the story about how you ended up at MIT as opposed to Swarthmore, where your parents had gone?

Kain: My parents had both gone to Swarthmore and that's where they met, got married, and I'm the oldest child. I was sent, basically, to Andover — not my choice — because my mother's good friend had found Andover for her son, who was just an acquaintance of mine, not a really good friend. They lived next door to us when we first moved to Louisville for about a year, then we moved bought a house, and they bought a house, and we were dispersed. But the parents kept up a friendship.

Misa: So you were living in Louisville, Kentucky?

Kain: Louisville, Kentucky, yes. We had moved there when I was about four. My father taught at the University of Louisville, that was his career, in English Literature. So I went to Andover and then when the question of college came up — I was a good student there — my parents were pushing Swarthmore and I was pushing to do something where I could do computing. And I didn't know exactly where that was or how that was going to be. I had a friend there, I think his name was Al Sterns, and he was applying to MIT, and so I thought about it; yeah, let's try that. I also had applied to Swarthmore and my parents were giving me all kinds of pressure about how I should go to Swarthmore. And finally, I looked up all the details and found out that indeed, if I'd gone to Swarthmore versus MIT, Swarthmore would let me get by with fewer humanities courses than MIT would require. So I'd have to get more humanities at MIT than at Swarthmore and so that tipped the scales and they didn't talk about Swarthmore anymore. Actually, my brother went to Haverford, and later, my sister went to Swarthmore.

Misa: A family tradition there.

Kain: So Haverford is the arch rival among the Quaker colleges in the eastern Pennsylvania area. I went to MIT — should I keep going?

Misa: Yes.

Kain: When I arrived at MIT in the fall of 1953, then the question I was asked, as all freshmen would be asked, was what do you want to major in? To which I said I want to study computers, what should I major in? They said either math or electrical engineering. I knew something about math; I didn't know much about electrical engineering because that wasn't in my background, or in my family, or anything like that. So I said okay. I was a good student, and I guess I didn't have to commit right away. I don't know what they wrote down, but anyhow, I was a good student in calculus, and I had a gung-ho young instructor who later became famous. I don't know whether I should mention his name or not, but anyhow, Gus Solomon. He encouraged me to major in math and to take a math course of linear algebra as an elective in spring semester. This course being in the normal course of scheduling for math majors, you would get that in your junior year, but it did not require the calculus that you would get in the first two years. So from a logical sense of what prerequisites were required, it would fit. He also encouraged me to take the more theoretical version of the second semester calculus, which was a real loss for me because that was theorems and proofs, and that's not my shtick. I'm good at solving `problems but I'm not good at theorems and proofs. I mean, I can do the geometry ones but when you get into those, that's too much. So after about, I don't know, three or four weeks, I started thinking about this algebra course and I was wondering why do they even ask these questions and why does anybody care about the answers to these questions? And so therefore, I dropped the course and I became an electrical engineer.

Misa: What kind of courses in EE were you taking at the time? Were there computer courses *per se* at MIT at the time?

Kain: Well this was freshman year. Freshman year all you're taking [are] basic physics, math, chemistry, humanities kind of stuff; and so that's all pretty much a set piece and the only difference among all the majors at MIT in those days, between what you would take freshman year, had to do with whether you were doing something chemically related or not, because then you had to have more emphasis on chemistry. Otherwise, it was all pretty much the same bag.

Misa: Physical sciences plus some version of calculus.

Kain: Yes. Physical science, calculus, and a humanities course — one of these general surveys of eastern, western philosophy and life, and stuff; starting with Greek stuff, which I was no good at. Then it got really bad because my grandfather died during exam week and I had to go away for the funeral, [and] then I had to come and take the later exam.

Misa: Kind of a challenging time, but you moved into EE then?

Kain: Yes, and it was EE from then on.

Misa: Were the courses oriented more to power engineering or were there electronics courses? I don't know whether there were any computer courses at MIT in the 1950s. By 1960, of course, things had changed.

Kain: There were. Let's see, I can't give you precise timing of all this, but I took Sam Caldwell, [who] taught switching theory. I took the switching theory, I think in my senior year. A couple of the lab instructors became friends later on; Fred Hennie and what's his name? I'll get it back later. And so I had this course in switching theory, and then about the time — I think this was senior year — MIT got an IBM 704, which was installed in their new building. It had a new wing, it had glass on three sides, and it had this big computer in it. So I took a course in programming around senior year or first year of graduate school.

Misa: What kind of languages did you use?

Kain: Assembly language.

Misa: Assembly language.

Kain: Assembly language; this was before FORTRAN. This was 1956 sort of time; FORTRAN was 1957.

Misa: FORTRAN was being developed at the time.

Kain: Yes. Well, somewhere around that time. And then later, if you're into that stuff just as curiosity, later on I was supported as a graduate student by being a teaching assistant, and later being promoted to being an instructor, which gave you more responsibility, in terms of leading recitation sessions and big lecture courses, especially. I did that for electronics and circuits, and for computer related topics.

Misa: Let's continue for a moment —

Kain: Wait, let's finish the FORTRAN story. So then later on, as a graduate student, we were teaching MAD. Do you know MAD? MAD was Michigan Algorithmic Decoder. Bernie Galler did this and this was just being developed so all we had was printed out versions of a manual for this. We were teaching the students programming language in this implementation running on the IBM 704, or whatever model it was at that time. We had a problem because we gave the students a project to work on and one of the problems was to program the motion of a double pendulum. You have one pendulum hinged on the weight of the higher pendulum, and so there are two ways of solving this problem. One of the ways is to do an energy calculation, which involves taking a square root, and the people who were doing it that way were getting an error message that it was taking the square root of a negative number, which of course, it shouldn't have been.

Misa: Right.

Kain: And so it turns out the problem was that when Bernie Galler made MAD, they co-opted the FORTRAN subroutines to do the math functions but they didn't get the calling sequences right, so they had made an assumption that had to do with whether the parameter for something simple like taking a square root was in the accumulator or whether it was in a memory location. And they got it wrong, so because there was an intermediate calculation to determine what was under the square root sign, what you're taking a square root of, got messed up and so they ended up with a negative number passed as a parameter. So this program just plain didn't work. We were discovering a bug in the way they'd done all their subroutines and so we then read by hand all these solutions that these students had turned in to see if they were correct because we couldn't run the programs. [Laughter.] So that's the little piece of FORTRAN-related history.

Misa: Okay. Back to your undergrad time at MIT; you did some programming. Did you continue with math or did you go straight into EE then your second year?

Kain: I was just an EE major. I did not take super advanced math courses, I just took the regular required stuff. Later on, as a graduate student, I think I signed up for tensor calculus but I dropped out of that, I think, after a while because I got too confused about all the — I don't know if you know tensor calculus stuff, but basically they're all kinds of implied summation signs that are never written down, because if you have a subscript and a superscript that appear with the same symbol on two different variables, that means you have to sum over that variable over all possible values of it. I got too confused.

Misa: There's something I still wanted to ask you about your undergraduate years. You said you worked with Wes Clark for your thesis. Could you say a little bit about that? Wes became quite famous for LINC and had quite a distinguished and notable career.

Kain: Then he later went to Washington University, yes. Okay, back up. MIT required in those days, and may still do, that anybody granted a bachelor's degree had to do a thesis. So I went searching around [for] what can I do in computing, and somehow, which I don't have any recollection whatsoever, I got hooked up with Lincoln Labs, where they were developing and finishing up working on a computer that was called the TX-0 [zero], which was an experimental machine designed to answer, as I was told, two questions. One question was can you build a computer using these new-fangled transistors, these high speed transistors; can you build a high speed computer? And the second question was gee, over in the corner, we have this massive memory which has 36-bit words because that was the size of the words in IBM scientific machines, which I understand came from the fact that nuclear physicists wanted that kind of precision, so that's where that 36-bit number came from originally, as I understand it. It had 32,000 words of memory, which was about a seven-foot cube, as I remember it, with vacuum tubes on the outside and the core stacks in the middle. So they wanted to check that out so they were building an experimental machine using transistors. They were using surface barrier transistors using a kind of logic which is not interesting anymore. But anyhow. And so my project was to look at the way they had designed this machine and see if they could have simplified some things and saved some logic and so on; do some analysis of the logic design of this machine.

Misa: The TX-0?

Kain: The TX-0, exactly. The technology of this was quite primitive; the logic gates and transistors were stuck in little plastic tubes like little medicine bottles that stuck out of a vertical panel, vertical chassis stuff, many feet long, of stuff. In the course of this, I wrote probably my first computer program that really ran on the computer, which was in assembly language for this machine. So this group of people had developed ways to make transistor logic, which they then, eventually many of them — not Wes Clark — but many of the other people went off and formed a company to market logic boards with logic gates on them, transistorized logic gates on them, and that was Digital Equipment Corporation, which started from there, from this project.

Misa: There's a question whether Digital at that time was aiming more at computers *per se*, or whether they were trying to limit themselves to stick to the logic boards. Do you have any ideas about what they were looking for?

Kain: I know about that. Oh, yes, definitely they were just selling logic boards. Definitely that was their business. In the Cambridge area was Bolt, Beranek and Newman [BBN], and Bolt, Beranek and Newman knew about them and wanted to have a computer, and approached Digital Equipment and said we would like to have a computer. Digital Equipment said we don't have enough money to develop and build a computer because in order to really do it right, you have to build a prototype, debug the prototype, keep the prototype, and then build Serial Number One, and then that's what you would deliver to the customer. Bolt, Beranek and Newman said you don't need to do all that; we'll take the prototype. So Bolt, Beranek and Newman put them into the computer business, in effect, by agreeing to basically support; to help underwrite this prototype, using their logic boards and stuff, to actually build an operational computer. Later on, I think it was after I had gotten my doctor's degree; I had consulting at Bolt, Beranek and Newman, which involved actually programming that machine.

Misa: Oh really. Okay.

Kain: That same machine, in basically a language called Decal. I don't know if you know that one or not. That's another interesting story, but anyhow, I won't digress into that unless you want me to. But that's the same machine where Licklider was developing time-sharing.

Misa: Oh.

Kain: It was on the same kind of time frame. He was down the hall and it was in the same kind of time frame that we were actually working there on developing, trying to support some symbolic mathematics.

Misa: I’ve got that symbolic math paper here, “Computer Aid for Symbolic Mathematics.”

Kain: From the computer conference?

Misa: Well, it says Lewis Clapp and Richard Y. Kain; 1963 is when it was published.

Kain: Yes, that’s it.

Misa: Do you want to say a few words about that? Because the manipulation of math, if I understood the paper right, you weren’t interested solely in evaluating the mathematics and drawing graphs or tables, but you were interested in the manipulations of equations and using a computer to assist with the manipulation of equations directly, symbolically.

Kain: Yes.

Misa: How did that come about?

Kain: Where did the project originate from?

Misa: Yes, or the idea.

Kain: I think the project came from somebody else; maybe from Lew, I don't recall that. But that was our project, yes.

Misa: It seems atypical use of machines, if you're thinking about trying to get computers to do something, this kind of symbolic manipulation is quite notable.

Kain: Thank you. [Laughs.] What else can I say? And, you know, after I came to Minnesota we started working a similar kind of thing, working in the control theory area with Fred Bailey. That was much later, because I came out here in 1966 and started that with Fred Bailey in the electrical engineering department.

Misa: Okay. Let's stick with Cambridge for just a moment because there's a question then about how you ended up going to graduate school at MIT and your graduate work there, and then you also served as an instructor, also as an assistant professor at MIT.

Kain: Yes.

Misa: So can you fill that out?

Kain: Sure. I was a good student and I finished undergraduate and just applied to graduate school. So I got into graduate school there and became a teaching assistant, basically teaching circuits and electronics labs for a couple of years. And then I was teaching recitation sections, and so I did a master's thesis, writing programs again for this

TX-0, which had moved onto the campus by that time. I was doing some pattern recognition studies, which didn't come to much in the long run, but did some pattern recognition programming, and then I went on to get a doctor's degree. When you applied for a doctor's degree in those days, you could get either a doctor of science or a doctor of philosophy. The requirements were the same and I thought I'm not a philosopher, so my degree is ScD, Doctor of Science instead of Doctor of Philosophy, although the requirements were identical. Apparently, historically the requirements had been different, at least with respect to foreign languages. When you had a doctor of science, you only had to have one foreign language; for a Ph.D., you had to have two. But by the time I came around, you had to have two for both of them, so they were identical degrees except for the name.

Misa: Could you describe the colleagues and the working environment? Were you attached to a lab? Sounds like you had worked as an undergrad with this project, the Lincoln Lab. Did that continue as a graduate student?

Kain: No. I was kind of an outlier kind of person, who hooked up with an outlier kind of person, so I was not involved. I was not indoctrinated into the research mode of operation or any of that kind of stuff. I was kind of an independent person with my own ideas and running on my own time, and not getting paid by a project. I was always getting paid by teaching. So I did not get trained in the way to play the game.

Misa: Project MAC was evolving during some of these years. That was a huge project and a source of substantial employment. Do you recall interactions with the project?

Kain: Yes, I was actually paid partly by them when I was an assistant professor, for a couple years. I don't remember the exact timing of all of that, but through that I got connections with other people that have lasted longer.

Misa: That was a very fertile group. I don't know that we mentioned your Ph.D. advisor and your thesis topic.

Kain: No we haven't. Okay, when I was looking around for a Ph.D. thesis I went around talking to people in the computer field and got connected with Dave Huffman. Dave Huffman posed me a question, which I've forgotten exactly the nature of the question, but it had to do with whether there was some different way of doing logic. I think it had to do with whether you could make circuits that had multiple states in various ways.

Misa: Rather than just binary?

Kain: Rather than just binary. Of course at that time, there were machines, Burroughs, I think, had developed — I've forgotten which company — but it developed a vacuum tube that had 10 states, so they were using that for some kind of computing. I don't remember details of that at all; but this question got me thinking about various things so I actually ended up doing something about. I don't know how much description you want here, but if you make a chart, a graph of current versus voltage for a passive circuit, and you make it be a piece of straight line, so straight line segments of any shape, and you make a circuit like that using diodes and resistors, and batteries, and things like that to make those threshold voltages. I kind of solved that problem and reduced the whole thing to a question, which I think is still unanswered, that I could give you the matrix of conductances. And the question is can you build a circuit that has that matrix of conductances, that exact matrix? So I think that synthesis problem is still not really solved.

Misa: That was the basis of your thesis?

Kain: The basis of my thesis was finding all the combinations, and finding computing around some geometric constructions and stuff — which was great fun — about the break points, the places where the line segments meet and make an angle in this curve. And so the whole thing was called, the title was “Diode Network Synthesis.”

Misa: That is quite different from programming . . .

Kain: It’s totally unrelated.

Misa: . . . but to the circuitry necessary to computing.

Kain: It has no application as far as I know, except that it got me a degree. [Laughs.] That's a good application.

Misa: That's obviously something. Can you describe MIT in the early 1960s? You were an assistant professor there. We talked about Project MAC, you were a consultant with BBN. I was interested in the relationship between MIT faculty and the BBN group.

Kain: You know, I don't remember exactly how I got connected with the BBN group. I can't dredge that up at all; I have no recollection of how that connected. It may or may not be a very indirect thing. My first wife worked in education and she eventually got a doctor's degree in education from Harvard. She got a job at BBN, where they were having a project about teaching machines. She was basically writing scripts for teaching machines. I don't know if that had anything to do with me getting a job at BBN or not, but I got basically started out as having a summer job. I think that was the summer after I got my doctor's degree, which was in 1962; so maybe that was in the summer of 1962 I got connected up with BBN and got involved in this project about doing the symbolic mathematics and that continued as a consulting job.

Misa: BBN, of course, started out as being acoustics related, but by the 1960s it seemed to be doing two things. From what I've read, they were still doing acoustics and also a research shop. Some people have called BBN a third Cambridge [area] university. Do you think that's a fair statement?

Kain: I don’t know about calling it a third Cambridge university, because I’m not sure they were educating anybody and I’m not sure . . .

Misa: In the sense of doing high level research that was not connected at all to acoustics and was, in a way, quite forward looking.

Kain: I can’t comment about the forward looking piece of it, and I can’t comment about the diversity of it, but I can comment about a small piece of it because I was just a research subject in one little piece. And this was all connected with — my emphasis was I thought they were doing a lot of stuff that was related to psychology. So as far as I was concerned, they were doing acoustics, psychology, and computing. Those are the only things I’m aware of, really. And the psychology part, they asked for some volunteers to participate in an experiment where the question was whether they could relieve people’s sensation of pain by irritation. And so we would go in and we would get our forearms wired up and they would give us electrical shocks and we would say how strong the shock was, and they would up the ante progressively. They would do it with or without a vibrator vibrating on you to irritate you to see whether that made any difference. So that was one of the things that was going on at the same time there.

Misa: So it’s kind of . . .

Kain: And my first wife’s connection with this teaching machine stuff that was being done. I think one of the people who was there, I think, on the computer side was named Tom Merrill. I don’t know if you know that name or not.

Misa: I’ve heard it, but I don’t know much about him.

Kain: I can’t tell you much about him either, but I think he may have been the manager of the group under which I was working with Lew Clapp and we were doing this symbolic mathematics stuff.

Misa: J.C.R. Licklider was another one of the people that moved back and forth between BBN and MIT; did you have any contact with him?

Kain: Yes, a little bit because he was developing the time-sharing system on the same TX-0 machine, so yes, I ran into him from time to time.

Misa: Did you also work on time-sharing? You were working with the TX-0 doing some programming.

Kain: But it was not time-sharing.

Misa: Not time-sharing.

Kain: Not time-sharing, no. It was just an application running, which I don't remember whether it was time-shared or not, but it doesn't make much difference.

Misa: People look back and see time-sharing as a major turning point. Do you have any sense of what it looked like to you in the early 1960s? Was it an intriguing experiment, or wacky idea, or a clear way of moving computing forward? People look back and have all kinds of visionary ideas about Licklider.

Kain: Well I was talking about time-sharing rather than about like — it just seemed to me that it was something [that] made sense because there were a lot of resources going; when you were doing batch processing it was very, very difficult to get feedback. You know, I did a lot of keypunching back for some projects for courses and stuff in the basement of the [MIT] building 26. You had to go down and program the little thing to put on the drum, which makes the skip fields and makes it easy to fill out the cards that go into the machine. But then you'd put it in the hopper and you wouldn't get it back for hours, or overnight, or next day, or whatever. So it was very inefficient debugging and time-sharing was a good way to do that.

Misa: Going back then to this paper, "Symbolic Mathematics," you were actually interacting not with cards but you had an oscilloscope in front of you, if the picture is correct. In other words, that was some kind of interaction with computing that was not real time, in the sense of a control system, but that was not being programmed through cards in the kind of classical batch mode. There are pictures here where you can see on the screen mathematical notations, and then there's a certain thing you would underline and then you could declare the part of an equation that was underlined as a variable. That was all quite interactive. I don't know whether you call it time-share, but that's not batch processing and that was this BBN paper, "A Computer Aid for Symbolic Mathematics" that you and Lewis Clapp did.

Kain: That's interesting because I don't recall that piece of it. I really don't recall what the interaction was like. I mean, that may be totally accurate but I just don't recall it. Oh, here's pictures, okay.

Misa: I have pictures here.

Kain: Okay, well I haven't gone back to look at this paper for years.

Misa: Here's the one that struck me. Figure 7: Definition and Substitution of a New Symbol. This is not batch cards, this is a live screen; you have a light pen, you're interacting in a way quite differently than the classic punch cards. Very, very differently.

Kain: Yes, there are no punch cards involved with the TX-0. It's punched paper tape printed out on a Flexowriter, created on a Flexowriter, so you program by typing into the Flexowriter.

Misa: But the interaction as the program is running is just completely different from getting static feedback of a printout.

Kain: Oh, yes, it's totally interactive, but that doesn't mean it's time-shared.

Misa: No, no, of course not.

Kain: No, so I think it's totally interactive, there's no question about that, but I'm not sure that it's running in a time-shared context. I think it's a devoted program; you have the computer while this is going on and nobody else has access to the machine.

Misa: Right, but the I/O is not through cards,

Kain: No.

Misa: . . . it's not through tape,

Kain: No.

Misa: . . . it's not through printouts,

Kain: No.

Misa: . . . it's live, on the screen, that's what was making it useful for symbolic manipulation is that, as you put in the paper, that a scientist could be working with equations live, if you will.

Kain: And seeing them. That's right. I don't know if you want this, but as part of that history, you know the early part about interactions with light pens and stuff occurs basically at Lincoln Labs with the TX-2, which was the follow-on to the TX-0, which is where Ivan Sutherland did the interactive stuff. So that's the follow-on, and the SAGE system, which was developed basically at Lincoln Labs, but built by IBM I believe, involved light pens.

Misa: Light pens on big computer screens.

Kain: On big computer screens.

Misa: Radar screens, yes.

Kain: Radar screens and so on, but basically it's a computer display of radar information and light pen. But the light pen is only sensed by programming the dot and seeing the spot come back, seeing the light come back on the other side through the pen.

Misa: Were there any other aspects about those Cambridge and MIT years that you'd like to record at this time?

Kain: Another interesting historical thing about all this has to do with tablets as input devices. I don’t know if you’re aware about Herb Teager.

Misa: You mentioned Teager. Please continue.

Kain: Herb Teager was a professor at MIT and he developed a tablet method, which is I believe what Wacom uses now, which has to do with having a grid of wires, two-dimensional grid of wires, and signals on them, and a sensor which moves around on it, and could be positioned and could be providing computer input. I don’t think at that time much came of it. The timing of it was roughly 1963 or 1964; it’s the year that the Chinese blew up their first atomic bomb. And at that time, [I] had a consulting job with a senior colleague in the EE department who had a big business, and we were trying to sense train cars going by before they had the bar code stuff. We were competing with the bar code stuff, which got adopted to detect train cars going by at high speed on tracks. So we were down in New Jersey in a train yard running an experiment about this stuff with commuter trains that were going into New York City, and that was the weekend that the Chinese blew up their atomic bomb. This is now correlated with the fact that it’s roughly the same time that Herb Teager I think was involved with this project and that’s how I knew about this tablet stuff.

Misa: Okay.

Kain: Because he was on the periphery.

Misa: So the tablet was an alternative way of interacting with computers, getting signals into a computer without needing to do light pens. The tablet would be — not exactly like we would have today as a tablet — but something that you could serve to do handwriting, or write out equations, or do non-keyboard.

Kain: Non-keyboard, yes. Basically you could correlate that with pointing at a screen, you know, just like the Wacom tablets of today.

Misa: You mentioned several different summer jobs and I don't know whether it's helpful to connect that. You worked for IBM in Poughkeepsie, and had a couple of other things that you did during the summer during that Cambridge time.

Kain: The first summer job I had was after my junior year in 1956. I drove all the way to St. Paul from Louisville to earn $72 a week working with UNIVAC, which was UNIVAC then but we still had drawings that said "Engineering Research Associates."

Misa: Oh really, ERA hadn't been firmly stamped out yet.

Kain: No, had not been firmly stamped out yet, no. We were working in a warehouse that still exists on the southwest corner of Prior and University, behind the motel that's there now. Where the motel is was a big gravel parking lot. We were in the back of that warehouse and as they moved the sugar out of the front of the warehouse we took over more and more space. They were building the UNIVAC File Computer at that point, and it was still being debugged at the beginning of that summer. There were four of us who, like me, had finished junior year in electrical engineering, and our job was trivial. There were women doing hand wiring of the back panels of these machines, which were in those days I think vacuum tube machines.

Misa: These are the File Computers?

Kain: UNIVAC File Computer was the name of the product.

Misa: Magnetic drums.

Kain: Big magnetic drums and the very first models were programmed in a plug board. And then later in the summer they developed a more advanced version, which had a small drum that could hold a program, but the first ones didn't. I never did get to the level of learning anything about how you program that machine, but I think you could only have 48 steps in your program. A step could be search all the drums, of which there could be, I think, up to a max of 10 very large drums, of which there were two in a cabinet. So it could be a lot of hardware because these were very large scale drums, you know, two-foot diameter or something like that, and three or four feet long. So our job basically was that there were these wiring books, a loose leaf notebook in which each page was about one connector and each row line was about one pin; and you had to write; keep the information there was what part number of the wire is connected to this and where are the other one or two ends, because some of these wires were preconstructed with three ends.

Misa: Oh boy.

Kain: And there was a different part number, and the part numbers varied by how long the wires were on the connections, and so on and so forth.

Misa: Making a triangular connection between three separate pins?

Kain: Well not triangular, it's a center node, but connecting three pins, so it's making a little fan. And so our job was that as the engineers figured out things weren't working and they would make mods, then we would have to propagate those changes into the notebooks and make sure that the women who were wiring this stuff had made the corrections on the stuff that was already wired up because we were wiring up like serial number 10 already, and it still wasn't debugged. This machine is running in a warehouse with a five-ton air conditioner blowing at it and even in the summer, even with all the doors open — which were garage doors because there was a loading dock on the back side of this building with a train siding — sometimes the machine would shut down because it would be too hot. So that's where we were working; that was our job.

Misa: So a vacuum tube machine.

Kain: It was a vacuum tube machine, it made a lot of heat; I guess it made them some money eventually. So the next summer, I understand, the job we were doing was done by a computer. By 1957 they'd already programmed this function. In 1956, in that summer, they actually took their prototype and delivered it to their display room, which was here at Clarence and University Avenue, which is now the audio visual place.

Misa: Oh, it's not far away from right here.

Kain: Right here in Prospect Park, next to the park. They delivered it up there to the second floor to demonstrate, because that was the UNIVAC showroom, and they plugged it up wrong. They put 400 volts on a signal line and they burned up many diodes because there was no keying or something like that on connectors like there are now, so it's foolproof to connect up things; but then it was foolish to try to connect up things without reading and being very, very careful. So they spent about two weeks to find; I think they burned up like 400 diodes and they had to find them all and replace them all before they could actually show the machine off to people. So some lessons were learned here. So that was a summer job.

Misa: With UNIVAC, but also later with IBM . . .

Kain: That was the next summer.

Misa: . . . in Poughkeepsie.

Kain: In 1957, after I got my bachelor's degree I had a summer job in Poughkeepsie with IBM. My job was not very interesting or fulfilling or whatever, but what we were working on was the IBM 705, which was the newer version of the 702, which was the business machine; big computer kind of thing. I was involved with trivial stuff that had nothing to do with computing and it was really not interesting, and it was really unsatisfactory as far as I was concerned, but I got all kinds of peripheral stuff that was going on. And I'm sure that the big innovation then was "channels," which was independent control of input/output operations so that the central processor didn't have to do that step-by-step like it had to do in the previous versions, in the IBM 704.

Misa: It was a huge investment on IBM, in terms of the cost of the computer to have separate channels.

Kain: Probably. I don't know the scale of that, but yes. And that summer — just a total aside — at lunch time they would show us in the auditorium, one reel of a movie, and so you could go progressively, so you had like 20 minutes of a movie and then you get the next 20 minutes the next working day, and so on. When the movie was over you'd come and you'd expect to see the first reel of another movie. One day we went down there and we didn't get the first reel of another movie, we got the reel of a new movie showing the new property they'd bought at Yorktown Heights to build their research center, and about how people were going to be moved down there to work in the future. And that was summer I think that FORTRAN was announced.

Misa: 1957.

Kain: 1957, I think that's when FORTRAN was announced. FORTRAN development had been done in that facility in Poughkeepsie, in the research lab up on the hill.

Misa: Can you just step back and think about; seems like you'd had experience with a variety of things prior to coming to Minnesota. You'd had experience with UNIVAC, with IBM, with BBN, with the TX-0, with MIT; I don't think that was very common.

Kain: With Project MAC, too.

Misa: That's a pretty broad portfolio of experience that early on in computing. Was that common do you think, or is that something distinctive about your career?

Kain: I don't really know how common it was, I can't say. I mean, that's what I did. What can I say about this? That's what I did. Maybe it's partly because I did not get into the research lab and focus on a project, and be under the tutelage of somebody who was focused on a project, because Dave Huffman never was. He was just focused on what he was really interested in.

Misa: As opposed to some big project that may have been more controlled or directed students working with him, and that could've been you.

Kain: Very few students worked with him. Fred Hennie was one and I was one, and I don't know of any others but there may have been others because he would focus on some insight and solve it, then go on to find some other problem to work on, so he was kind of a free spirit. He wasn't focused on a project either.

Misa: Was he integrated with Project MAC or a little to the side?

Kain: I don't think he was integrated with Project MAC at all. I really can't tell you that but I think that's correct.

Misa: It's notable to have you go from Cambridge to Minneapolis. How did that come about?

Kain: Well, partly because I was feeling too [much] pressure at MIT about things I didn't want to do. I liked teaching and I was choosing, for example, to lecture the junior year course on electronics, where you teach all the EE majors in a big lecture hall. So you'd have 300 students up there in the seats, and you're lecturing; and I was criticized about why was I doing that by Bob Fano, who was in charge of Project MAC because it wasn't going to help my career. Why was I doing this? I said because I like teaching. That wasn't the right answer, of course, because I'm supposed to be involved with Project MAC and doing something useful for their project. Yes. So I left. I went looking for jobs and decided to come to Minnesota. I came to Minnesota in the summer of 1966.

Misa: Who recruited you? Department chair or was Shepherd still involved?

Kain: Gerry Shepherd was the vice president by then. Bob Collins was the department head, so Bob Collins was the one who hired me, in some sense. I think I initiated contact here, partly because I'd been here before; I'd been here in the summer 10 years before. Well, 10 years when I came; nine years when I was still looking for positions and decided to move. So I probably initiated the contact here; I had interviewed other places also but I won't bother going into that.

Misa: In 1966, Control Data was on a tear, growing very rapidly. UNIVAC was developing many different computers; Honeywell was in a very promising stage; it was kind of a kick-off moment for Minnesota in the computing industry. Did you see that as well or did you have some sense of this was an up and coming place?

Kain: I don't think so. No, I don't recall that. It was a place to come teach about computers and electronics, and that was the kind of stuff I liked doing.

Misa: What courses of types of courses did you end up teaching at the University of Minnesota?

Kain: Mostly I taught stuff related to computers and to electronics. Dick Halverson had been the computer person in electrical engineering, and his background originally was magnetics and he had worked with Bill Brown, who was a senior professor in the department, and done stuff working on magnetics. So got into computing through magnetic core memory thinking. So we taught courses in logic design and stuff like that. There was — oh goodness — we had some senior electives on analog computing, we had an analog computing lab, and we had digital computing.

Misa: Was that a REAC machine? I think the University had a Reeves Analog —

Kain: Yes, I think over in Shepherd Labs, in the Space Science Center. No, we had some desktop analog computers that we bought from a company called Simulators, or Simulation, or something like that, through a guy who I think was based in Chicago, as I recall. He came up and sold us on these machines, and they were nice machines, and we taught the students analog computing.

Misa: What types of problems would an analog computer be well adapted to solve?

Kain: Differential equation problems; problems where the system is to express a system of differential equations, and so you set up basically a network of integrators and amplifiers and stuff like that, and wire it up on a plug board, and put it in a machine.

Misa: And turn it on and it integrates.

Kain: And turn it on and it simulates the behavior and you watch a graph on an oscilloscope, basically.

Misa: Were there any particular types of integration problems or differential equation problems?

Kain: No.

Misa: . . . it was quite a general purpose . . .

Kain: Yes, it was general purpose for solving differential equations and you just plug up your thing. And the biggest problem the students always had was with scaling because you had to scale things because it had to go within the range of the electronic circuitry. So you had to scale your variables into plus or minus 10 volts, or whatever the number was. And you had to scale time. That would be a source of great confusion.

Misa: Some people look back and think that digital computing was an obvious winner but it's not so clear. Even at MIT through the 1950s there were many, many analog computers being used like you're describing to solve certain classes of equations. Digital computers were buggy and all that sort of thing. Do you have a sense that this was a viable path for computing, something that could be continued for some longer period of time or was is sort of a one-time accident that this guy from Chicago ended up selling you?

Kain: He wasn't the only person selling analog computers. Analog computers were being used all over the place in industry. They were being used by the aerospace industry to simulate behavior of aircraft, and so on. They were being used all over the place. It was a very long time before they began to accept that you could do digital integration, and do it accurately, and work this out right.

Misa: So it was a viable path.

Kain: It was a viable path in those days and then it faded.

Misa: History hasn't been kind to analog computing so I'm interested in your perspective to the contrary.

Kain: No, and you know there was an attempt here to have this — I've forgotten what they called it — but one of my colleagues who's named Steve Kahne was director of the center over there, where they had basically a mixed system. They had an analog computer that was connected to a little simple digital computer, I think, in the Shepherd Labs.

Misa: And Shepherd Lab went up, I think, 1968; does that sound right?

Kain: Late 1960s but I can't pin down an exact year for that. They called it a hybrid computer lab.

Misa: Analog and digital.

Kain: Analog and digital interconnected.

Misa: When you came here, you weren't necessarily focusing your attention the computer industry but you ended up getting connected in a consulting capacity with a number of the Minnesota-based computing companies. How did you get visibility with the companies? I think that connected to your teaching.

Kain: Well actually, I had two different connections here. The first connection, historically, would be starting with Project MAC; that Project MAC had industry people coming in to give advice on the occasional basis. They would have group meetings; I don't know how they describe what those were, but the people, the industry committee — let's call it that, I don't know what it was officially called — but the industry committee would come in and talk to Project MAC people in Cambridge about stuff. And one of the people who participated in that was Arnie Cohen, and so I knew Arnie Cohen from MIT days. When I came out here to Minnesota and wanted to get some connection, I contacted Arnie Cohen, who was a UNIVAC person. He got me a consulting job there, so I got consulting at UNIVAC around 1966, 1967; I can't pin down the exact date.

Misa: Soon after you came.

Kain: Soon after I came I started doing some consulting for them. That lasted a little while, and then in about 1969, I needed a summer job. We were building a house and I needed a summer job, so I contacted Arnie and he hooked me up with people at UNIVAC up in Roseville in the commercial side of UNIVAC, who were actually working on preliminary design specifications of a new mainframe, general purpose, large computer to be a successor to the 1108. So they had a guy who was acting like guru behavior, and so I was hired to basically tail him, follow him around, and try to document a consistent version of what Brian was espousing.

Misa: Reducing the brilliance to concrete form?

Kain: Yes, trying to produce a consistent form because it turned out this was not consistent stuff. I was hired to basically tail him and make this consistent. What actually happened . . .

Misa: Just to be clear, you were hired as a consultant and you were still working for the University.

Kain: No, this is a summer job now. I'm hired now as a full time person for the summer and the amazing bureaucratic thing about this is that when I first came in, they processed me like new people coming in, they took my picture and they gave me an employee number and all this stuff; and within a week they came back and said we've got to take your picture again. I said why? Well, we found out you've got an employee number already from 1956, when I had worked there. [Laughs.]

Misa: Presumably a lower number.

Kain: Well in fact, that's the piece of this amusing story; it was a four-digit number, and the people who were coming in all had five-digit numbers, and how come I had a four-digit number, because those weren't numbers. Actually, Arnie Cohen was still working there and he had a three-digit number because when they first started having numbers — maybe he even had two digits, I don't know — but they assigned numbers . . .

Misa: He went way back into the ERA days.

Kain: Yes, and they assigned numbers alphabetically when they first did it, so he had some low number and I had a four-digit number instead of a five-digit number. So I was an employee for the summer and then in the fall, of course I had to go back to teaching so I became a consultant because we did not have a consistent version of Brian. I became a one-day-a-week person and I'd come in and do the same thing, follow Brian around and work on documentation, writing consistent versions of this stuff, various aspects of this stuff, whatever. And that fall, that'd be 1969, probably, Brian went to the Western Computer Conference — I don't remember where it was, Nevada or California probably — and he got hired by I think it was SDS then, Scientific Data Systems, which later became Xerox Data, XDS.

Misa: Right.

Kain: He got hired by them so then he was leaving in February, so the big job was to get me to really finish this up. They pressured to really finish this up and get a consistent version before Brian departed for California, which didn't quite happen, but we got consistent Brian capture, whatever. So then I became the consultant and expert on architecture for this project. That went on for quite some time. I would come in once a week and talk to people who were doing software design and various other issues. We were doing instruction set design.

Misa: Was it common for people in the EE department to have that one-day-a-week consulting activity with local industry? The computer industry was all around.

Kain: Generally, I think the answer is yes, quite a number of people did have contacts with local industry. I'm not familiar with many niches of the department in that aspect of their lives, but I know the control folks did have connections with Honeywell, for example, doing control systems at Honeywell as consultant jobs. And I know that Dick Halverson had connections with various people, doing consulting work so it was happening, yes.

Misa: Early on, I think this was the late 1950s, Gerry Shepherd was for a time; I think they gave him a title of Director of Advanced Research, or something, at Control Data. Were there connections also with that company?

Kain: There probably were but I don't recall any, specifically, right now.

Misa: Just continuing, Dick, we were going down the path of your consulting with UNIVAC, and the project there involved your doing some translation of the "guru" as well as then ongoing consulting with computer architecture. Maybe you can follow up that story and continue moving forward.

Kain: Okay, by the way, his name was I think Brian Warner. So he left, then I became the expert on architecture, so I continued to come in and refine this work. Other people came from other places and introduced some other ideas, which we all tried to incorporate. Meanwhile, in effect we were going down a path which was not 100 percent compatible with the 1108 architecture instruction set and so on, because we were trying to generalize that and improve the performance of certain logical structures like subroutine calling and stuff like that to make it more efficient and more effective. So management knew that what we were doing was really not quite the same as the 1108 series of machines, but this project kept getting delayed and delayed, and there was another group of more aggressive manager-type people who eventually produced I think it was the 1195, which became a follow-on to the 1108. I think I've got my numbers right; that maybe slightly off.

Misa: We can check.

Kain: Yes. And so our project became the next one after that and eventually, I would come in and consult with a couple people who were worried mostly about the structure of system software and how would this all integrate into the whole scheme of things? I was actually showing up in their organization charts as a box attached to the manager, as a consultant to the manager of the group. The amusing thing was that there was a lot of turnover of managers and at one point, an organization chart came out in which I'm consultant to a box which is blank. I'm consulting to a blank person. [Laughs.]

Misa: Not necessarily a good sign there.

Kain: And so eventually, a manager came in, who was under pressure from his management to produce a product and have this done. So he promised his management that architecture would be done at the end of that fiscal year. I was always amused that UNIVAC's fiscal year started on April Fool's Day. So March 31st was the end of the fiscal year and this new manager had promised that by the end of the fiscal year, architecture would be finished. So after March 31st I couldn't show up anymore because I was the consultant on architecture and so therefore; it was after architecture was "done" even though the people that I was talking to on a weekly basis wanted me to continue to come in and talk to them, and help them out. I was not allowed to do that, so then I had to look for some other work. But before I go to that, I'll just add that that project continued for several years and eventually had — I don't know — four- or five-thousand people working on it. It was called Roanoke, which is generally not very well known about. And one night, overnight, it was abruptly cancelled by high management who decided that this is not compatible and this is not following our marketing strategy, or whatever, so the whole thing was locked up in a safe and not talked about after that. People were all reassigned within UNIVAC.

Misa: There are two pieces that I found from *Computerworld*, one was December 1977, "UNIVAC Halts Development." They were determining that uniform product line — they also used the code name "Roanoke" — can you make a comment on how it was a uniform product line?

Kain: No, sorry, I can't help you on that. I think that part of the problem had to do with the generic issue of upward compatibility. I think we were having upward compatibility in the sense that we were having firmware, or something like that, that was running old modes but I don't remember that level of detail anymore. Can't be specific or fill you in on fact there.

Misa: The piece from *Computerworld* said "Project Roanoke, which would have led to a uniform product line in late 1978 or early 1980" etcetera, and this was a sense that [Univac was] adopting instead a more, the article goes, "evolutionary approach." Another piece in *Computerworld* from April 17, 1978, said that 9,000 users were getting an upgrade, again as a result of Roanoke being cancelled.

Kain: You sent me those in an e-mail and I looked at them, and I was interested to see that they did say that some of the ideas that were used in Roanoke were going to be used in future products, so that was good. I never knew that happened.

Misa: I think it was real challenge to do a uniform product line. I suppose the IBM 360 was an example in people's minds; that was a hugely complicated hardware story and a massively complicated and expensive software story, too.

Kain: It was basically a firmware story there at IBM, with the system 360 stuff. My understanding was that they were dealing with all these different platforms. But by using different firmware they're making them all look the same at the assembly language level, so therefore everything was compatible.

Misa: So that was the technical mediation.

Kain: Right.

Misa: You ended up having a fairly long relationship with UNIVAC; a number of years that started in the 1960s and went well into the 1970s?

Kain: Into the 1970s but I can't pin down the exact time I left. I could go back, maybe, and find it in my income tax when I started income with them, but, yes. It was probably in the early 1970s.

Misa: Maybe we can return to the university setting. You said that there were some teaching activities that you wanted to record. Then we'll want to hear more about your connection to Honeywell where the security part is particularly notable and pronounced.

Kain: Okay, a couple of things. The first thing I wanted to talk about was that at the university, probably in 1969, a professor in metallurgy, chem- — I don't remember what the department structure was at that time — Morrie Nicholson was an entrepreneurial type of guy, and he persuaded IBM in Rochester that the University could do something for them, namely to have a retreat for managers at a retreat center for two or three weeks; I think it was three weeks to start with.

Misa: Three weeks? Okay, that's something.

Kain: I think it was three weeks; which was called Modern Technical Concepts for Engineering Managers. I was a participant. I would come out there and lecture to them about computer architecture for two or three sessions over a couple days. So go out to the retreat center, spend a couple days out there, lecture about computer architecture, and go back. I think — in fact I'm pretty sure — that some of the people who were the people who developed the — was it called the AS system? — from Rochester?

Misa: AS/400.

Kain: AS/400 stuff.

Misa: Extremely well known.

Kain: Yes, and I think some of the ideas I talked about were incorporated in that series of machines, from this Modern Technical Concepts course.

Misa: AS/400 was a huge technical and economic success for IBM Rochester.

Kain: Yes. And this Modern Technical Concepts course went on for a number of summers. One summer it was very amusing because as I drove out there to the retreat center, the managers had been there already for 10 days or something like that, hearing about backgrounds of physics and chemistry, and stuff like that, metallurgy and whatever [pause]

Misa: This is somewhere close to Rochester itself?

Kain: No, the first ones were out at Chi Rho, which was a church retreat center on Lake Sylvia Peninsula. And one of the times later was at Koinonia, I think, which is on the other side of that lake, out past Annandale.

Misa: So they were taking managers out of Rochester.

Kain: Taking them totally out of context for these several weeks to focus on thinking about this stuff and not the everyday demands of what we have to do on our current projects, you know, just to refresh their thinking and get them revved up again. It was a good idea. So one of these times, I drove out there and they announced on the news radio, as I was driving up there, that IBM just announced this new portable computer, which was the 5100 I think it was called.

Misa: Oh yes. [Laughs.] I lifted one of those once; it's portable but not very.

Kain: Well I'll tell you about that. So I went out there and I said something about I just heard this. Oh, it just got announced; they didn't know it had been announced and they said they had worked on it and they wanted to call it a portable computer and the marketing people said you can't do that. So the engineers called up TV stores in Rochester and asked what's your biggest portable TV set?

Misa: Weight-wise?

Kain: And how much does it weigh? The biggest so-called portable TV sets weighed more than the 5100, and therefore the 5100 was called a portable computer because the engineers insisted that it was portable. [Laughs.] And that was a curious machine because it ran on magnetic tapes and basically ran APL programs and BASIC, I guess.

Misa: So this set of courses for IBM Rochester continued across a couple of summers?

Kain: Yes.

Misa: That was sort of an ongoing summer school.

Kain: Different groups of people; different groups of managers would come; but it went on for — I don't remember — four or five years at least.

Misa: That's an important connection between IBM Rochester and the University of Minnesota that I think we don't know much about.

Kain: Maybe you don't know much about it at all.

Misa: Should know more.

Kain: Morrie Nicholson had started it. Anyhow, so back to the other consulting line; I had been teaching; when I came to the university, the university was doing courses for; basically the idea was to let practicing engineers in local industry get master's degrees while they’re working full time. When that was initially set up, which was before I came, there was a connection to IBM Rochester.

Misa: Before 1966.

Kain: Before 1966. There were a couple of classrooms in the top floor of the aerospace building, which had three TV cameras mounted from the ceiling, pointing to the blackboard.

Misa: That’s the university’s Akerman Hall.

Kain: It’s now but it wasn’t called Akerman Hall then.

Misa: Aerospace.

Kain: Yes. And they had these three TV cameras pointing at the blackboard, and on these blackboards there were painted outlines for what the TV cameras could see, because there was no adjustments of this; the TV cameras just were there. So that was connected, broadcast by microwave — I guess first by cable — to IBM Rochester and there would be a live class, and it was done in the evening. So in the evening you’d come in and teach the course; and you’d have to stand, you’d have to work within the confines of these rectangles on the blackboard. There were three cameras but there were only two lines to Rochester so the one on the left was always on, and the one on the middle or the right, there was a switch underneath the chalk tray so the instructor could control what they could see. Then there was talk back, through loudspeakers in the ceiling.

Misa: Were there students also in the classroom while you were teaching?

Kain: Yes, we had the local engineers, people from Control Data and Honeywell, and whatever in Rosemont, and whoever else here in town would come in and sit in the classroom, so there'd be a live class.

Misa: Live, and then also televised.

Kain: And then it was also televised to Rochester and the people in Rochester had this talkback, question back system. So later that was replaced by the UNITE system, which was then modeled on the Stanford system, that Stanford had developed, which was broadcast stuff. But it was broadcast from the top of the Foshay Tower, I think originally, and then later the IDS Tower. They had like a 10-watt transmitter and so eight watts was radiated, and two watts was an addition and went to a relay tower in Hader, which went to . . .

Misa: Ten watts, very impressive. [Laughs.]

Kain: Yes, right. And that went to the IBM plant in Rochester so they could then see courses. There were several classrooms; eventually, I think, three or four classrooms that were wired up. Those then were daytime classes, and they had the live students of the daytime class. And then the engineers in local industry would then have an additional antenna and an FM talkback channel in this educational TV bandwidth, a separate allocation of bandwidth from the FCC. So then we would have people in local industry watching the daytime classes.

Misa: So it's another connection with the computer industry and the EE department, doing this professional master's.

Kain: All of industry in town, because there were mechanical engineering degrees, and math or computer science degrees, being offered and so on; there were other degrees being offered also in this besides EE. It was not just an EE thing at all. It was called it the UNITE system, did I say that?

Misa: Yes.

Kain: That was University Industry Teaching Engineering or something, but UNITE is the acronym. Through the UNITE system, I think, I was known to people in local industry because I was teaching these courses about logic design and computer architecture. By the way, I was also teaching — back up a minute on the teaching history — one of the courses I started was Automata Theory. I taught Automata Theory, and I think we called it Advanced Computer Architecture or something like that because we were teaching Basic Computer Architecture as a senior elective, and then we had this other switching theory, and we had this architecture course. And then I also taught automata theory for which there was no good book that explained everything so I wrote a textbook, *Automata Theory* [1972] about that, which does have a typo in it because they made more typo errors in a complicated formula, which was wrong in the galley proofs, and so I fixed it and they made more errors when they fixed it so it is not right in the printed version, but it's close. [Laughs.] Anyhow, so through that connection I was known in local industry and eventually I was approached by some people at Honeywell who were working in the systems and research center, which was up on Stinson Boulevard at the time, about whether I could join them on a consulting basis. So I worked on some projects up there, which eventually led to the secure computing work. Do you want to get into that now or do you have some more questions about the previous?

Misa: The Honeywell work was eventually involved with security, so when we go into that . . .

Kain: Not at the beginning.

Misa: . . . no, not at the beginning, but I was just going to pick up a paper of yours from 1977 that Honeywell Modular Microprogram Machine M3, the paper you did with Douglas Jensen for the Proceedings of the Fourth Annual Symposium on Computer Architecture. That wasn't about security at all . . .

Kain: No.

Misa: . . . that was about computer architecture. So at the time, you'd done computer architecture for UNIVAC, so that was one of your areas of expertise. This is not a security paper at all.

Kain: No.

Misa: What was the connection then to Honeywell, Systems and Research Center, they have it here at 2600 Ridgeway Parkway Northeast?

Kain: That's the one I was talking about.

Misa: That's the same one?

Kain: Yes, that's the same place, up on Stinson Boulevard; yes that's it. I mean, the front of the building is on Stinson Boulevard, but Ridgeway Parkway is [the] side road. So the address is on the side road. Same building; it's a complex of buildings up there, by what's now called The Quarry, the big shopping center, next to the Sunset Memorial Park Cemetery; across the street from the cemetery.

Misa: There is one statement here that struck my eye with the security angle in mind; it says "M3 consists of an application independent kernel machine, to which application dependent functional modules are attached." I take it the whole idea was to develop functional modules highly customizable, so that you could do some sort of hardware emulation of many different types of computing. The kernel concept, of course, is one of the notable concepts in security; but this is quite independent, I think. Is that a fair reading?

Kain: Yes. I think that's correct. I had forgotten about that stuff.

Misa: I've seen all these old papers.

Kain: Thank you. [Laughs.]

Misa: It's kind of interesting, Dick, because you worked on many fields. It wasn't security alone, or architecture alone, you were working on several different lines.

Kain: I drove the graphics department absolutely crazy with that. I think that paper was presented at a conference that was out in D.C. area, or Maryland.

Misa: It could be.

Kain: College Park or something like that. I drove them totally crazy because I like graphics and overlays, and so I made lots of overlays in different colors and they were going berserk because they had to do all those separate sheets, and here I am, I'm not an employee and I'm making all this trouble. Eventually there was a doctrine passed down about you can’t have too many overlays, and you can’t have too many colors in your view graphs, for anybody in the whole system in Honeywell. [Laughs.] I overloaded the system.

Misa: So originally your connections to Honeywell were more computer architecture-related.

Kain: Yes.

Misa: Were there problems that Honeywell was working on at the time that made architecture a notable problem?

Kain: I don’t know. Basically, the Systems and Research Center was really a contract [research] place; it was not connected to their main line computer business at all. It was not connected to the development of hardware, or of the big computers, or anything like that, at all.

Misa: It says in the note on the bottom of the page that Honeywell Modular Microprogram machine, supported by Air Force Rome Air Development Center, under contract so-and-so, U.S.A. Ballistic Missile Defense Advanced Technology Center. So it’s a contract, then.

Kain: Yes. I was not part of getting the contracts. I just was part of solving the problems that came along with, as part of the contracts. So that was the contract; I don’t think that — and I may be speaking out of turn or incorrectly — but I don’t think that there was really any connection between the Systems and Research Center stuff and the commercial business until we got in the security business.

Misa: Okay, so it was contract work but not necessarily part of Honeywell’s commercial business.

Kain: The computer business, right. This was just a research center that was supported by contract work, basically, yes. And I don’t think I was involved in anything that ever connected; I did some work one summer down in Burnsville or Bloomington; I guess it was Bloomington, because it was north of the river; which was where we were studying how to simulate various things and I wrote a paper about that ChimpNet stuff, which appeared in a conference somewhere along the line. It’s a very obscure little thing.

Misa: The next step in my notes is your work on PSOS.

Kain: Yes.

Misa: Earl Boebert ended up doing a genealogy, beginning in the 1980s, he’d worked on SCOMP, the secure computer processor. So he ended up saying SCOMP led to PSOS, which led to the Secure ADA Target; which led to LOCK, one other thing had been

Sidewinder. So there's a genealogy that made sense to him, and I think that he worked on each one of those as a Honeywell employee. And I think you worked on some, but not all of them. Is that a fair way of saying it?

Kain: Yes.

Misa: You were still a faculty member at the university; you had a consulting relationship with Honeywell.

Kain: Yes.

Kain: You and Earl ended up working very closely together and publishing many papers, obviously having a long-time collaboration. His genealogy is his career as a manager with each one of those projects. You were involved as a consultant, in essence, again still working from the university.

Kain: Yes, that's correct.

Misa: Would it make sense to go through each one of those and relate your role? Because that's a major lineage that's worth exploring.

Kain: That makes sense from your end but I'm not sure I can distinguish which project was which.

Misa: Earl had the same problem, too, because . . .

Kain: It was a continuum.

Misa: . . . it was a continuum. I don’t think that was the phrase he used but he said the same thing; it wasn’t that they were unique but there was a longer term series of projects. Each had their names, they had to be funded, they had to be organized, they had market relationships; but this seemed to me to be a long-term emphasis that Earl had worked on in trying to do secure computing. It’s worth pointing out that this is a long trajectory.

Kain: It’s kind of a logical evolution of the thinking, yes.

Misa: Would it make sense for us to go through each one of those in sequence or would you prefer to start some more general discussion about the problem of secure computing and what you inherited from the 1960s and 1970s?

Kain: Okay, let’s talk about that.

Misa: Multics fits in here, in some fashion. I don’t know whether it’s necessary to understand your particular role, whether we need to start with Multics, or start with the Orange Book.

Kain: [Laughs.] Well, I don't know the best way to do that, when you bring up all those options I would say maybe a place to start is Multics, and basically there is a paper with Earl Van Horn, about capabilities. I don't remember the exact timing of that, but I was I think a reader on his thesis with Jerry Saltzer.

Misa: This goes back to your time as an MIT faculty member.

Kain: Yes.

Misa: You were also a reader, I think, on Peter Denning's thesis.

Kain: So I was familiar with the notion of capabilities from my MIT time in the early 1960s. The next time this all came up, I think, was that summer that I was working for Honeywell down in Bloomington. I got a call from — I don't remember whether it was Earl or it was the marketing guy that was associated with the group, whose name was I think Jim Stevens — and they called me and asked me if I could go on this whirlwind trip. So this would be about 1980, I think. I was getting divorced and I didn't have any money, so I said if I can go to the airport and pick up an airplane ticket from the counter, I'll go. And so that trip was basically the project which started with NSA wanting to implement PSOS. Should I explain what PSOS is?

Misa: Go ahead and just describe it so that it's clear for everyone.

Kain: Okay. PSOS stands for Provably Secure Operating System, and was developed by a large group at Stanford Research Institute, SRI, and involved a number of different people but we'll just work with Peter Neumann and Karl Levitt. So people in general had been working on the notion of trying to prove programs correct, which was a big problem, and it went for many, many, many years and they're still working on it. But the basic idea was that somehow you would write specifications, and then you would write the program, and that you would then be able to prove that the program met the specifications and that would be a systematic way of proving it. It might involve so many steps that it would have to be done with computer assistance, but that was the basic idea. And so the SRI group had convinced NSA that they could develop a way that you could build an operating system which would be secure, where security was defined in terms of the Bell-LaPadula model, which is colloquially referred to as the Orange Book because the specification book had an orange cover. So that was a report from, I think, MITRE; so that was supposed to be the description in computerese of the military security hierarchical system. The idea was the you could have blocks of data and so on, which would have security properties and that the access and ability to read and write, and modify, this information would be limited to individuals based on the security policy, based on the Orange Book. So the Stanford SRI people had devised this system, which was claimed to work — we'll talk about that in a minute — claimed to work based on having capabilities. The idea of having capabilities is that the capability object itself is not really data, it's a descriptor of data. It points to data and then it contains within it information about access permissions to this data. So if you're the wrong kind of person or the wrong kind of program and the permission in this capability does not allow you access, then you can't get there. So the underlying mechanism is supposed to enforce this policy of what does the capability say? What is the program trying to do? Who's the program running on behalf of? And is this access therefore allowed or not allowed? And if it's not allowed, it will get intercepted and the program will terminate whatever will happen. If the program terminates, that might be information so you have to be careful about how you work this because that might be what they call a covert channel, because you might be able to infer from the fact that certain behavior happened that there's information here. So that's another tough nut to crack.

Misa: You have a paper on that too.

Kain: [Laughs.] I don't remember too much about that but that's where we get in trouble here. Basically, the SRI people said okay, here's this design and this can be proven to be secure, and it's all based on capabilities and the fact that the capabilities can control this. So that was a big report, which was the product of a project from NSA to SRI. So then NSA decided that they would like to have somebody build this and that among other things, that this should be implemented on a 32-bit minicomputer. Alright, so the Honeywell marketing people found out about this because the RFP had come out, and thought that this project had been as we say, wired to somebody else. I don't know who they thought it was wired to, but they thought it was wired to somebody not Honeywell. But they decided they would make a stab at it, so they made a marketing decision to make a stab at it. So that was this whirlwind trip; [it] was to be the first steps at making a stab at it. So the first step of this was that Honeywell did not have a publicly announced 32-bit minicomputer.

Misa: But 16-bit machines all over the place.

Kain: There was a 32-bit machine that was about to be announced, but had not been announced, which had been designed out at their facility in Billerica [Massachusetts], which is something they had bought from — what was the company [that was] out there before?

Misa: Control Computing?

Kain: No, that's not right. Anyhow, they bought this business from some other place and they had this facility out there in Billerica where they did the design. I think they had some manufacturing out there in that area, as well; Billerica being a suburb of Boston, northwestern suburb of Boston. So the first flight of this whirlwind trip was fly to Boston and go out to Billerica and spend the morning in a conference room, and learn something about this 32-bit unannounced minicomputer, and some kind of strategy for implementing this PSOS system on this machine, to be formed [into] the basis of a project. Now, the project from NSA was to involve a complete design of a system, so that was hardware and software. The Honeywell folks didn't really want to be in the software business on this thing, they just wanted to do the hardware side so the question was — two questions, really — how can we use our 32-bit design, our 32-bit minicomputer, as a basis for this implementation of PSOS; and the second one was how can we find somebody to cooperate with us to do the software part? And because of the way that the RFP was phrased was that whoever did the software was going to be the primary contractor. Okay, so we sit in the conference room into the early afternoon learning about this 32-bit machine and figuring out how we can [pause]

Misa: Which is still being developed?

Kain: No, it was pretty much finished but it wasn't being marketed yet. I don't know whether they were still working on operating issues, but it was pretty much finished, so it was a pretty stable kind of entity that we were going to build on. So one of the engineers that we worked with a lot there was named Herb Terakawa. I don't know if you've run into his name or not but Herb Terakawa was a good engineer there and we had a lot of interaction with him over the years. So we went out, met with Herb and other people, and then drove to Logan Airport and flew to San Francisco, and then the next morning went out and touted our design to Ford Aerospace.

Misa: [Laughing] Just recently received, okay.

Kain: [Laughs.] Just recently devised, yes, like yesterday; to try to convince those people to participate in doing the software side of this development. We've got the hardware, we know how to fit the software on the hardware, and you can do the software.

Misa: Just wondering why Ford Aerospace was the chosen partner?

Kain: That's beyond my knowledge.

Misa: That was your assignment.

Kain: This was where my airplane ticket took me. [Laughs.] Remember, I'm just on an airplane ticket here, and going along, and making notes, and contributing to design issues, and so on and so forth. I think partly because they had some of the people who had worked at SRI there at that time. Then we got on the plane and flew to LA, and did the same thing with another entity the next day, and it became clear by lunchtime that they were not really interested in talking to us. It may have been that they were talking to somebody else who was competing with us, I don't know. But anyhow, they were not interested in this so we got an early airplane back home. The Ford Aerospace people agreed to work with us so Earl and me, and maybe Herb came out from Boston, and we spent like Labor Day weekend all holed up writing a proposal basically figuring out how to do all this, and writing a proposal and drinking some nice wine. [Laughs.] Silver Oak cabernet is very nice wine.

Misa: The proposal was to NSA or whatever contracting agency they were using at the time.

Kain: Yes. I don't know the details about how; I think it was NSA. I think it was directly to NSA but I'm not really sure because I was never involved in that level of detail about the legal negotiations or any of that kind of stuff at all. So, surprisingly, we won the contract; it wasn't wired to somebody else, we got it. Then we started working with Ford Aerospace people.

Misa: So that was basically a computer development contract to implement SRI's PSOS design, with a 32-bit minicomputer from Billerica; Ford Aerospace to do the software; this could've been something pretty notable.

Kain: Could've been, yes; could've been. Only problem is we figured out after a few months that PSOS didn't really work. So this presented a problem.

Misa: How did you find that out?

Kain: We sat down, I think it was Earl and I probably together but I don't remember the details about this, but we basically figured out some scenarios by which capabilities could be copied from one place to another, and because the capabilities carried with them permissions, then permissions could get copied and used, and maybe the permissions were not valid in the new context where the capability had been copied. It turns out that the PSOS report then had some stuff in it which I think was called "copy controls," which somebody, Earl or somebody had inferred that maybe the SRI people figured out at the very last minute when they were writing the final report that dammit, it doesn't work.

Misa: Oops.

Kain: Yes, oops. So they stuck in “copy controls” because that was supposed to solve the problem. It didn’t. I don’t remember the details out of my head right now to reconstruct this, but anyhow we can construct scenarios by which you could subvert the Orange Book, by copying these things around in ways that were permitted, or by fudging copy controls, or whatever, I don’t remember the details about any of that but anyhow, basically, the PSOS didn’t work. So basically as part of the reports, we had a bunch of progress reports; I would go to these meetings out in California at Ford Aerospace and I don’t remember how frequent they were — it must have been monthly or something like that — and I went out there several different times and one of the things that repeatedly happened was okay, this whole project was not classified in the military sense, however, it was stated from the get-go that this was not to be talked about. Even though it was not classified, it was not publicly acknowledged that this project even existed so we couldn’t talk about this. At Honeywell, we basically commandeered a conference room that had two doors with locks on them. So we locked those doors and had our meetings and our work in that room.

Misa: You had physical security.

Kain: We had physical security because there was always looming over us the question, the uncertainty because NSA people wouldn’t commit one way or the other about whether they were or were not going to impose a military classification on this project or not. So Honeywell people were being very careful to behave as though it had a classification on it and trying to keep everything confined into this room, for example. Some of the other people in the group would tease me when I did showed up, ‘Are you here to work on No Such Project?’

Misa: No Such Project.

Kain: NSP, which is of course, the power company [Northern States Power]. So are you here to work on NSP? [Laughs.] No Such Project.

Misa: No Such Agency, too, I suppose; the NSA.

Kain: Yes. So I would disappear behind the locked door and work on this stuff. So the people out at Ford Aerospace were not doing very much. We’ll be kind and say they were not doing very much, and they were; one of the issues that they were being harassed about by the agency, and the agency had some consultants I think, who came to these meetings. The [question asked by the] agency and the consultants was when are you going to have a computer to facilitate the development of this software? The Ford Aerospace people argued we don’t know whether this is going to be classified or not so we don’t know whether we have to get a separate computer and park it away in a hidden place to preserve the security aspects of this, and so on and so forth. So they basically didn’t do much of anything and as a result of many months of frustration like this the agency cancelled the contract.

Misa: So partly because of slow performance, but also partly possibly because of the flaw in copying of capabilities.

Kain: No, I don’t think it had anything to do with the fact that copying of capabilities; that the PSOS didn’t work because we were devising other ways to try to get around this. We were going to do — not PSOS — but something that was secure. We were going to follow the Orange Book but not do the PSOS way because PSOS didn’t work. You couldn’t really use capabilities as they were using them.

Misa: So implementing the Orange Book, just not through the PSOS mode.

Kain: I agree, that’s accurate. So that’s what we were trying to do, but the Ford Aerospace people weren’t really doing much of anything so the agency cancelled the contract. Then Honeywell eventually decided, okay, we’ll do the whole thing and we’ll write a new proposal. But the government regulations at the time, maybe still now, said that if you’re a subcontractor — which is what we were —

Misa: The prime contractor had been Ford Aerospace.

Kain: The prime contract had been with Ford Aerospace, exactly. And they had defaulted, in some sense; they had been kicked out. But we couldn't rise up from the ashes for nine months, so there was a nine-month interval where nothing officially happened. I think Earl and I were talking a little bit, but nothing officially was happening for quite some time there. And then we got the project back. Meanwhile somewhere in there, Earl devised a scheme which I think he called Copy Permissions, which was not the same as these "copy controls" that had come from the SRI folks, but it was something to try to limit the access or the ability to move around capabilities in ways that could subvert the Orange Book constraints.

Misa: Because it's not the capabilities themselves, it's their copying that subverts the Orange Book.

Kain: Yes. I mean, if the capability has the right permissions in it, and you enforce the capability, then the permissions will be enforced and if they're the right permissions — right meaning conformity with your security policy which in this case was Orange Book — then obviously it's going to work right. But the problem was that these things could be subverted simply because they could be copied without any what shall we say, messaging of the permissions in the copying process. If you had somehow let's figuratively say, if you had a little magic box — that the process of copying had to go through our magic box — if the magic box knows the security policy and enforces it, it will produce on the other side, your copying capability, which has been modified so that wherever it ends up, it won't break the policy. It won't crack your security scheme. Okay, so Earl devised this thing and filed for a patent; put my name on it; even though it wasn't my; I didn't know he had invented it all but we had worked so much together that he had put my name on it as a second inventor; filed for a patent. As the patent process went on, of course, in filing for a patent you've got to put it into patent-ese, and in the course of that, you've got to have somebody who knows patent-ese write what's called the specification part of it, which is where all the technical details reside in the patent. It turns out that the lawyer or whoever wrote that up, did it wrong, so that the specification of this patent was really not what we were inventing and didn't work. So the question, back to the Honeywell patent lawyers, was what do we do now? And the answer was, file for another patent.

Misa: A second patent, rather than modify the first patent.

Kain: Don't modify the first patent, let's file a second patent. So we filed a second patent. So the two patents, to start with, were based on this "copy permissions" idea and it turns out this doesn't work either so we still had a puzzle. Meanwhile, we get the renewal of the contract, and I don't know exactly the timing of all that but it doesn't make any difference logically. So then, how does this go? Anyhow, we get the project and we're working on it, and the government is happy and they're saying at some point — and this timing may be off — they're saying you need to expand this, you need to get more people working on this. We went back and said how can we do that if we can't talk about this? [Laughs.] How can we get anybody to work on something that doesn't exist in the outside world? Oh, I guess it's okay, you can talk about it.

Misa: If you're to recruit you've got to talk about it. The chicken and the egg.

Kain: Very much, yes, Catch 22, or whatever. Perfect example. So we were allowed to talk about it. So in the course of that, they hired some people and I was asked one afternoon when I was out there, I was asked to brief one of these people new to the project. I don't think he was new to Honeywell but he was new to the project, about the project and what we were doing. In the course of explaining to him what was to be done in the project and how we were approaching the problem, it occurred to me how to solve the problem, while talking to him.

Misa: The capabilities problem?

Kain: Going back to the general problem about — that we haven't gotten solved, really — about how do we make this tight? I figured out in my head how to do that. I, of course, didn't say anything it to the new guy, but I then quickly went and wrote it in my notebook. So that became the use of — I've forgotten what we called it but anyhow — with domains and type enforcement. So that was the type enforcement/domain stuff. That's basically my invention, when I'm briefing Scott Hansohn into joining the project, which is the third patent, which really ***does*** work. So that patent is also Earl Boebert and Richard Kain; Earl first because he was the employee. It's all licensed to Honeywell.

Misa: You mention the Honeywell notebooks — I'm just noticing that on your box across the way — those are the notebooks that you're talking about.

Kain: Yes.

Misa: Those are highly interesting to us.

Kain: Yes, okay.

Misa: Just to fill this in: from Honeywell's corporate point of view, this fits into the trajectory of essentially getting Multics from GE, enhancing Multics during SCOMP, and then moving to PSOS. Did you have any previous contact, besides MIT, with Multics [or] with that trajectory, that was part of Honeywell?

Kain: No, not really. The only connection, which becomes of peripheral interest a little bit later here in this conversation maybe, is the fact that we were using Multics as an e-mail server. [Laughs.] Way back then.

Misa: Right, but SCOMP was not a project that you had direct contact with?

Kain: That's uncertain; I'm not clear about that. I can't answer that.

Misa: Do you want to move forward and get the Secure Ada Target project? Or is there something that should be further described.

Kain: No, I think that's fine. I think that Secure Ada Target was yet another name for this community evolution. Was that actually the name of the contract after Ford Aerospace fell apart? It might've been because this is the era where DoD has developed Ada, the programming language, and insisting everything happen in Ada. So somebody decided to call it Secure Ada Target, but it's basically the evolution of the same project; it's a contract renewal, I think.

Misa: Why was Ada?

Kain: It was Ada because the Department of Defense said everything should be Ada, that's why.

Misa: There's this 1985 IEEE conference paper that you and Earl, along with W.D. Young and S.A. Hansohn [pause]

Kain: Bill Young and Scott Hansohn.

Misa: H-A-N-S-O-H-N, so a non-standard spelling of Hansohn.

Kain: Yes, he's the one I was indoctrinating into the project when I figured out how to do it.

Misa: Bingo, the same Scott.

Kain: Yes.

Misa: So Secure Ada Target issues system to design and verification . . .

Kain: The other guy is Bill Young.

Misa: It says Bill Young is University of Texas, consultant to Honeywell, just as yourself.

Kain: Yes. He had been here and moved to Texas. He had been moved, and then moved to Texas, [when] that paper came out.

Misa: So you appear University of Minnesota, consultant to Honeywell; and then Bill Young, University of Texas, consultant to Honeywell. But he had been a Honeywell employee. When you read this paper, it's not at all clear where *Ada* fits because you're describing a computer system, and it happens to be that it will run Ada.

Kain: Yes, that's right.

Misa: And if it runs Ada, it might run COBOL, or FORTRAN, or who knows what else.

Kain: But for marketing to the government in those days, Ada was what you wanted to run.

Misa: That’s right.

Kain: So we’d become Secure Ada Target, SAT.

Misa: SAT, yes.

Kain: So that’s strictly nonsense marketing stuff.

Misa: This may be why Earl Boebert described this whole lineage as being a continuous evolutionary development rather than being driven by a new project coming through. So that was something entirely different from anything that followed. This was a continuation of this attempt, in essence then, to build a computer system that would observe the Orange Book . . .

Kain: Honor the Orange Book.

Misa: . . . honor the Orange Book, and then deal with some of the shortcomings of the PSOS model.

Kain: Well, to honor the Orange Book, the PSOS model doesn't work so you've got to honor the Orange Book. You've got to throw out PSOS because it doesn't work.

Misa: Which is the criteria that the computing world is [pause]

Kain: The overarching thing is let's do security, and what kind of security do we have right now? We have Orange Book so let's do Orange Book.

Kain: So the Secure Ada Target, as you describe it, is basically a way of continuing the line of investigation and research aimed at making clear to the DoD that this was going to Ada-complaint and secure.

Kain: Secure, Orange Book secure, yes. So it's basically how do you rename your renewal project? As far as I'm concerned, it's a marketing effort of what are we going to call this now? But it is strictly an evolution, yes.

Misa: It's an evolution instead of any kind of abrupt change. If you'd read the paper you'd think that this could be a quite separate development.

Kain: But you see, there's the whole issue about whether we're under wraps or out in public. So we probably don't come out in public until it's called Secure Ada Target, or something like that. So it's probably the renewal after Ford Aerospace went away.

Misa: Okay.

Kain: Maybe the renewal then was called Secure Ada Target, I'm not sure about that, but that could be where that name arose.

Misa: Were there any notable technical refinements in the SAT, Secure Ada Target project, that you can recall? If you need to look at the paper, it's not a memory test or anything.

Kain: [Laughs.] Oh, this is basically describing Earl's scheme to tagged objects.

Misa: Yes, tagged objects. I think you're right.

Kain: Yes, tagged objects, and this is before domains. Yes, this is tagged objects, so this is before my stuff with Scott Hansohn. Oh, Scott is in here, that's interesting; okay.

Misa: As are you.

Kain: Yes. Basically, the idea here was that we were still operating on this 32-bit machine because we're talking about level six architecture, which is the 32-bit machine here; that we were going to have to add on this specialized processor, which was going to be the enforcer of security, a tagged object processor, which is what I was describing earlier about if you had this magic box that you could pass the capability through and get it out the other side and it would be correct. So that's what this tagged object processor is supposed to be doing. It's the magic box.

Misa: This precedes, then, the type enforcement?

Kain: The type enforcement stuff is not in here, right. So this automatic proof stuff, and I think that's Bill Young's stuff, is the automatic proofing.

Misa: Gypsy.

Kain: Yes, that's Bill Young's stuff.

Misa: Oh, and then it appears as being from the University of Texas, but then that must derive in some way arrived from Honeywell, is that correct?

Kain: This was done before he moved to Texas.

Misa: That's what I mean. But Gypsy is described as being from the University of Texas, but through Bill Young, it's got a connection, at least an origin back to Honeywell.

Kain: Gypsy is not part of this project. Gypsy was a tool which had been developed at Texas, is my recollection, as a part of this whole issue about proving software correct. So there were people at Texas who were very active with this question. I can't remember their names.

Misa: Separate from [pause]

Kain: Totally separate from this.

Misa: I misunderstood. Okay.

Kain: Separate. They come beforehand, so they precede this, and this was their attempt to prove programs correct, and Gypsy was their tool. And so Bill Young was involved in this kind of effort and eventually went to Texas and worked with those people down there.

Misa: But [Gypsy] had a separate origin at Texas.

Kain: Yes.

Misa: Not at Honeywell.

Kain: It had nothing to do with [Honeywell]. It was a tool which got hooked into this project and became useful.

Misa: I have to keep reminding myself that you're doing this as a consultant while still a full time faculty member, teaching classes at the University of Minnesota. You're not a Honeywell employee; you're not an NSA employee; you're doing this essentially on consulting time.

Kain: Yes.

Misa: It seems to me that — I mentioned this before that you have a whole string of article and papers with Earl — it must have been quite a significant part of your technical career at this time.

Kain: Oh probably. Yeah.

Misa: There's a paper you did with it looks like Earl Boebert, Dick Kain, and Bill Young; you're all described as Honeywell Systems and Research Center; "The Extended Access Matrix Model of Computer Security." This was published — the version that I have — in ACM's SIGSOFT Software Engineering Notes, Volume 10, in August 1985.

Kain: I don't recall this.

Misa: Here, again, take a look at it.

Kain: Oh, this does type enforcement in it because it’s got domains. Okay, so there are domains in here so this is the type enforcement stuff of mine, and then put into Gypsy by Bill. I don’t recall this. I didn’t go to this conference obviously. But yes.

Misa: It may be in that paper; there’s a little comment that Gypsy takes 30 pages of proof to go through one potential [pause]; you know; it’s quite detailed.

Kain: [Laughs.] No proof is simple. No. That’s why it has very limited possibilities and there have been many efforts including Barbara Liskov at MIT. Attempting to prove stuff secure; correct, I mean.

Misa: Back to Earl Boebert lineage, SCOMP, PSOS, Ada security; LOCK for him is the next one. Did you work on that? Also LOCK — I should just put a footnote on it — is connected with a spinoff, as I understand it from Earl’s interview, of Secure Computing Corporation from Honeywell. So it’s almost like this work is continuing but it gets spun off as a separate company from Honeywell in 1989 or something. But LOCK would’ve been the successor project that Earl identified.

Kain: Yes. My recollection of this history — which really is a corporate history issue rather than a technological issue, because it’s all linear progression — is basically that at some point around that time, a little before that time, Honeywell decided they were no longer in the computer business. They wanted to exit the computer business. We were still in the Systems and Resource Center, and because what we were doing was considered to be part of the computer business, then they didn't want to have any part of it anymore. So the question was what to do with this group of people, to which I was just consultant but not much anymore. I think I still was just a little bit. So Earl was basically going around trying to sell the group to various entities and eventually got interest from these people out in Baltimore, I think it was — I'm not sure of that — who were venture capital people who eventually agreed to basically take over this thing, and that became Secure Computing Corporation. Initially, I think it was just run by the venture capital people, and then later they issued stock.

Misa: Earl said there was a public offering later but he had left by that point in time.

Kain: I'm not sure whether he was or not but I got annoyed because basically they were selling my technology and they never invited me to invest. The stock came out at 16 and closed at 48 the first day. [Laughs.]

Misa: Oh, okay, well a couple of shares would help out.

Kain: Yes. So that was the basis of Secure Computing Corporation, which eventually got bought by McAfee, which has now been bought by Intel.

Misa: Did you end up having a consulting relationship with Secure Computing as you had with Honeywell?

Kain: No, I was out of the picture then.

Misa: What kinds of activities were you continuing to do? Were they more architecture oriented or security oriented?

Kain: I think at that point, I was mostly being Director of Graduate Studies [DGS] and teaching in the department and not doing research.

Misa: A full time job.

Kain: Yes, because Director of Graduate Studies was like two-thirds time job because we had a couple hundred graduate students at any time, and 600-700 applicants every year.

Misa: Do you want to say anything about that? There's one final paper I want to make sure we talk about, and that's your final paper with Carl Landwehr.

Kain: That's a long story too. Want to do that now?

Misa: Sure. Unless you want to say something about the DGS responsibilities.

Kain: No.

Misa: This is “Access Checking in Capability-based Systems: a Review and Taxonomy” that you and Carl had done.

Kain: I’m familiar with that. So this was with Carl Landwehr, who was at Naval Research Laboratory at the time and this is the history. Basically, Earl gave a paper at a workshop — I don’t know the name of the workshop — with some kind of title about capability-based systems; inherent something-or-other capability-based systems. I don’t remember the exact title and it doesn’t make any difference for the purposes of this history. He wrote this paper and went to this workshop in Maryland. I didn’t go and I wasn’t really connected with it at all; this is one of Earl going off and doing his thing kind of things. Carl Landwehr attended that workshop and after the workshop was over, with some whatever delay, Carl sent an e-mail to Earl — and you have to realize this is 1982 or something like that, whatever — [pause]

Misa: The paper finally is published in 1986, IEEE Symposium on Security and Privacy.

Kain: Maybe it’s 1984, then. So Carl sends an e-mail to Earl asking for clarification about the meaning of capability-based systems, or the meaning of whatever — I’ve forgotten exactly what the initial question was. So Earl got this question and decided he would pass the buck to me. So he said Dick, can you answer this question? So that starts a long e-mail interaction with Carl Landwehr, over e-mail, for many months in the course of which we clarify a lot of issues about exactly how does the structure have to work to really be able to enforce a security policy when you are using capabilities as your basic assess control element. When some program is trying to access some piece of information, what's the magic box that limits that, and that's the capability, and how does that work so that you can enforce a security policy? So after many months of this interaction back and forth on e-mail, from the acoustic coupler in my kitchen [laughs]

Misa: Through the phone system.

Kain: Yes, and the little portable teletype machine type of thing. We finally get this all worked out; mostly I get it worked out answering his questions. So when we got it all worked out, then Carl sends me an e-mail that says, 'I've been saving all these e-mails — which I hadn't been doing — one of my colleagues is interested in this and would it be okay if I shared all of this with my colleague?'

Misa: This is all more or less informal interchange?

Kain: Yes, we were just having this conversation. Basically doing a back and forth, trying to resolve the logical problems of getting this consistent with the security policy so we can be sure that the enforcement flags that say what you can and can't do are correct. So that when an access is made and it's enforced by the bits in the capability that allow permissions or not, that that's consistent with the security policy, so therefore the whole system is running the security policy. So the question is what do you have to do to make this all work right. Basically, we figured out what are the design options? What is the design space of; you’ve got a process, you’ve got data, you’ve got somehow this intermediary and it’s kind of capability-based, [pause]

Misa: You’ve got segment, capability, data security attributes, process, security attributes, reference one.

Kain: The segment is the block of data to which the capability is the monitor of access. Is that clear?

Misa: I think so.

Kain: So basically, going back to the history, Carl said can I share this with my colleague; to which I responded why don’t we write a paper for the security conference. So that’s the paper you’re referring to. I basically wrote most of the paper and Carl wrote an example or two into the paper. We had, I think, one phone conversation, and the rest had been done by e-mail. Then we both went to Oakland and met for the first time, shook hands, I presented the paper, and we shared best paper award with another paper. [Laughs.] So that was a big success of e-mail interactions in the early days of e-mail. This was in 1984 so this was, oh-my-God, 30-some years ago. We did all this by e-mail and got something that people really appreciated and we got a best paper award, which was a nice little gold star. I don’t think we got any money for it. [Laughs.]

Misa: I think people point to it because it's not merely descriptive, but it's also diagnostic, it seems.

Kain: It sets up all the possibilities; are in there.

Misa: Yes, so it not only tells you what they are, but also where. You've got these five diagnostic questions, and depending on the answers to these five diagnostic questions, it is a diagnosis of where it is, pinpointing what part of the system that's being evaluated, where the weaknesses are.

Kain: I never thought about it that way.

Misa: It's not a descriptive taxonomy. Just like, you know, there's like giraffe's out there, but it also tells you *where* the weaknesses are. I think that's how this paper is understood.

Kain: I think that's a fair statement. I wouldn't have thought about it that way, but I think that's a fair statement. Also I would have thought about *when* do you have to check *what*? Or to make sure that the capability is correct so that access is in accordance with the security policy so there's a whole question about the timing. When you give the capability to the process, have you checked it? Or when the access is made, do you check it? Those kinds of questions are there, and I think there were four or five possible ways of solving the problem in there. So it's a taxonomy of all the possibilities and you can think about it as if you don’t check at this certain point, if you’re doing a certain scheme and you don’t check at this certain point, then you’re in trouble. So you can think about it as a diagnostic in that sense, but you can also think about it as the other side of the coin as dammit, you better check it here because otherwise you’re in trouble.

Misa: You’re in big trouble.

Kain: Yes, if your scheme is the rest of this scheme.

Misa: So even though Earl doesn’t appear on the title, he was essentially the connection that got you into the conversation with Carl.

Kain: Yes, I got into the conversation with Carl through e-mail because Carl asked the question of Earl, and Earl referred it to me.

Misa: Right, it was your conversation with Carl, that’s clear.

Kain: It was totally my conversation with Carl. Earl has no part of this conversation and probably really not part of the discussion that I recall.

Misa: Well he passed the problem to you probably because he had other things to do. In his oral history, as Earl describes it, he had all this management to do and essentially trying to keep large enterprises afloat and fighting various organizational battles. He had very different responsibilities.

Kain: That's what I said; he was off trying to sell this group.

Misa: That's right.

Kain: To somebody. That's a big responsibility.

Misa: And then the last paper that I had a chance to print out was from Security Technology 1996, "A Further Note on the Confinement Problem." That's Boebert and Kain, and this goes back to the confinement problem in convert channels, right?

Kain: I'm blank.

[PAPERS HANDLED.]

Misa: Butler Lampson had written something on the confinement problem and that's his early paper where he identifies and italicizes *covert channels*. And here, the paper seems to be moving that into an era where software agents can come in, applets are coming in, how in the world do you set up a secure environment with mobile computing.

Kain: Well, you know, I don't know whether I had anything to do with writing this paper; and the LOCK business is Secure Computing Corporation thing; but this is all the type enforcement thing that I invented. He said back here that it was invented in 1984.

Misa: Just to be clear, by the way, it's Earl at Sandia, after Honeywell, then after leaving Secure Computing as well.

Kain: Yes. He went out to Albuquerque to Sandia.

Misa: Exactly.

Kain: See, he puts my name on it here, but I don't remember that I had much to do with this except that I'm the inventor of the underlying technology so he put my name on here. [Laughs.]

Misa: Okay.

Kain: It's like a logical follow-through to all this stuff; and basically they built this Sidewinder thing and I remember Earl calling me up and telling me how happy he was. I don't know all the vagaries and all the ins and outs of all this, hackers on the web and all that kind of stuff, but basically when they developed this system and they decided okay, we'll put it online one day and see what happens. So they put it online and they had a gazillion attacks on it right away because it showed up as a new thing online and there are all these people looking at what's online and seeing if they can break into it. And nobody broke into it; nobody succeeded and they were very pleased.

Misa: Sidewinder.

Kain: Yes. But that was after I was not involved. I mean, I knew Earl socially, but I didn't really have a role with that. They were just implementing the stuff based on my ideas.

Misa: Earl had this sort of extended trip through and maybe, Dick, you can summarize. It seems to me that the type enforcement was a really important contribution that you ended up adding to this ongoing conversation. Were there other things that you feel that your work was particularly crucial in the clarification of security ideas, besides type enforcement?

Kain: I think type enforcement is the end of all of it. It's just an evolution of all these things, and contributing by figuring out that PSOS doesn't work, for example. That was important to know.

Misa: There's a much more detailed discussion that we might have at some point in time about your Honeywell notebooks. I'm wondering if you could just describe for the recording here, the general content. It's a box full.

Kain: There are only three notebooks in the front. This other stuff is old computer printouts that I don't know if anybody cares about them but for some reason, have them in this box and so I brought them up from downstairs, where they've been sitting in this box. I looked at this, it's been several weeks ago. Here's three notebooks. There's all kinds of papers stuck in. These are big computer printouts, and some reports. Hardware progress report. Briefing for the SAT kickoff March, 1984.

Misa: Bingo.

Kain: You probably don't have that. This is just fun; we had a progress report at our progress meeting with the customer and their consultants at our little locked up conference room in Systems Research Center; and then after we'd done our presentation we were invited to leave the room and we came back, they held up scores. So here's the paper that says 9.8 out of 10 for the score for our presentation. [Laughs.]

Misa: I'd say that's pretty good. [Joins laughter.]

Kain: Somehow, I saved that. It's in my box.

Misa: We'd be very interested in the possibility of adding these to the Babbage archive, because it's notable materials that center squarely on security, and also then on your career.

Kain: Okay, yes. This is old e-mails. These are e-mails with Morrie Gasser, you know him?

Misa: You mentioned him. I don't know much about him.

Kain: Morrie Gasser was, I think, primarily a MITRE person, who was also very interested in security and was a consultant to NSA. I don't remember why I saved those. I'd have to read the whole content and see why on earth they're still in the box.

Misa: Just thinking about our conversation today, are there any other observations you'd like to add to help understand the security evolution dilemma?

Kain: I don't think so but, you know, there's a generic, really tough nut here about — which is true when you have the internet going everywhere — about how do you control what's going on? And it's really, really hard. We tried to do it; then it gets much more complicated when you're trying to do policies that are not hierarchical like the Orange Book. It gets much messier. The Orange Book is the easy case.

Misa: Is the easy case?

Kain: The easy case, yes, because everything's hierarchical and there's things you can't write up and you can't read down.

Misa: That’s right.

Kain: You *can’t write down* and you *can’t read up*, I said it backwards. If you can, then the horse has left the stable and you’re in trouble, it’s running wild.

Misa: It seems critical, almost, to me that even the relatively simple case is really tough to build a workable computing system. I’ve heard it said that many of the early machines that were built to Orange Book specifications, the security took all the processing power, essentially unable to do anything useful because it was secure.

Kain: Because it was working so hard to be secure.

Misa: So this is a difficult problem.

Kain: Yes, and you know the old joke about what’s the most secure thing?

Misa: Tell me.

Kain: A brick.

Misa: A brick.

[Laughter.]

Kain: You have a secure brick, it doesn’t reveal anything, nobody can change it.

Misa: It doesn’t get copied. It’s more or less stable until somebody throws it through the window.

Kain: Yes. This is really hard. And in some sense, the thing that we were trying to do in a lot of this — you even saw this thing in this tagged object processor notion of a co-processor, you know, the co-processor being like an add-on on the bus, where you’re basically offloading the security policy into a magic box, which is sitting on the side — and then in modern technology, that could be its own little computer system, which is sitting there just enforcing security and it’s just passing out things that get enforced in the hardware. Every time you attempt to make a memory access, you’ve got to check to make sure that it’s okay. Especially when you have anything multi-processed and time shared, the whole memory is potentially open to any access and some pieces of the memory might contain top secret and some of the pieces may contain unclassified, so you can’t just allow any arbitrary memory access, any arbitrarily generated memory address to go through. So if the processor is attempting to make a memory access there’s got to be a check, and a check has to be in accordance with security policy. You have to interpose this checking, even if you’re not trying to do military security, to just protect users from each other. So you’ve got to have some kind of check in there, and security policy is about making that check conform to a security policy and that’s the tougher nut because the security policy can be something very complicated. If you start taking the Bell-LaPadula model, for example, and adding all kinds of categories to the security, and then you’ve got all these little bins at these various levels of the hierarchy and it gets really messy.

Misa: Even at the same security, “need to know” means that some people can see if they have a need to know, and other people don’t have the need to know, so they wouldn’t be able to see that.

Kain: Right. And then the complementary question — which I don’t whether you’ve come up with it — is the integrity question. You familiar with that?

Misa: The Ken Biba model; I think he was someone who basically said that, as opposed to Bell-LaPadula — which was about security — he said that data integrity is a whole different dimension . . .

Kain: Yes.

Misa: . . . very important for finance, less important maybe for the DoD or military.

Kain: There’s no question it’s important; of course it’s important for DoD, because the integrity question is do you really believe this information?

Misa: Okay, sure.

Kain: If really believe this information, you might lob a bomb. If you don’t believe this information or are not sure about this information, you’re not going to shoot the missile.

Misa: Right, okay.

Kain: So it’s very important in DoD context because who do you believe? In some sense the word is trust. Do you trust this information? Do you believe this information is an accurate or correct, or whatever you want to call it, reflection of the state of the world. That’s an inverse problem to the security problem. Who can play with data? Who do you believe?

Misa: To know when data’s been . . .

Kain: Corrupted.

Misa: . . . corrupted or changed or manipulated or whatever.

Kain: Yes. So that you no longer trust it so much. Or maybe somebody came by and went through it and reviewed it, and said this is really good, so your trust level should go up.

Misa: That’s right.

Kain: Mr. Trustworthy Expert or Mrs. Trustworthy Expert came in here and said yes, this is really good stuff. And then your trust level, your integrity score should increase.

Misa: Should go up.

Kain: Yes.

Misa: I think it's really interesting to work on a historical project like this where history is definitely a part of the computing problems that we face today. This isn't an easy problem like just getting a computer to run faster, but it's a really difficult problem so I appreciate your comments today.

Kain: The other general comment about the history of all this business is the flipping and flopping about *where* is computing happening? Namely, you start with mainframe computers, back in the 1940s. You introduce time-sharing, so now you had the possibility of people sitting somewhere, and the computing is somewhere else, and it's on a shared resource. You get to personal computing and you bring that back; so now you've got your own local computer, which is your resource, and then you go to the cloud, and it's the same problem as time-sharing. It's the same thing all over again. It's all these repeats in the computer business, so it's the cloud is just time-sharing all over again but it's phrased differently.

Misa: Yes. Well this computer utility model that was popular in the 1960s, also you've got recognizable echoes of that today.

Kain: Logically it's the same thing. It's a question of basically who can afford to have the processing power; where can you put the processing power; and of course, the contemporary case is vast memory. Where can you afford to put that stuff within the general scheme of things? But it's just repeated *déjà vu* all over again. You have a lot of that going on here, a lot of that.

Misa: That's part of our project is hopefully we can get some lessons from history and feed those forward.

Kain: Part of the problem, of course, these days I think, has to do with the fact that people look stuff up online and only see the stuff that's recent, and don't see the history; and don't realize they're inventing something that was done 30 years before and doing it all over again and calling it a big new idea, and it really happened 30-40 years ago. What's the big deal here?

Misa: Thank you so much for our conversation. It's been a real pleasure.

Kain: You're welcome.