Catalyzing Computing Podcast Episode 27 – Global Security and Graph Analytics with Nadya Bliss (Part 1)

The transcript below is lightly edited for readability. Listen to “Global Security and Graph Analytics with Nadya Bliss (Part 1)” here.

Table of Contents

[Intro - 00:10]

Khari: Hello, I'm your host, Khari Douglas, and welcome to Catalyzing Computing, the official podcast of the Computing Community Consortium. The Computing Community Consortium, or CCC for short, is a programmatic committee of the Computing Research Association. The mission of the CCC is to catalyze the computing research community and enable the pursuit of innovative, high-impact research.

In this episode, I interview Dr. Nadya Bliss, a CCC Council Member and the Executive Director of Arizona State's Global Security Initiative. Before joining ASU in 2012, Bliss spent 10 years at MIT's Lincoln Laboratory, most recently as a founding group leader of the Computing and Analytics Group. In this episode, we discussed her time in Lincoln Lab, what a federally funded research and development center (FFRDC) does, and the history of graph analytics. Enjoy.

[Moving to the USA from the USSR - 01:00]

Khari: Today via Zoom we have CCC Council Member and Director of the Global Security Initiative at Arizona State University Nadya Bliss. Nadya, how are you doing today?

Nadya: I am pretty good. How are you?

Khari: Doing well, doing well, all things considered. So let me ask you, where did you grow up and how did you first get involved with computer science?

Nadya: So I was born in St. Petersburg, Russia. What is now St. Petersburg, Russia, at the time it was Leningrad in the former USSR — the Soviet Union doesn't exist anymore.

I have always loved math. I was one of those little kids that didn't want to be a ballerina or a princess or a queen. I really wanted to be a mathematician. So from a very, very young age, I just remember really, really loving math.

And then when I was almost 13, about 12 1/2, my family came to the U.S. and I continued to love math, but I was a little bit concerned that I would need to spend too much time in school and I really wanted to make sure I had a job when I graduated college; that was important. I had a ton of college loans and...I knew I was going to have a ton of college loans...I was obviously still in high school at the time.

[Laughter]

So I decided to try a computer programming course in high school and took computer science AP classes in high school. Fortunately, I had a really good math teacher that was also my computer science teacher. There's some beautiful parallels between computer science and math that allowed me to kind of maintain my passion for both, and I realized that there are aspects of computer science that are basically mathematics. So that's how I got into it.

Khari: Ok, so in high school?

Nadya: Yeah, I would say so. The Soviet Union was a funny place, a lot of the technical education was very advanced; so I think I had some very basic programming classes back in fourth grade in the Soviet Union, but in terms of a career, I think I decided [in]...it was either ninth or 10th grade of high school in Ohio — [a] public school in Ohio [in] a suburb of Cleveland — that I would major in computer science in college.

I realize it's really weird. A lot of ninth and 10th graders have not picked their majors, but I am that person that has been planning everything pretty much since I was six.

[Laughter]

Khari: Yeah, that does seem pretty early. So how old were you when you moved from the Soviet Union to the United States?

Nadya: I was 12 1/2 when my family moved.

Khari: So did you notice — I mean, obviously there is some degree of culture shock — but in terms of, like, school and math and computer science, did you notice any differences when you moved?

Nadya: Oh, lots of differences. And it's strange, you know, 12 1/2 it's a funny age, especially for us nerds. Generally that time period is a little bit challenging, but then on top of it you're that foreign kid in class that is not necessarily totally fluent in the language [and] it's a little extra hard. So I'll tell kind of two little anecdotes.

One is not really related to the technical side, but one of my early memories of when we moved to the U.S. was the first time we went to a basic American supermarket. In the Soviet Union — and the Soviet Union was falling apart during that time as well — you often had to stand in line for food and kind of buy the food that was available that day. I remember going to one of those supermarkets and it was like this feeling of light and all the aisles full of all this food. It was just surreal.
I mean, it's not a very sophisticated memory, but there's something about the abundance and choice that just was surreal.

As far as the technical side, what I think was very interesting is...I was taken aback by the negative connotation, especially, *especially* as a teenage girl that came associated with being good at math. In the Soviet Union...and again, I was in St. Petersburg and I come from a family that's very highly educated: my mom is an engineer, my grandma is an engineer, and my other grandma was a medical doctor. So it was sort of expected that you're good at math and it didn't matter if you were a girl good at math or who you were good at math.

I pretty much realized early on in high school that being good at math is not cool and this is not going to be one of those, you know, movie or Saved by the Bell high school experiences, and I'm just going to put my head down and work through it. That was strange. I thought that was odd.

[Working at Lincoln Laboratory - 06:30]

Khari: So, prior to joining GSI I know you spent a long time at MIT's Lincoln Laboratory. What is that? Could you talk a little bit about that?

Nadya: Yeah. So MIT Lincoln Laboratory is a federally funded research and development center. You sometimes hear people refer to it as an FFRDC. It's not the only FFRDC, there are other FFRDCs in the United States. For example, Sandia National Labs is an FFRDC, Oak Ridge National Laboratories is an FFRDC.

What's pretty special about Lincoln Laboratory is it is both formally part of MIT — Massachusetts Institute of Technology, the university — and also the fact that it is a prototyping national lab. There are a number of FFRDCs that do analyses, Lincoln Laboratory actually builds things like processors, prototypes, algorithms, and technologies. It doesn't produce them, but it builds these prototypes, and that's pretty cool.

When you're sort of thinking about the type of things that you want to do...for someone like me, I really, really, really love research, but I also really wanted to have that research be connected to something tangible. So when I was graduating from Cornell with my Master's, I applied very, very broadly. I sent applications to companies in the financial sector and big companies that had IT departments. Microsoft was another place I potentially considered going to.

Lincoln Laboratory to me was one of those places that really combined all the elements of research and application in this one place and that's what I really loved about both selecting it as a place to go work and then spending 10 years there.

Khari: So, can you explain a little bit more about what an FFRDC is for people who don't know?

Nadya: Yeah. Essentially, Lincoln is a Department of Defense FFRDC. So it works very closely with Department of Defense on applying the most advanced concepts in technology research to big Department of Defense problems. There is this huge emphasis on technical rigor, combined with a requirement for applicability to the mission of the Department of Defense.

For example, an FFRDC like Lincoln Laboratory, typically does not do basic research, but often it does early-stage applied research and then applies it to the problems in the national security space. It is a technology FFRDC, so the mission of Lincoln Laboratory is technology in support of national security. The vast majority of the technical staff there fall into disciplines like electrical engineering, computer science, mathematics, and physics. It's like this place that is just full of nerds and it's amazing.

[Laughter]

Khari: What's the difference between basic and applied research? I know I've heard the terms like 6.1, 6.2, and 6.3. Could you explain those distinctions?

Nadya: Yeah. Typically, when people say basic research, it's usually very, very early stage research. And I would also like to distinguish between basic research that is curiosity driven and basic research that is application driven. So, typically in those very, very early stages you are just testing out basic concepts. For example, in computer science, it may be a pseudocode for an algorithm or a set of equations to try out; if you are, for example, working out in graph theory and looking at sparse matrices and how they apply in [the] context of graph theory.

Sometimes, if it’s curiosity driven, it could just be because I want to try this out. I want to try to understand what a projection of this type of matrix is going to look like in a two dimensional space — you're just curious about what would happen. If it's application driven you may be wanting to solve a particular problem, but it's still in those very, very early stages. Once research becomes applied, typically there is a particular set of specifications that are coming that are informed by an application. For example, you may know the scale of the data that you're working with, so [you know] how many data points [there are] or the types of patterns you may be interested in identifying.

[Read more about DOD research appropriations structure here.]

Khari: Ok, that makes sense. So can you talk a little bit about the kind of work that you did while you were at Lincoln Laboratory?

Nadya: So, when I left Lincoln Laboratory I was the group leader of the computing and analytics group and essentially we did a whole bunch of different computer science type things. We were one of the, kind of, major computer science groups in the laboratory. I would say, when I first started at the laboratory there wasn't as much computer science, there was some, but it was very much an electrical engineering shop. Then, by the time I left, I think there was an appreciation of computer science as a discipline as opposed to [a] support discipline like developing software for an electrical engineering project.

The types of things that we did in my group were...for example, we worked on advanced computer architectures [and] instruction set architectures, particularly for either novel applications like graph applications, graph theoretic applications, network applications, or using emerging novel devices. For example, photonic interconnects as opposed to electronic interconnects.

We also had...basically all of the high performance computing (HPC) in the laboratory was under the oversight of the group that I ran. A lot of my early work at Lincoln really focused on high-performance computing. Basically, how do we take complex signal processing codes and map them onto multiprocessor architectures? A lot of that work was happening in the group, and then we had a pretty significant effort on all types of analytics, but the biggest one was on graph analytics: analysis of networks, analysis of relationships between entities and trying to see if there is [an] emergence of patterns in those.

[High Performance Computing - 13:13]

Khari: Ok. Well, we'll talk more about graph theory in a bit. In the prep for this, I know you said that you were the group leader for the computing and analytics group and that Lincoln Labs' high-performance computers were under your oversight. What's involved in doing oversight of an HPC?

Nadya: Oh, you know, it's really funny. One of the first times I thought, "Oh, wow, this is what a leadership position is like," was when....So we really thought about how do we, essentially, have green computing. Not just sufficient computing, but efficient computing per power unit, computing that uses minimum power. At the time, one of the ways to implement it was using these containers. I remember the location where these containers with high performance computing existed were somewhere by a river in Massachusetts. And I remember getting a call — I think it was on the weekend — that the river had flooded, and I was thinking, "What am I supposed to do when there's a container that's flooded?"

[Laughter]

I mean, there's lots of people that are involved that help you manage through that, but that's one of the examples of what's involved. But on a more serious note, I think what's interesting about high performance computing is how it is both a resource and a research area at the same time.

We had amazing groups of people that...I mean, that effort has now spun out and I think Lincoln Laboratory now has a high performance computing facility that's its own thing. I believe Dr. Jeremy Kepner runs it, and I worked with Jeremy since I first started at the lab. So, you have a ton of amazing researchers doing this very advanced early-stage research on how to make the most efficient parallel and high performance computation, and then you also have a ton of brilliant technical people that work with the rest of the laboratory to help parallelize codes, and make sure code scales well, and develop these systems that allow for [the] minimum intervention of an individual when you have to rewrite a code. It’s this interesting balance that lives in between research and application. But then there's also flooding. So that's kind of a good example.

[Laughter]

Khari: Yeah. So I don't know how involved you still are with the field, but...

Nadya: Not very much.

Khari: Ok. Well if you can speak to it, has it changed in any way since when you first got involved?

Nadya: So the thing that I will say — and this is probably an artifact of just getting older — is one thing that I've been noticing a lot of the really challenging problems persist. For example, mapping codes onto computer architectures is not a solved problem and still is the subject of theses for Ph.D. students. I mean, that's a very interesting area.

The other thing that I would say is, when I first started at Lincoln Laboratory I remember a lot of discussions about the slowing down of Moore's Law — essentially the fact that we're taking for granted that computational speed is doubling about every year and a half; which, of course, Moore's Law is not about computational speed, but about the space, how many transistors you can put on a chip. But then, you know, the effect of it is that it's been easier to get faster processors but essentially very little work on the programming side. What's interesting is I was just recently in another meeting and people are still talking about Moore's Law.

So, there is both a ton of innovation and then there are these persistent themes that stay, and if you're lucky enough to pick good problems, you can both make impact and then also see longevity of that research. In many ways, high-performance computing has changed drastically because now we essentially have a high-performance computer in our pocket — because our iPhones are all high-performance computers — and computational speed and just the amount of time we can do both locally and remotely has drastically changed. But those big research questions like Moore's Law, novel architectures, architectures with different memory access patterns, and programming models for increasingly complex architectures — those problems, there's still things to work on in the community.

Khari: Of those problems, are there any that you personally find more interesting or you think apply more to topic areas that you're interested in?

Nadya: For me, what I think is particularly interesting...let's see. I would say, I think the underlying processor architectures still have trouble with the shape of matrices that could be used to represent networks and graphs, and I think that's interesting. If we go back to the beginning of how I got into this, ultimately my passion has always been the algorithmic and mathematical aspects of computer science. I think hardware is brilliantly interesting but it is where that coupling happens that to me I find personally the most interesting. So, I think when I worked on developing computer architectures for graph algorithms, that was the most interesting type of computer architecture and processing system work for me personally.

Khari: Ok. So you said the shape of matrices. What do you mean by shape?

Nadya: I'm sorry, I probably shouldn't have said the shape, it was more the sparsity pattern. If you think about, let's say, traditional fluid mechanics type of codes, the sparse matrices, even if they're sparse, often have a very, kind of, regular structure in them. So there are either dense rows, dense columns, or dense diagonals...well not dense, but diagonals that are filled out. If you have a sparse matrix that's very structured from a computational perspective you essentially end up having something that you can almost optimize on as if it is a dense matrix.

However, when you go to a sparse matrix derived from a network like, for example, the type of social networks that Facebook or Twitter or whatever use, they do not look like those matrices at all. Some of them have power law structure, but there's other very irregular structures that you can take advantage of. There's no dense blocks, and that's what I meant. So, it's essentially the sparsity pattern.

If you, essentially, take your matrix and create a little square where you color in where the non-zeros are it looks like just a weird random pattern. It's not actually random, generally it's not at all, but it looks very irregular and it's hard to take advantage of computationally from a performance optimization perspective.

[Graph Analytics - 20:47]

Khari: Ok, that makes sense. So this might be a good segue way then. What is graph analytics — which you said is one of the things you worked on [and what] your dissertation was about — and why is it important?

Nadya: Maybe we can start a little bit with, if it's OK, if I can say what a graph is. Yeah?

So a graph is essentially a mathematical abstraction or mathematical construct that allows you to represent entities — which in graph theory are referred to as vertices — and relationships — which in graph theory are referred to as edges. And you can represent graphs as edge-vertex pairs; so, you essentially can encode, like, these two vertices are connected so that makes an edge and these two vertices are connected so it's essentially a set; or you can represent graphs as an adjacency matrix, which essentially gives you that space matrix representation. You basically have rows and columns, represent your vertices, your entities, and then there's a non-zero entry in the matrix if those two vertices are connected.

So that's a graph, basically an abstraction; and why graphs are important is — and of course I'm going to sound like someone who likes graphs here — almost anything can be represented as a graph! Things like relationships between people, organizational structures, financial transactions, social media communication, and Girl Scout troops are graphs. Relationships between proteins, for example, are often studied as graphs. Roadways [and] transportation networks are graphs; and people use the word graph and network interchangeably. A network tends to be a more descriptive term. Whereas a graph is like the formal mathematical definition.

I even think — and this is probably because I sort of loved this kind of stuff as a kid and I'm very much a nerd — just, I generally think of concepts as graphs. If I think about the brain and how things are interconnected I'll be like, " Oh, this thing over there it connects to this thing over there" and that essentially is a graph.

The other thing that's really interesting about graphs is that it changes the distance between entities. For example, if you think about social networks you have your geometric distance, which is in a two dimensional space, but in graph space you can actually be really close to someone in DC even though you don't live close together at all. There are these fundamental principles that encode these relationships in this highly multidimensional way.

I mean, obviously, the Internet is a graph that's kind of an obvious one. So they're just everywhere and to be honest, they're not that well understood. And the other thing that I think is really fascinating is as we've scaled into this world of connectivity the scale of graphs around us has changed, and I think that also creates a more interesting space. I'm going to pause here because I feel like I could just go on and on and on about graphs.

Khari: So I was actually going to ask you about...because you sent me your PowerPoint presentation from your dissertation, and it has a couple of different examples of different graphs.

Nadya: Yes.

Khari: The first one was something I read about prepping for this, the [Seven Bridges of Königsberg] problem, which I believe is sort of the original graph theory problem.

Nadya: Yes, so that's the other thing that's so cool about it, right? I mean, graphs have existed way before computer science! And if we go back to the six-year-old Nadya, to me that was just like, there's so much mathematics that underlie it but it's also a core area in computer science as well.

Khari: So can you explain the Bridge of Köenigsberg or I can find it on Wikipedia and read the….

Nayda: Hold on a second, let me just pull it up here. I could explain it but I just want to make sure I explain the right one. Yeah, traversing...it's traversing the bridges, right?

Khari: Yeah, um, I can just read the Wikipedia summary, I guess.

Nayda: Yeah, that's probably better instead of me trying to repeat it.

Khari: Yeah. So, the Wikipedia summary: "The Seven Bridges of Königsberg is a historically notable problem in mathematics. Its negative resolution by Leonhard Euler in 1736 laid the foundations for graph theory and prefigured the idea of topology.

The city of Königsberg in Prussia at the time, was set on both sides of the Pregel River, which included two large islands that were interconnected to each other or two mainland portions of the city by seven bridges. The problem was to devise a walk through the city that would cross each of those bridges once and only once.

By way of specifying the logical task unambiguously, solutions involving either reaching an island or mainland bank other than via one of the bridges, or accessing any bridge without crossing to its other end, are unacceptable. Euler proved that the problem has no solution."

So that was in the 1700s. I saw there were a couple of other examples, I don't know if any of them are of interest: Hamilton's game or Zachary's karate club?

Nadya: The karate club is the one that I think was studied a ton before there was this explosion of social networks. People sort of studied the evolution of the club, and the breakups of the club, and all of those [things] of the clubs. So yeah, there's been a bunch of them.

I think the difference is that even with the Euler....I mean, even there he's already looking at the combinatorics. So you're basically looking at all these combinations of these vertices and edges to try to understand if there's any way to traverse it. So when you're looking at the seven bridges that is tractable. You can do it, you can actually enumerate all the options with seven bridges. Why is this relevant? It's relevant because you want to, sort of, optimize the construction of transportation networks and as you start to look at increasingly larger cities, larger bridge networks, and larger road networks it becomes increasingly more computationally expensive.

And a lot of these problems — I mean, one of the things that's exciting and challenging about graph theory at the same time is a lot of these problems they don't have polynomial time solutions; so they are in this combinatorial space, which I think is why people like graph theory. It's computationally challenging, so you look for approximation algorithms or you look to see if you can actually make progress on something [and see if] you can find a polynomial solution, which typically has been continuously proven to not exist or not be found.

Khari: Could you explain what you mean by a polynomial time solution?

Nadya: Maybe [we can] sort of come back — the other term that often comes up here is NP complete. If you are, for example, writing an algorithm and say your algorithm operates on X number of data points, a polynomial time algorithm is going to take X squared time, or X third time, or X fourth time. In the context here, it basically ends up being whatever the combinatorial complexity is. So you don't have an algorithm that allows you to basically take an exponent of your number of data points.

Khari: Ok. So your dissertation was on graphs; *Statistical Signal Processing for Graphs* is the title, I believe, and in that paper you are using large scale publication data sets to do some graph analysis. Could you explain sort of the background on that and what you found?

Nadya: Sure. I'd like to start with some of the early motivation for this work, and this is interesting because a lot of this research and the work that I've done in graph theory started back at Lincoln Laboratory. I was very much inspired by statistical signal processing theory of detecting things. There's a very formal set of methods to detect some sort of entity in context of a radar return, and in graph theory detecting patterns and graphs tends to be a lot more heuristic, especially large graphs.

So a lot of motivation for my Ph.D. dissertation, and a lot of the research that preceded this, was essentially how do you create these techniques to be able to detect patterns? [How are you] able to create distributions of signal and noise for graph theoretic data sets. So essentially what I was curious [about]...specifically for my dissertation the problem that I looked for is: is there a detectable signature of the emergence of a new academic field in a large publication dataset? And work included both the development of techniques themselves and applying them to this publication data.

So the techniques themselves essentially look at things like can you apply a filter — if you know what a pattern looks like can you apply it to a graph dataset and pull it out? And then the application, I actually worked with these publication data [sets] — I believe it was world of science data [*note: it was Thompson Reuters Web of Science*] — to see if you can actually identify those signatures.

I wrote my dissertation at ASU, so I was working with a number of interdisciplinary scholars who have studied [the] history of science. One of the things that's really hard in this space is truth data. So, how do you know that a new field emerged? Like, I actually think in terms of interdisciplinarity this is a really good example. Computer scientists often postulate that they have detected something in a data set, but it's not until you go to a historian, sociologist, or someone else that's a domain expert that they can actually validate that for you. So that was something exciting, I got to work with Manfred Laubichler here at ASU. He is a historian of science, and he had some examples in evolutionary biology and other biology types of fields where new fields were merged, so I was able to test if my mathematical techniques were working.

Khari: Ok. So in your PowerPoint about your dissertation, there is a line that says "results confirm that innovation leads to rewiring of networks." Could you explain what that means?

Nadya: Yeah. So if you think about...let's start with the type of networks that you can construct from publication data. Say you have a bunch of publications and one of the networks that is the most natural to construct is co-authorships networks. Essentially you make all the authors vertices — those are your entities — and then you make the papers that they co-authored together edges. If you and I write a paper together we're connected, if I also write a paper with Dan Lopresti, he and I would be connected, and you can build out networks out of that. So, when fields are changing, those patterns are fundamentally changing.

Say, you and I are both computer scientists and we are continuously publishing together, but if all of a sudden I started publishing with someone in supply chain that is going to create different sets of connections, and those were some of the things that we were able to validate and [were] able to pull out of these large data sets when we did the analysis.

Khari: Ok. Has there been any further analysis of publication data?

Nadya: Yeah. I know for sure that Manfred has continued to graduate Ph.D. students so...In my dissertation, I've actually identified a number of future directions and people have looked at it both from the perspective of techniques and from a perspective of applying it to broader data sets. The other thing that I would say is, generally this field is pretty active, so some of my colleagues at Lincoln Laboratory are still working in analysis of networks with these spectral graph theory techniques and matrix algebra techniques.

One of the people I will mention is Ben Miller. He's working on his Ph.D. at Northeastern at their Network Science Institute. So there's certainly continuation of this work, and I do want to be very clear that I've worked on this area probably for about 10 years or so. There's a lot of things that Ben and I started at Lincoln Laboratory and then we've had collaborations with Harvard and Patrick Wolfe. Then at some point it was very cathartic to write up a lot of this work and a dissertation.

[Deciding to Pursue a Ph.D. - 34:39]

I think one of the things that I mentioned is I don't have an exactly traditional academic trajectory, and sometimes people look at it and say,”Oh, you know, you did your Ph.D. in 18 months.” And I'm like, well, I did my Ph.D. in 18 months, but I did a lot of research in that 10 year career before that. So I think that's really important to acknowledge.

Khari: So can you kind of lay out your timeline? We may be glossed over this in the beginning, as far as from when you went to undergrad until you got to ASU.

Nadya: Sure. So, as I mentioned, at about six I decided I was going to be a mathematician. I know you said undergrad but we’re going to go back to six.

[Laughter]

So at six I decided I wanted to be a mathematician, at about ninth or 10th grade I decided that that was not necessarily a career that would allow me to pay off my future loans, so I decided to major in computer science. I wanted to go to a top CS program. At the time, as a high school senior, I remember I was deciding between Cornell and Carnegie Mellon, and I just fell in love with Cornell on my visit. I have the best memories of my time there.

At Cornell, I was a computer science major. About a little bit into my computer science degree, I realized that I had quite a bit of AP classes that I took in my high school back in Ohio, and what I actually was thinking of doing was getting a double degree. I wanted to do a double degree in computer science, which would be a B.S. through the engineering school, and a math degree through the college of liberal arts and sciences;

but when I went to talk to my undergrad advisor they actually gave me some really good advice, which, I don't remember this person, but that was probably really valuable.

He basically said, “Look, why do you want to do the dual degree? You're already taking a ton of math in your computer science curriculum, why don't you do a Master’s? You can do a Master’s in four years.” So I finished my undergrad in about 3 1/2 years and then I did a Master’s in just one more semester. When I graduated from Cornell with a Master’s I went to Lincoln Laboratory.

At Lincoln Laboratory I considered getting a Ph.D. because I had always wanted to go get a Ph.D., but I was having so much fun and I got to do research. You do the Ph.D. to do the research and I was getting to do the research, but then also having impact. So I was publishing a ton and I was doing very interesting research in a number of different areas. Eventually I got promoted to a level….I mean, I essentially broke the Ph.D. ceiling at Lincoln Laboratory, so there wasn't really career motivation for a Ph.D and I, intellectually, was very happy doing the work that I was doing.

Then in about 2012, my husband, who is a professor here at ASU, decided that we should go to ASU. He didn't decide, we discussed it, obviously...

[Laughter]

But it was something that he really wanted to do because he wanted to be a professor, that's been his lifelong dream and he had an opportunity to do it. He was a senior scientist at Lincoln as well. So he came to ASU, and once I came to ASU I realized that in an academic environment it's really helpful to have the Ph.D. credential, and I actually had my advisor, Manfred, approach me, and he said, “Why don't you have a Ph.D.? You have all these publications, why don't you have a Ph.D.?”

I said, “Well, look, I did the stuff that was the interesting thing. Do I really need this credential?”

He's like, “You should have a Ph.D.”

I said, “OK, can I get a Ph.D. in 18 months?” And I put out a plan because I'm a planner. I mapped it all out and I said, “If we can do it in 18 months and fulfill the requirements that are necessary, I will do it,” because at the time I was an assistant vice president, had a little kid, and a husband pursuing tenure. I was not willing to stop my job and obviously I needed to spend time with my kid.

So, that's how I got my Ph.D., but it was great because I got to write up that body of work. I mean, a dissertation is a really nice thing to have. There's something very meaningful about it, and to be honest, like, it's a neat thing to show my daughter. Like, “Look, I could do this and you could do this if you want to. You don't have to, but you could do this.” So that's kind of the trajectory and that was five years ago.

Khari: So do you think getting a Ph.D. has made a difference in your life and career? Would you give people that might be in a similar position as you the same advice?

Nadya: So, I think it depends right? For me, I think there were a number of dimensions of it that were deeply meaningful. I actually have a document that I wrote as a ninth grader that said that I was going to get a Ph.D., and not just a Ph.D., but a Ph.D. in math and my dissertation is actually in applied mathematics. So something about fulfilling that childhood thing was very meaningful even though I wasn't necessarily planning to do that if I would have stayed in Boston. I will also say that there is a degree of credibility that comes with a Ph.D which is helpful, particularly when you're working in kind of advanced R&D and S&T.

So I think it very much depends on the type of career you're in. In the career that I am in, absolutely I would recommend it. But there are many other places where you don't need a Ph.D. and certainly many studies will tell you not to get it. I think the other thing that I would say from an advice perspective [is], what helped me is that I knew why I was doing it. It was very clear, it was sort of like I wanted to basically cohere this body of work. It was relevant to the position that I held in the university, and it was, you know, to respond to that ninth grade Nadya. And on top of it I had a list of things I wanted to do research on. I think it is a lot tougher if it's not so clear.

[Outro - 41:23]

Khari: That's it for this episode of the podcast. We'll be back next week with part two of my interview with Dr Nadya Bliss. In that episode we discuss the work of Arizona State University's Global Security Initiative, how to combat the spread of misinformation, and the impact of sustainability on security. Until next week, peace.