Donna Dodson: What we'd like to do next is to begin a conversation with the community on priorities looking at some of the priorities as we move forward in the conference. And we all recognize the criteria for SHA-3 it needs to support digital signatures, key derivation, HMAC, random number generation, but are there other important applications that we didn't identify in the Federal Register?

Let me back up, maybe I missed a slide. Yes, well, it's also important to remember that in this SHA-3 competition we need to be working with cryptographers, but we also need to meet the requirements of security engineers, the protocol mechanics who will actually end up implementing SHA-3, developers, product implementers and standard developers.

So as we move forward in this process, we've gone from the first round where we've said to here are the minimum qualifications, here are the algorithms that met the minimum qualifications. Now, we're taking another look at this as we enter a new phase, where we're now starting to look at the selection for not any and all who met the qualifications, but again, looking for best of the breed, if you will. What does that mean in the case of SHA-3? Well, certainly first and foremost is security. So really, what we ought to say is here, put security above this question and say we recognize the importance of security that that's the critical requirement. But what other attributes are required for acceptance and use of SHA-3? In addition to the research and looking at the beauty of the math through the SHA-3 competition we have other goals. And, of course, one of our primary goals is the development of an algorithm that ends up in products, in protocols and in real world solutions. So that is a primary goal of NIST. And how do we get from where we are today to that end point? When NIST folks get together in our small group and we start looking at these, we think about application compatibility. We think about platform suitability. We're looking at things like efficiency, processing environment, power consumption.

We look at some of the application context. So one of the people that I need to give a lot of credit to is Tim Polk when we started thinking about this session, Tim Polk from NIST is one of the security directors for IETF. So he reminds us about the networking infrastructure components. And we think about embedded systems. We think about smart cards. We think about all of the personal networking devices that we're starting to see come online. Sensors, should the SHA-3 candidates be concerned about implementation in a sensor environment? What are the other environments that we need to consider?

We look at some of the performance and implementation tradeoffs, gate counts from a hardware standpoint, dependence of hash performance on hardware. Should we be concerned about algorithms that require multipliers, yes, or no? Can we make the assumption that hardware implementations have this? What about storage requirements? Some of the different candidate algorithms make some assumptions on storage requirements, is that a concern? Is that a feature? We look at the ability to parallelize things and parallelize some of these algorithms and we've seen a lot of discussion on that. How important are some of these environments? Are there other things that we need to be taking into context?

So what we'd like to do now is have a discussion with you to say are there other applications, some of the questions that we'd like to talk about in the next 35 minutes other applications from SHA-3 that should be considered? What are the application contexts that SHA-3 needs to support and why? What platforms are most important and why? What about application context and platforms are less important and why? Particular systems that are challenging for a multipurpose hash standard? So we think about this question, a lot of the work that some of the other folks at NIST are doing is, for example in the authentication area and looking at the use of small personal devices, smart cards, things like that, for authentication to a larger environment on the other side, say up in the cloud, something like that. Well, should we consider a hash on the smart card, I think that's something we'd all like to see, but is it practical at this point in time? And there you have the context of the card. You have the context of the protocol. You have the context of the larger environment. And lastly, we'd like to get your feedback on what you all think really are the critical attributes as we move forward to take a look at in addition to security. Again, let me emphasize that we think the security aspect is critical but as we move forward to additional phases of the competition, what are other criteria that we need to begin considering? And with that, I'd like to open it up to the floor for some discussion. Anybody want to go first?

Niels Ferguson: Hi. Niels Ferguson, Microsoft. I think one of your questions was where would this be used? And I would just like to say that standards get used everywhere. Hash functions are the nails of cryptography. They hold the world together. And there isn't almost no cryptographic systems that would not use a hash function of some sort. And it would be nice if that could be SHA-3.

Donna Dodson: So we hear that they need to be used everywhere, yes.

Xin Qiu: Qiu <off mike>. I think the migration <inaudible> is a big problem. Every time you migrated from an older system or structure to a new one there is a lot of changes everywhere. Tools need to be changed. Testing need to be changed. You can start from the unit test. So by the time this standard will become used you will be like people already long gone with the SHA-2 or with SHA-1 I'm sure probably still in the picture at the time. So how do you consider the new standard will help people ease in on their migration, on their adoption? And how do you make sure people can use some existing components that are already put in the system and to save the cost. I think you need to take the migration as a consideration of one of your criteria.

Donna Dodson: So you'd like to see as much backward compatibility as possible?

Xin Qiu: Not much backward compatibility. I'm not sure whether it's you will really want to do something very different than currently in the SHA-1 and SHA-2 family. But I'm just thinking the reality is by the time this is coming out a lot of people already have the SHA-1, SHA-2, AES in the implementation. If anything the preference-- I'm from Motorola, and the preference is if you can have something close to that and the change is from a hardware point of view, maybe even the structure point of view, don't need to be that dramatic, that definitely will be an easy adoption of this standard. But I don't have the answer for it.

Donna Dodson: Right. Neither do we. That's why we're asking some of the questions. And Niels, one quick question for you, do you see a lower bound on some of this or an upper bound that we at least need to say be in this range? I mean everywhere is a pretty broad statement.

Niels Ferguson: Yes, I had a conversation with someone who was doing battery operated sensors that had to run for three years and they were trying to save on microwatts and power. And there are always situations like that when they say, well we can't use the standard function, we're going to do something ourselves. But everybody would like to use the standard function, so if they can they will. And it's kind of undesirable if they have to go to specially designed functions for that particular problem space. And these people are doing it. They're designing special crypto and they basically design a gate layout and say this is only 100 gates and it takes so many nano joules to compute and we can afford that. When you get down to that level no standard can help you any more. But as far as possible, we should be able to use the same function.

Donna Dodson: So when you look at the potential life of SHA-3 what's sort of in your mind the lower practical bound to consider?

Niels Ferguson: Certainly smart cards, very important. Most smart cards today have some kind of hash function MD-5 or SHA-1. So that's certainly-- there's a billion smart cards out there and that's an important platform and it will remain an important platform.

Donna Dodson: And upper bound?

Niels Ferguson: Whatever Intel gives us.

Donna Dodson: Okay.

Charanjit Jutla: While it is true that hash functions are used everywhere, but I think one application which NIST must keep in mind is digital signatures. I mean that is the key application for which security of hash functions is required. Now, of course you could say that your private RSA key is protected under passwords and for that itself you may need some hash function, that is true. But I mean this one application is kind of a key driver. And I'm not saying that others are not important. In fact, once you build the hash function you might as well have it working for everything else, that's true.

Donna Dodson: Thank you.

Carmi Gressel: I think very important in smart cards and other applications is loading encrypted data and authenticating it transparently and fast. In other words, booting a secure-- an application from a non-secured memory. In fact, we felt that the engine should be put on these memory sticks, for instance.

Donna Dodson: I think that corresponds to what Niels was saying about smart cards as being a driver in terms of small handheld …

Carmi Gressel: Of course, it has to be low power consumption. We also believe that we did that. But the idea that you're receiving encrypted data, you want to make sure also that it's authenticated.

Donna Dodson: Bruce.

Bruce Schneier: I want to make sure we mention that short messages are very important. I mean we like think of hashing large parts of data but whether it's small Internet packets or small protocol messages that when we look at speed it's not just the flat out run rate but the start up speed looking at short or very short messages.

Peter Schmidt-Nielsen: One thing that I'd like to point out that is in terms of getting people to start using SHA-3 quickly, if we have a fast software implementation then people can immediately switch over. After all, AES has been around for quite a while, and we're only just now getting Intel instructions for it. So if you have a really fast hardware implementation then you may be able to get speed advantages later on; but if we want a lot of people to switch over to SHA-3 quickly, then a fast software implementation allows people to almost instantly switch over to an efficient algorithm

Donna Dodson: In the back.

Danilo Gligoroski: Many people already mentioned that the hash functions are used everywhere but NIST in their slides or consideration should think about what is coming on in one field which is called digital forensics. And there the hash functions will be used in hundreds of ways. But as a trend, and you as a standardization body, should be aware that field is really fast developing as our society is going to a digital society.

Donna Dodson: Okay, up in the back.

Jason Martin: So I have a question that's more for the experts that may be in the audience, I keep hearing the term smart card or RFID card thrown around. And when I look at the standards, there seem to be lots of different types of smart cards and RFID tags. And particularly with RFID tags they're the tags that are more like shipping pallet tags versus the tags that go in passports and they have different capabilities. So could we have like a more concrete idea of which particular of these resource limited devices are we talking about when we say smart card or RFID tag? Because it makes a big difference as to what's practical and what's not.

Dan Brown: Yes, I would just like to comment that for message authentication you have GMAC and CMAC already. So that might be examples where you might not need a hash.

Donna Dodson: Okay. Adi.

Adi Shamir: I would say that the main goal of the SHA-3 competition is to make sure that there will not be a SHA-4 competition within the next 20 or 30 years. Now, in order to achieve it, I believe that it's important to choose one algorithm but to keep the parameters of that algorithm as flexible as possible. So unlike the case of AES in which everything is custom concrete and all of the parameters are fixed, I think there is room in the choice of SHA-3 to leave some parameters free and especially to make it possible to choose them in such a long way that for the processors that will be available in 20 years when this will not cause any big overhead we will be able to use the same algorithm with great enhanced security. And one final comment, there are modes of operations for hash functions. I don't have time to get into them but just to throw one idea, you might be able to take the message and in many cases the message has to be hashed once at the source, and then checked a large number of times by other processors. So the time to perform the initial hash is less important than the time to verify the hash. Now, one idea how to utilize this feature is, for example, to take the message and hash it with multiple choices of the parameters. So if you have to hash it only once, for example, an operating system update, so Microsoft is going to hash it with parameters at one, two, three, five, up to ten. And then anyone who wants to check it can decide. If it is a weak processor it will check it only the part which had been hashed which had been concatenated with parameter level five. And if it's more powerful it will check it all the way with parameter level ten. So to have mode of operation which can make it much more flexible then we are used to with encryption algorithms.

Donna Dodson: So up here in the balcony.

Man 10: Well, since we're discussing different implementation and different applications of hash functions, I would like to point out that most of the applications that were mentioned now are one time hashing applications like you're checking the digital signature authentication. How often does the authentication occur, just once per session? What is the application where hashing is used all of the time that engages the processor continuously? That is distributed file systems.

And I see a small number but I see them in development right now widely distributed file systems on untrusted networks using file sharing and with the development of the Internet it seems to be a very, very interesting application for the future. I would certainly prefer my files to be widely distributed so I could access them from anywhere. And hashing is performed there continuously.

So not only speeds matter in this regard, I'm surprised to hear from the developers that they are especially interested in hashes that fit on 8-bit processors that is that can hash with less than 256 bytes of RAM. So they really want it to be secure, of course, with such a widely distributed network, they also need a wide range of implementations from SSE and 64 bit processors down to 8-bit smart cards. So I mean I also see a hash function used in login pages as in Java script. So the ranges of course very large, but that is I see the most interesting application to pay attention to where hashing is continuous.

And also I'd like to point out that the hashes are also used continuously by a computer, by the antivirus software, hashing every byte of your memory and every application that is running. However, they still use MD5 because simply they do not care about its collision resistance. And I mean I would prefer if it changed in the future with a good submission that is flexible enough to address these implications, that we're really going to be hurt by if the hash is too large to fit on the Web page, or too slow for such applications, or simply do not fit on a smart card. Look what happened to the Swedish airplanes that could not use the standard. So they do not encrypt their communications, the military air jets. So the same thing happens to the smart card and RFID applications if they cannot fit a standard algorithm they do not use cryptography at all. So we all suffer in the end with having insecure applications. So I mean I would really stress the fact that the SHA-3 must be sufficiently small for the small devices so that we will not have the next MIFARE fiascos affecting billions.

Donna Dodson: Thank you.

Man 11: In my opinion, if we take let's say the two extremes of the applications that are like low power, like mobile phones or ultra low cost device like sensor, or on the other side various speed network caravan [ph?], these devices we use mainly are the implementation of hash, and if we look back at SHA-1 and SHA-2 for an algorithm designer there is almost no space for building a very fast or a very low cost hash engine. So I would say that algorithm scalability is a big issue for this kind of hash function.

Donna Dodson: Do you have any bounds that you can put on some of that so that we get a feeling like when you talk about low power, what that means to you?

Man 11: Well, for low power I don't have number, but if we say low cost it's usually like a few kilo gates, and it's not very important let's say the amount of cycle that you compute the hash, while if we look at the trend in network, I think that after gigabit Internet, there will be 10 gigabit Internet and feed [ph?] like that. And this will not be in keeping with an Intel PC for doing the hash.

Donna Dodson: Thank you. Bruce.

Bruce Schneier: I think if there's any moral over the last couple of decades of computing is that the high end always gets better and the low end never goes away; it just gets smaller and lower power and cheaper. So when you think about what are the bounds, I think you're better off thinking in terms of the footprint of the hash function, that it's got to work at the high end, the low end, low power, low memory, high memory, low gate count, high gate count and fast, and the bigger footprint the hash function we choose can fill, the more it will be used. So really it is the broadest we can do it without sacrificing other things; I mean there are going to be tradeoffs at the edges, but the low end never goes away, it just gets in smaller and cheaper and lower powered devices.

Donna Dodson: Are there general comments?

Xin Qiu: I want to make a comment about somebody made the fastest SHA software implementation will get at the adoption of SHA-3 easier. I somewhat disagree with that. I think currently the performance bottleneck in the software I'm not sure is that insignificant. So anyway by that time the SHA-2 is widely used I suppose, and I think the consideration really should more focus on the hardware adoption side. If you think back, from a hardware point of view, like people using Intel as example now of the 10 years AES as a standard Intel is putting fast implementation into their processor. So from a hardware vendor point of view, I think maybe something if it's really like AES like or SHA-2, SHA-1-like, then the change will be much faster and much easier. Because the software change relatively speaking is much easier than hardware change. And also if you reuse AES and the SHA-1, SHA-2 components, I see a lot of variance, a lot of proposals that are totally different and I'm not familiar with those constructions, so if you use something really new, there's not much reuse that can come out from the current implementation; if it's AES-like, Intel already have very fast instruction coming up, so building on top of that, that would be much faster for SHA-3 got used. So I think AES-based SHA-3 will be very attractive.

Donna Dodson: Other? Niels?

Niels Ferguson: The previous speaker raises an interesting thing. By the time SHA-3 comes out like five years later when people actually start thinking about adopting it, a lot of people will be using SHA-2, and if there isn't anything where you say SHA-3 is better than SHA-2 now, I'm afraid adoption is going to be very slow, and that might kind of defeat the whole point of doing this process.

Donna Dodson: Okay. Dan.

Dan Bernstein: I'd like to return to a previous comment about GMAC and CMAC and in general whether there's any use of HMAC. It seems for people putting together authenticated encryption, we now have very fast, I mean faster than even HMAC MD5, and very trustworthy, as secure as AES, authenticated encryption systems, so what's the use in HMAC? Is there anybody who can explain why we still need HMAC? Why HMAC SHA-3 is a valuable application area?

Donna Dodson: Up here.

Christian Wenzel-Benner: We've heard a lot of comments about the lower end smart card small devices, and everyone seems to have a different understanding depending on their background, whether they're doing wireless, mobile phones, smart cards, whatever. It might be a good idea to just define three typical low end setups, whatever typical means, and then have the people and companies involved here vote on what is the smallest platform that you personally think you will want to use the standard on. So that way we can maybe get a feel for the lower bound of the footprint that the standard is going to cater to.

Donna Dodson: So if people want to comment on what those three typical lower bounds should be, I think today or in some of the forums some of this is, starting to answer some of these questions is very appropriate.

Man 14: Mine is just a pretty similar question but related to SHA-2. What is the lower bound that people are using SHA-2 at this point in time so that we have some idea? I think we need some feedback from industry to have some idea where SHA-2 is failing us and that will help.

Donna Dodson: Obviously, I'm not in industry, but I don't think SHA-2 is used all that much to date to kind of take a look at the lower bound on that.

Niels Ferguson: So to answer Dan's question why you would use HMAC, if you are on small chips and you need a hash function anyway for some part of your protocol, then you're using that for your authentication it's just really nice, that's one of the reasons that you would use that. As to where SHA-2 is used, I haven't seen it used that much, but trying to start using it on the PCs now, but I actually haven't seen any small …

Peter Schmidt-Nielsen: Can you put the microphone closer?

Niels Ferguson: Sorry.

Peter Schmidt-Nielsen: Thank you.

Niels Ferguson: I haven't seen any SHA-2 use on the really small devices yet, but other people might have more experience there.

Donna Dodson: Ron.

Ron Rivest: The discussion about low end devices is an interesting one and I think if you're trying to develop levels, I think, there's two levels of interest that I see. One is where you've got a situation where you're doing digital signatures, so one class of devices are those that are using hashes because they're using digital signatures, and there you've already got a fairly beefy device if you're going to be doing public key digital signatures. And then below that, one would have devices which are not doing digital signatures but maybe doing some form of authentication so using some sort of authentication code, and the arguments there about using CBC MAC or some other form of authenticated encryption are fairly compelling too. This is a hash function design contest, not an authentication code design contest. And at the very low end below where you're doing public keys I think authentication is probably the only real application that I see. I don't see a real need for collision resistance below that, below the level of digital signature as much. So when one asks what is it you're doing? And I think these two levels of small devices that are doing public key anyway and then small devices below that. And below that, if you can identify applications that are different than just merely message authentication I'd like to know about them.

Donna Dodson: Yes.

Dan Brown: NIST has set a deadline of 2010 for the 80-bit security level and SHA-3 is going to be decided upon in 2012 so that's a two-year window where you'll be forced to use SHA-2 if you want to go higher than-- you have to use SHA-2 for digital signatures, at least. So that will force people to use SHA-2 and SHA-2 will get its foot in even if it's not there today. So how do we get SHA-3 to take over from there? I don't know.

Donna Dodson: I think we've heard from folks in the audience and I would claim some of our thinking is some of these attributes that we're looking at in SHA-3 need to look better than what we have with SHA-2.

Xin Qiu: I think Microsoft just said they don't know if SHA-2 is really used. Motorola is already putting the SHA-2 for the digital signature for signing code into our phones. Also, I'm not sure if people are aware WiMAX that's one of the standards. Right now it's pretty popular. It's specified SHA-2 to be used in all of their PKIs, their certificates. So it will be in the various devices. So SHA-2 will be widely used.

Donna Dodson: Adi.

Adi Shamir: There is one issue which is related to the way we run this competition, mainly any kind of attack which is better than the earthly [ph?] bound or inversion bound is considered an attack on the hash function. And this leads to designs and to choices of parameters by the designers which give you security of say 2^{256}. Now, it's okay to play this game in order to level the playing field for all the participants, and in order to try to find the possible attacks, but in reality no one needs 2^{256} security. And as a result, I would recommend that the designers should give a design which should withstand all possible attacks, recommend particular parameters that guarantee it, but also give some kind of analysis or supporting evidence or something what is actually needed in their design to get realistic levels of security like 2^{100}. So if they use it then, maybe not to be an official parameter which will be recommended by NIST, but at least something that, unless disputed by the research community and shown to be wrong, people will be able to use on their own risk, for example, in order to speed up the implementation on smaller devices. Right now, we are recommending the level of security which is way too high and only in order to win the competition.

Donna Dodson: In the back.

Man 17: I'd like to come back to the previous question about small devices. I think three obvious reference platforms for small devices would be like the [Intel] 8051, Atmel AVR and the ARM7TDMI. So as far as I know, all of these three devices are used in wireless centers, C-states [ph?]. I've seen research projects by various universities that have used one out of these three,

and while I don't know of any other usages besides HMAC in these scenarios, but that was one of the applications I've seen.

Donna Dodson: Thank you. Yes.

Christian Wenzel-Benner: Maybe I should clarify on the intent of my comment. I was more thinking in terms of memory than of the CPU use. These three are the obvious embedded platforms in use today, but my question was more like - can everyone live with 16K of program space and 2K of RAM, or do we need to go down to 8K and 1K, or can we go up to 64K and 4K of RAM respectively. I think that will be the critical parameter for many applications, whereas the time it takes to actually compute the hash, I personally think is secondary but that, of course, depends on the viewpoint.

Donna Dodson: Just a second, in the back.

Man 18: Yes, I would to respond to I heard it mentioned that someone never seen a SHA-1 or SHA-2 implemented on an RFID chip or a small microchip, there's a reason for that they just don't fit. They're just way too big. The code base is insanely large. The area size is just too big. And Bruce Schneier mentioned correctly, those chips will never go away. In fact, they spread. They get wider and wider used for authentication and not just authentication for access control to buildings. They're used by the military for access control. They're used in our car keys. And when the industry cannot implement a standard they either invent something themselves, which is very rare, but most often they just don't use anything at all. They do not encrypt. They do not hash and we suffer. Billions of people suffer. And it will just get worse and worse. So we really should consider those very small microchips and I mean like 1,000 gates of available area plus CMOS memory available area for the hash function. I know EnRUPT would fit. CubeHash would fit in that. So there are submissions that can actually satisfy those applications. And we will see those hashes in those small long distance RFID chips and small smart cards.

Donna Dodson: Niels.

Niels Ferguson: I'd like to answer Ron's question why you would use a hash function. You use it for things like key derivation, random number generation, generating unique identifiers; the crypto protocols are full of hash functions everywhere, and they just get used for everything, a bit like duct tape.

Man 19: You mean why would people use a hash function in an RFID chip for access control?

Niels Ferguson: No. Ron was saying why would you use it if it wasn't for digital signatures or authentication, and my experience is you always end up using hash function somewhere.

Man 19: Yes, that's true.

Donna Dodson: Carmi.

Carmi Gressel: I'd like to disagree with Adi Shamir. If we're talking about long life and we have to think about the future methods that would include quantum computing and in other words limit the number of-- by limiting the number of monomials to simplify is the antithesis of looking for long life if quantum computing is going to be our next enemy.

Donna Dodson: Rich.

Man 20: One general question about the procedure. So in many slides of NIST that have shown questions, so do you plan to have a questionnaire or a poll that people can fill out?

Donna Dodson: Are we planning to poll people?

Man 20: Do we plan to have a paper form that people can fill out or maybe a form on your web site?

Donna Dodson: Certainly, we can take advantage of some of the online forums for people to communicate those answers. For NIST to actually do an official survey requires that we get approval from the U.S. government and there's a lot of red tape involved in that. I believe during the AES competition we might have gotten some permission to do a little of raising of hands. And I know Bill's used humming in the past, like at IETF for acceptance of certain things because of our strict requirements on surveys.

Man 20: Maybe you can kind of unofficially ask IACR, IACR can do it on your behalf?

Donna Dodson: Well, certainly, we'll have the beginning of a list of questions that we came up with. I think there are other questions that could easily be added by today's conversations and be thinking about your responses and your answers because we are interested. Someone has the mike? Yes, Rich.

Rich Schroepel: I was thinking that it's a little odd that most of hash functions have been designed to operate on 32 and 64-bit computer with gigahertz clocks. And yet, we're mostly talking about machines that wouldn't even qualify as computers today as a selection criteria. It seems it's probably more important to think about the mainframes and our desktops and things that have a lot of power, than to worry too much about the fringes. Now, I know the fringes have some relevance, but we haven't even actually specified the fringes very carefully. And saying that those are going to be our selection criteria, primary selection criteria is probably a mistake. And we've got to put security first, and then we've got to ask what are our major applications. It may be that the people with the little tiny computers have to use SHA-1 or MD5.

Donna Dodson: I think we have time for just a couple of comments, Ron.

Ron Rivest: Thanks. So I wanted to continue the dialogue with Niels. So at the low end you're talking-- I'm familiar as you are with many of these protocols and how often hashes get used, but if you look at them carefully, you're in situations where you either have digital signatures or you've got some other encryption algorithm you're doing a key derivation for an encryption algorithm or something. So we've already got an encrypted AES implemented on the low end device you're going to want to use that for your authentication algorithm or whatever to your key derivation, so the places where you have only a hash function are very hard to come up with. The second point I wanted to make about low end devices, and this fringe category is low end devices and I think that's a good term for them, is that those systems are by and large proprietary systems too. One of the points of the standard is interoperability. And you get those with digital signatures, you get those certificates and so on too. Almost always a low end system is such cost constraints that they are not interoperable with the universe at large and the Internet at large, they're mostly a closed system. That may change over time but at present that's what they look like.

Donna Dodson: Okay. We have time for two more comments.

Nicky Mouha: Okay. I think tunable parameters are a good idea and can make a very flexible algorithm, but the tunable parameter shouldn't be part of the standard because the main purpose of the standards, according to me, should be interoperability. And interoperability is seriously hampered, I think, if everyone just picks a different set of parameters in this space.

Donna Dodson: Bruce.

Bruce Schneier: Yes, we've heard comments about <inaudible>, but when I see a standard, its interoperability, I think, is less important than it's a uniform security level. And when you see these proprietary systems, and I see them a lot and I'm certain a lot of other people do, when they do something themselves, the problem is not that they're not interoperable, the problem is they're doing something lousy. And that as we the community are going to give them a hash function, we want to give them something that it will be good and that they'll use, but they won't be tempted to roll their own and produce a lousy security as a result. So I think it's less interoperability and more uniformity.

Donna Dodson: Okay, unfortunately, we're running out of time, but I want to close with a couple of thoughts. First of all, we have some additional panels from the NIST perspective that are meant to be conversations that follow this same thread as the week moves on. Second of all, I want to thank you all for your thoughtfulness. And I'd like to see some of this continue on some of the different forums from my perspective. Third, I want to echo something that Dan Bernstein said in his briefing, I believe it was yours, and we thank the folks that are doing cryptanalysis, and that is critical for us to move forward in this. And as we begin to select and narrow down the range of things, I'm reminded of the great work that the Brian Gladmans of the world did for the AES competition, where he looked at some of these different platforms, for those of you who go back that far. And I want to encourage the community again to provide us with that kind of feedback as well as the cryptanalysis. Both of those are very critical to NIST, and we consider all

of that input very heavily as we move forward into these next phases, because again we're kind of switching modes, instead of what made the minimum criteria in the beginning, now we're starting to look for best of the breed in some of these. So I want to thank all of you again for your comments today, and I hope this conversation continues. Last is just an administrative note, we get back here at 1:30 today rather than the 2 o'clock that we have been coming back for lunch. Thanks so much.

End of Session4.mp3