Workshop Report
The First Cryptographic Hash Workshop

Gaithersburg, MD
Oct. 31-Nov. 1, 2005

Report prepared by
Shu-jen Chang and Morris Dworkin

Information Technology Laboratory,
National Institute of Standards and Technology,
Gaithersburg, MD 20899

Available online:
http://csrc.nist.gov/groups/ST/hash/documents/HashWshop_2005_Report.pdf

1. Introduction

On Oct. 31-Nov. 1, 2005, 180 members of the global cryptographic community gathered in Gaithersburg, MD to attend the first Cryptographic Hash Workshop. The workshop was organized in response to a recent attack on the NIST-approved Secure Hash Algorithm SHA-1. The purpose of the workshop was to discuss this attack, assess the status of other NIST-approved hash algorithms, and discuss possible near-and long-term options.

2. Workshop Program

The workshop program consisted of two days of presentations of papers that were submitted to the workshop and panel discussion sessions that NIST organized. The main topics for the discussions included the SHA-1 and SHA-2 hash functions, future research of hash functions, and NIST's strategy. The program is available at http://csrc.nist.gov/groups/ST/hash/first_workshop.html.

This report only briefly summarizes the presentations, because the above web site for the program includes links to the speakers' slides and papers. The main ideas of the discussion sessions, however, are described in considerable detail; in fact, the panelists and attendees are often paraphrased very closely, to minimize misinterpretation. Their statements are not necessarily presented in the order that they occurred, in order to organize them according to NIST's questions for each session.

3. Session Summary

Day 1 Keynote Speech: Cryptanalysis of SHA-1 Hash Function

Xiaoyun Wang of Tsinghua University delivered the keynote speech for the first day of the workshop. The cryptanalytic research performed by Dr Wang and her colleagues on SHA-1 was the catalyst for NIST to organize the workshop. The research results, published in the proceedings of the Crypto 2005 conference, describe a differential analysis of SHA-1 that should enable a collision search to succeed after an estimated 2^{69} hash function operations. In her keynote speech, she presented improvements in her results on SHA-1 in which the estimated operations for the search were reduced to 2^{63}. These results had been announced on her behalf at Crypto 2005. The methodology for both results was similar, building a map for a two-block collision of the compression function, based on a single differential path for a near-collision of the compression function. Sophisticated message modification techniques were applied to achieve the necessary conditions on the chaining variables and the message bits. The differential path for the improved results was very closely related to the previous differential path, beginning and ending two steps earlier.

Session 1: Papers - Hash Collisions: Impacts and Workarounds

Session 1 consisted of five talks on the practical impact of hash function collisions, including two proposals for strengthening collision resistance that often could be implemented at the protocol level. Steve Bellovin of Columbia University began the session with a talk about the technical issues involved in replacing/upgrading the hash function within cryptographic protocols. His work had focused on S/MIME, TLS, IPSec/IKE/IKEv2, all of which, he concluded, would require more work to prepare for and manage any transition to new hash functions.

George Illies of the Bundesamt für Sicherheit in der Informationstechnik (BSI) (the Federal Office for Security in Information Technology, in Germany) explained how he and his colleagues had extended the recent results of Daum and Lucks on signatures of PostScript files to other file formats, such as PDF, TIFF, and MSWord97. In each case, an apparently meaningless collision of the underlying Merkle-Damgård hash function could be leveraged to create two distinct, meaningful documents whose signatures were identical.

Hugo Krawczyk of IBM T.J. Watson Research Center proposed a mode of operation for existing and future hash functions, which he called randomized hashing. The signer would incorporate a random salt value into the hash function input and transmit the salt with the signature for verification. He asserted that the resulting increase in security against collision attacks was worth the changes to signature standards in the encoding and processing of data.

John Kelsey of NIST explained how, given sufficiently many precomputed hash collisions, it is possible to "herd" an arbitrary prefix string to a predetermined hash value by appending an appropriate suffix. This attack illustrates that digital signing is not the only use of Merkle-Damgård hash functions that relies on collision resistance: collisions are also relevant to various types of commitment schemes, where the hash function is used to demonstrate prior knowledge of an input value.

In the final talk of the session, Michael Szydlo of RSA Security proposed two types of message preprocessing, namely, word-wise message whitening and message interleaving, to prevent the known collision attacks on SHA-1 and MD5. Either measure would be a viable solution to increasing the secure life of the function.

Session 2: Panel Discussion - SHA-1: Practical Security Implications of Continued Use

This panel was chaired by Donna Dodson of NIST. The panelists included Steven Bellovin of Columbia University, James Randall of RSA Security, Hugo Krawczyk of IBM T.J. Watson Research Center, Georg Illies of Bundesamt für Sicherheit in der Informationstechnik, and Niels Ferguson of Microsoft.

This panel and the audience addressed the following issues related to SHA-1:

1. Where is SHA-1 widely deployed today?

   Hash functions are used in many places, such as in message authentication (e.g., HMACs), random number generation, key derivation, and digital signatures.

2. Which applications are threatened by collision attacks and which are not?

   Krawczyk stated that the application that is most affected is digital signatures with 3rd party verifiability, where a 3rd party verifies the signature, and there are legal consequences in verifying those signatures. An application, such as authentication, where a challenge (or nonce) is sent to a party who signs the challenge himself, and there are no consequences for it to another party, is not affected by a collision attack.

   Randall stated that message authentication, random number generation and key derivation are not affected by collision attacks. However, digital signatures, where the signer does not have control of the message that he is signing, are affected by collision attacks.

3. What are the security implications of current attacks on these applications? Is it practical to continue using SHA-1 for the next five years?

   Ferguson stated that the current use of SHA-1 is not a big problem, as a work factor of 2^{63} operations is still not a practical attack for most people. However, the distinction between collision resistance and pre-image resistance is an academic distinction. He is very much against the idea of categorizing the applications of hash functions, and determining whether it's safe to use SHA-1 for certain applications, but not for others. Hash functions are widely used for many applications, each with its own threat model. Conducting a security analysis for each application would take a lot of time, expertise, and effort; it would be better to spend the resources working on the next generation of hash functions.

   Other panelists disagreed. Bellovin stated that for applications that do not rely on collision resistance, such as HMAC or key derivation, he wouldn't mind using SHA-1, especially when the performance impact of SHA-256 compared with SHA-1 is significant.

   Krawczyk commented that, in addition to HMAC, protocols such as IKE (the Internet Key Exchange) can continue to use SHA-1 because the threat model is not collision resistance, but pre-image resistance. He cautioned against banning SHA-1 outright for any application, especially when we really do not have a better alternative.

4. Under what circumstances, would an immediate shift away from SHA-1 be required?

   Most panelists agreed that shifting away from SHA-1 was not urgent, and, absent an actual collision, it may be fine to continue using SHA-1 for applications that do not require collision resistance. Some suggested that if the estimated attack effort drops from 2^{63} to 2^{40} operations, or if someone finds a pre-image attack or publishes a SHA-1 collision, then immediate action would be required. Nonetheless, most panelists agreed that we should start planning for a new hash function now.

Kelsey commented that people had a few years of warning about the weakness of MD-5, but did not take it seriously until there was a practical attack on MD-5. So if we wait until we get a real attack, such as a preimage attack on SHA-1, it may be a little too late to shift away from using SHA-1, especially when a changeover tends to take a long time.

5. What are the benefits and costs of shifting to another algorithm in the next few years?

   Bellovin stated that for the IETF, it takes one year to develop a new standard; and every time an algorithm is changed, it takes about 5 to 7 years, at a minimum, to be able to interoperate with other implementations.

   One attendee spoke from the hardware industry's perspective. She commented that changing algorithms on a hardware platform is difficult and costly; therefore, if we know that it's safe to use SHA-256, it's fine to switch to SHA-256. However, if we are not sure at this time about whether SHA-256 is secure, and if current attacks on SHA-1 are just theoretic attacks, then we should not panic, we should wait to transition until we know more.

   Another attendee commented that getting vendors to support a new algorithm is the easy part, but getting users to change is the hard part. Users don't like to touch working systems, and that's something we will always face. So for new systems, nobody should use SHA-1 if one can avoid it.

   There was a question on applying countermeasures to block the collision attacks, Ferguson answered that whatever we switch to, patches do not work well in the practical sense. It would be better to design a new algorithm with an emergency double-the-round mode.

6. What should be done to help ensure a smooth transition?

   Bellovin suggested a more aggressive schedule for standardizing new algorithms. He argued that if people are aware of the plan in advance, they can plan ahead and adapt better.

   Krawczyk commented that, considering the cost and time to develop a new standard and deploy a new algorithm, we should do something to make it easier for the next time to upgrade. That means that there needs to be a lot of thinking and planning, including having a variable number of rounds, or parameters in the algorithm that allow a switch to a stronger version.

7. What kinds of guidance should we give to implementers?

   Ferguson advised implementers not to bother implementing SHA-1 for any new application, but to go directly to SHA-256 and make sure that their system is cryptographically agile. Most of the panelists agreed.

8. Would it be practical for NIST to issue a policy statement forbidding the use of SHA-1 after 2010?

Ferguson stated that getting rid of SHA-1 and MD-5 is a nice goal, but it can't really be done because of backward compatibility requirements. But, new applications should not use SHA-1, and people should not generate new data with old algorithms. People should negotiate using strong protocols so that the protocols would not automatically select weaker cryptographic algorithms.

Session 3: Papers - Status of SHA Family Hash Functions

Session 3 consisted of four talks related to the status of the SHA family of hash functions. The first two talks of the session were given by Christian Rechberger of the Institute for Applied Information Processing and Communications (IAIK) of the Graz University of Technology (TU-Graz). In his first talk, Rechberger presented research on the rotation constants within the step function and the message expansion of SHA-1. The research examined the effect of varying these constants, as well as the number of steps in the compression function, on a certain security metric. In his second talk, he presented results on simplified variants of the SHA-256 message expansion.

Hirotaka Yoshida of Hitachi, Ltd presented research on the application of neutral bit techniques to various SHA-like functions, including a linearized variant of SHA-256. John Kelsey of NIST proposed a truncation mode for SHA-256 and called for feedback on it.

Day 2 Keynote Speech: Design Principles for Hash Functions

Bart Preneel of Katholieke Universiteit Leuven delivered the keynote speech for the second day of the workshop. The topic was a survey of the design principles for hash functions. He first discussed the gaps between theory and practice in defining desirable properties of hash functions and provided a survey of a variety of generic attacks, constructions and related results in the theory of hash functions. He gave a series of recommendations for improving the Merkle-Damgård design paradigm and presented some new research on the pseudorandomness of the HMAC construction. Preneel concluded that the security of hash functions is not very well understood, especially in theory, while~~,~~ in practice, designers have repeatedly been too optimistic in choosing the number of steps in compression functions. He observed that there are several important areas of research to pursue, including the design of strong compression functions, and the understanding of various security properties in addition to collision resistance.

Session 4: Panel Discussion - SHA-256: A Suitable Replacement for SHA-1?

This panel was chaired by John Kelsey of NIST. The panelists included Orr Dunkelman of Technion, Antoine Joux of PRISM Laboratory, University of Versailles, Christian Rechberger of IAIK, TU-Graz, and Hirotaka Yoshida of Hitachi, Ltd.

This session was intended to address the following issues related to SHA-256:

1. What is the best current attack on SHA-256? How much confidence can we have in the security of SHA-256, considering its pedigree, whether or not the current attacks have been fully applied, and whether there are better attacks on the horizon?

Dunkelman estimated that we probably can break 40 rounds of SHA-256 with current techniques, and with some modifications and tricks here and there, we probably can break 45 rounds of SHA-256. To get beyond that, we will need new techniques. But, he suspected that better techniques are on the horizon, and new techniques will come in the next few years.

Rechberger stated that Wang's attack was not that straightforward, and even assuming that Wang's attack on SHA-1 could be applied to SHA-256, without new tools, it would be difficult to speculate on the effectiveness of the SHA-256 attacks.

Joux stated that current attacks have not been fully applied to SHA-256; analysts were still focusing on SHA-1 as a more promising target. He emphasized, however, that for attacks on SHA-256, we need something new. He did not think that it is easy to attack even 40 rounds of SHA-256, as Dunkelman had suggested.

Kelsey stated that there is no proof at the moment to indicate whether the complexity of SHA-2's message expansion makes it a stronger or weaker hash function than SHA-1. He wondered if there is some small variation that one can do, such as Jutla's technique for the SHA-1 message expansion (presented in Session 6), that can be used on SHA-256 and give us more confidence in its strength.

Rechberger answered that the techniques that Jutla used to prove his lower bound (on the Hamming weight of the disturbance vector in the expanded message) for SHA-1 would be difficult to apply to the more complicated SHA-256, even if its message expansion were linearized. Moreover, a proof for such a variation of SHA-256 would be of less value, because it would be difficult to find a rule that says anything about the complexity of the associated collision search.

2. How might we get more confidence in SHA-256?

   Yoshida stated that we need an updated security report on SHA-256; we need someone to apply all the known attacks on SHA-256, and to publish a security report. Once that is done, cryptographers can analyze the results and see if they can improve these attacks. Other panelists agreed that we need to learn more about SHA-256.

   In addition to cryptanalysis, by applying known attacks to SHA-256, Joux suggested that we could also look at the design principles and criteria of SHA-256.

   One attendee commented that if we could have the design and evaluation report of SHA-256, it would help to gain confidence in SHA-256.

3. How likely is SHA-256 to resist attacks for the next ten years?

   Dunkelman speculated that with current attack techniques, combined with other ideas, SHA-256 is likely to be broken in the next five to ten years. Others felt that, while theoretical attacks on SHA-256 are likely, real practical attacks may not be likely.

   Joux pointed out that the message expansion in SHA-256 is more complex than in SHA-1 and is not linear; he felt that new techniques are needed in order to break SHA-256, and finding such new techniques would be a research challenge. He also pointed out that even if academic attacks weaken SHA-256, the 256/128 bits of security is still a huge number for the second-preimage/collision attacks on SHA-256; he thought that the cryptographic community has a lot of work ahead to find practical attacks.

Other issues that were raised by the audience were:

4. Would SHA-256 be chosen in an AES-like competition?

   Joux commented that SHA-256 probably would make it to the second round, but might not be the ultimate winner. A variety of attributes were considered in the AES competition, and SHA-256 would have to accommodate more hardware platforms, for example, in order to be the final winner.

5. Should there be multiple hash functions?

   One attendee asked for multiple hash function standards, as “algorithm agility is important”. This viewpoint was echoed by some, but other attendees pointed out that algorithm agility is not easily attained in hardware, since it’s really hard to change hardware, so they argued for over-design of a single hash function.

   One attendee asked the panelists, “If you have to make a recommendation to your company next week, which two hash functions would you recommend using?” One panelist answered, “Take a standard compliant function like SHA-256, and a non-standard compliant one like Tiger.” Someone in the audience suggested Whirlpool.

   One panelist stated that if standards compliance is an issue when developing current applications, then SHA-256 and the other members of the SHA-2 family of hash functions should be used; if an immediate solution is not required, then it may be appropriate to wait for the results of a hash function competition.

6. What about using other hash function designs?

   One attendee suggested that we consider a block-cipher based construct for hash function design. Because we know more about block ciphers, he argued that a block-cipher based construct has a better chance of being secure for a long time. Joux's reply that there has not been a published attack on the block cipher constructed from SHA-256 suggested that SHA-256 already could be regarded as based on a block cipher. The question to ask is whether we are building the compression function out of block ciphers the right way.

   Another attendee suggested that we should give up on the Merkle-Damgård construction because the impact of a collision is worrisome.

Session 5: Papers - Merkle-Damgård Construction and Alternatives

Session 5 consisted of three talks on the Merkle-Damgård construction for hash functions. Prushant Puniya gave a talk on a new security notion for hash functions that is stronger than collision resistance, and which would provide resistance against generic attacks. He presented several extensions of the Merkle-Damgård construction that, unlike SHA-1, satisfied the new security notion. John Kelsey of NIST gave a talk on behalf of Ron Rivest of the Massachusetts Institute of Technology on a method for making the compression function of a Merkle-Damgård hash function dependent on the index of its iteration, i.e., the first invocation, second invocation, etc. This method, which he called abelian square-free dithering, was designed to protect the hash function against vulnerabilities arising from expandable messages. Yuliang Zheng of UNC Charlotte presented two methods, which he called local tagging and global tagging, which he claimed could strengthen the Merkle-Damgård construction; one attendee observed that the methods were equivalent to altering the compression function.

Session 6: Papers - Compression Function Design

Session 6 consisted of two talks on compression function design. Danilo Gligoroski of the University of Skopje proposed the application of a substitution box with certain algebraic properties, called a quasigroup fold, after every step of the compression functions of the MD4 family of hash functions. Charanjit Jutla of IBM T.J. Watson Research Center proposed a modified message expansion code for SHA-1. He proved, with computer assistance, a security property that he argued would make the compression function resistant to the recent differential attacks.

Session 7: Panel Discussion - Desiderata: Research Agenda for Future Hash Functions

Session 7 was chaired by Lily Chen of NIST. The panelists included Don Johnson of Entrust CygnaCom, John Kelsey of NIST, Arjen Lenstra of Lucent Technologies' Bell Laboratories, Bart Preneel of Katholieke Universiteit Leuven, and Thomas Shrimpton of Portland State University.

This panel and the audience addressed the following topics regarding the research agenda for future hash functions:

1. Provably secure hash functions vs. hash functions based on heuristic methods:

   Shrimpton stated that provable security would be a good thing if we could get it; however, such hash functions are less computationally efficient than algorithms that are based on heuristic methods. Preneel advised that we could, and should, look at a hash function and see how far it can be reduced (to a problem that is believed to be hard), but we should not over-sell provable security. Lenstra stated that, since no design details of SHA-256 are available, he would rather use functions that are provably secure. However, he did not think that the cryptographic community is ready to discuss this issue; he stressed that we need to learn more.

   One attendee commented that with provable security, the interpretation of the proofs is a problem. Therefore, we must discuss the underlying assumptions of the proofs. If someone stated that he could reduce a hash function to a factoring problem, while another stated that he had looked at that function for five years and still could not break it, how would you compare the two? Which proof do you trust more?

2. The Merkle-Damgård construction:

   Shrimpton felt that we don't know as much about the Merkle-Damgård construction as we thought we did. For example, this construction promotes collision resistance of the compression function to the entire hash function, but not target collision resistance, nor "random oracleness"; so why does it promote certain properties, but not others?

Shrimpton suggested that we should think about these, as the Merkle-Damgård construction is the only iterative structure that we have for hash functions at the moment.

Lenstra commented that Joux’s attack on the Merkle-Damgård construction was a very basic result and could have been expected, but it took cryptographers a long time to figure that out. He felt that we need to do more basic research on hash functions.

Preneel stated that he felt a bit more optimistic about the situation, because we do know about the universal hash function and know how to extend it.

3. Design criteria for future hash functions:

   Johnson discussed the lifecycle of a hash function, and reminded us to be humble about the lack of knowledge. He suggested that we should allow parameterization and more rounds, if necessary; in essence, we need to be able to adapt. Kelsey commented that there may be an engineering concern for having an emergency double-the-rounds mode; Johnson felt that not having such an alternative is worse.

   Preneel stated that it may be a good thing to be more conservative in the design, and to have some flexibility; for example, AES supports different key lengths: 128, 192, 256 bits. However, we should not have too many variations; otherwise, it will be too difficult to handle all the protocols.

4. A competition for the design criteria:

   Johnson recommended that we should continue the discussion of hash function criteria~~,~~ and possibly have a competition on what evaluation criteria should be used.

   One attendee expressed the need for a decision on the hash function criteria. He felt that defining the criteria may be more difficult than creating a new hash function itself, and it would be more difficult than defining the criteria for block ciphers. He suggested that NIST form a committee to develop a straw man proposal for these criteria, and publish it for public review.

   Another attendee favored the idea of a competition, even an informal one, just to stimulate hash function research. He suggested that it need not be a design competition, but a series of workshops or conferences would help stimulate research.

   A third attendee stated that we need a competition to have requirements that are consistent. He did not think that the current requirements for hash functions are complete or consistent. Preneel disagreed, and stated that the basic requirements are very clear; it’s the additional requirements that aren’t clear, because people keep coming up with new uses of hash functions for which the requirements have not been defined.

5. Possible countermeasures for future hash functions - Dithering, output truncation, randomized hash function, etc.

   One attendee expressed the view that designing a hash function is rather easy, but performance is a big factor. Considering some of the possible countermeasures for hash functions, such as message dithering, he questioned whether these countermeasures are a good use of resources, and wondered if it would be better to increase the number of rounds by 20% instead.

6. One hash function for all, or different functions for different applications?

   Kelsey asked one attendee who was speaking at the time whether it would be better to have a generic hash function for all applications, or to have different hash functions for different applications. The attendee answered that this is something for a committee to decide. He did not think that we can have a hash function that does everything. He felt that we need to have a list of the requirements. For the future hash function proposals, we should require cryptanalysis of the function as part of the submission, rather than just having the function submitted for the public to review and analyze.

7. Any missing pieces in the research topics for future hash functions?

   Donna Dodson of NIST asked whether there were any important pieces of hash function research that were missing from the list below.

   - Compression functions - "Provably secure" vs. heuristic methods?
   - Iterative structures – Any improvements on the Merkle-Damgård's?
     - Parallelize; block length extension; resist multicollisions; prevent second pre-image and herding attacks, etc.
   - Countermeasures - Which are necessary for future hash functions?
     - Dithering, output truncation, randomize, etc.
   - Other criteria – What should we explicitly specify for future hash functions?

   In response, Shrimpton stated that he would like to see definitions and desired properties added to the list. Preneel concurred and said that we need 1) Definitions; 2) Compression function properties and parameters; and 3) Modes of operation. Johnson would like to have pseudo randomness added as a required property for future hash functions. One attendee wanted to include definitions of properties that are needed by applications. Another attendee commented that with block ciphers, we had a clear idea of what kind of properties we were dealing with, but we don't have that for hash functions. As a protocol designer, he does not know what kind of properties he can depend on, and what kind of properties he doesn't need. A third attendee urged NIST to start planning for new hash functions in parallel to doing research in this area.

Session 8: Papers - New Hash Algorithms

Session 8 consisted of three talks proposing new hash functions. Jaechul Sung of the University of Seoul proposed Fork-256, whose compression function featured four branches computed in parallel. Donghoon Chang of Korea University proposed DHA-256 (double hash algorithm). Arjen K. Lenstra of Lucent Technologies' Bell Laboratories presented VSH (very smooth hash), for which collision resistance was claimed to be provably reducible to a (presumably hard) number theoretic problem.

Session 9: Open Discussion - Future Strategy: Where Should We Go From Here?

Session 9 was a wrap-up session chaired by Bill Burr, manager of the Security Technology Group, Computer Security Division of NIST. Burr summarized the discussions that have taken place at the workshop on the following key issues.

SHA-1 Collisions

The current best estimate for an attack on SHA-1 is 2^{63} operations, which is a considerable amount of work. However, the attack remains to be verified. If a collision pair is found, some other attacks may become possible. The audience had two different viewpoints. One view was that collisions are not that important; they only matter for a few instances where it is necessary to prove the authenticity of a signature to a 3rd party. The other view was that a collision is a warning of much bigger dangers ahead. There seemed to be a fair amount of disagreement about this issue.

SHA-1 Policy

NIST was interested in knowing what the public thinks the government's immediate policy on SHA-1 should be. The following positions were stated:

1. The first priority is to get rid of MD-5.

2. It is okay to continue to use SHA-1 for the next few years, at least in the current applications, but we should encourage new applications to use something else, presumably a SHA-2 hash function. It was noted that even if some of the new applications can support the SHA-2 hash functions, these hash functions can't interoperate with other systems until all of the support base can recognize the same hash function.

3. We are currently "stuck" with SHA-1 certificates. If certificates are issued with the SHA-2 hash functions, not many of the hosts and relying parties can actually use them. Furthermore, the smart cards to be issued by the Federal government in the next few years cannot hold two certificates in them, along with all the other required information.

The SHA-2 Hash Functions

There is not much cryptanalysis on the SHA-2 hash functions yet. The following observations were made by the attendees:

1. The consensus opinion of the SHA-2 session panelists was that there may be theoretical attacks in the next decade, although these attacks probably will not be practical.

2. The SHA-2 hash functions are not that efficient in hardware.

3. NIST has heard from some companies about their plans to adopt the SHA-2 hash functions; NIST should get input from the OpenSSL and Linux communities as well.

General Observations

There was a general agreement that treating the Merkle-Damgård hash functions as a random oracle could be a problem. Of the generic attacks that have been put forth, none seems to be practical.

Another assertion was that algorithm agility is needed, so that we can have several hash functions to pick from. However, other attendees argued that algorithm agility is difficult in hardware, which is a good reason to over-build (i.e., design a really strong algorithm).

Future Hash Functions

With regard to future hash functions and a hash function competition, the following statements were made:

1 When the AES competition was begun, we seemed to have software encryption in mind, but the actual selection of the algorithm had more to do with how the candidates ran in hardware. So whatever new hash algorithms are proposed in a competition, they probably should be suitable for hardware.

2. Whatever problems there are with SHA-1 or SHA-2, they may be fixed by increasing the number of rounds. However, if performance is important, then we don't want to do things that aren't necessary.

3. We want to get beyond the limitations in the Merkle-Damgård construction, and to block generic attacks.

4. We probably want a hash function with more states and with an end state that is different from the other states.

5. We may want specialized functions, since a general purpose hash function for digital signatures may not be the right function for HMAC, for example. Some people want a hash function that does everything, while others felt that a single hash function may not be appropriate for all applications. For example, is the signing of a message digest produced by a hash function the best way to produce a digital signature?

6. If a competition were begun now, better designs may be proposed than are currently being used, but much progress is expected in the analysis of hash functions in the next few years, so it may be appropriate to wait until we know more about the basic technology of hash functions before beginning a hash function selection process.

7. The notion of a cooperative competition was suggested, whereby the best elements of a hash function would be selected and combined.

8. Provable security is interesting if the security does not have to rely on more than one assumption. However, a provably secure hash function may be very inefficient in terms of performance.

9. Randomness can be introduced into protocols so that we don't depend as much on the strength of the hash functions, but we still need a hash function that does not require randomness.

10. Consider variability in the hash function output so that legacy systems with shorter output can be retrofitted.

Strategy for the Development of Future Hash Functions

The following points were made with respect to the development of future hash functions:

1. The SHA-2 hash functions are okay for at least the next decade.

2. Some attendees felt that, although a competition could be initiated immediately, it would be better to take some time to study the basic technology and to allow the development of requirements and criteria first. Others felt that if a competition is delayed too long, many vendors will not actively participate in a lengthy academic exercise. In addition, as vendors start to rewrite their applications to take advantage of the multi-core platforms, it would be a good opportunity to introduce new hash functions as well.

   A compromise may be to have one or two more workshops on the basic criteria and work on the theory and understanding of hash functions before the competition rules are established, and the call for submissions is made. A three round competition could be considered, with one round being a collaboration of ideas.

   It was suggested that NIST provide an approximate timeline for the selection of a new hash function for planning purposes, but an opposite viewpoint was also voiced that we currently do not understand hash functions well enough to make a selection. Nonetheless, our current knowledge of block ciphers could be leveraged, since hash functions are often similar to block ciphers. In any event, some preplanning would be required before a selection date can be determined.

3. We do not have as mature an understanding of hash functions as we did of block ciphers when the AES competition was begun.

4. How many hash functions are required? One attendee proposed no more than one or two.

5. NIST should list the hash function applications, explaining the risks and provide suggestions. NIST could also identify alternative means of providing the cryptographic services currently provided by hash functions. With such guidance from NIST, industry could develop their risk management strategy.

6. It may be appropriate to initiate a parallel process to determine how hash functions can be negotiated in protocols.

4. Workshop Wrap up

NIST is planning another workshop to follow Crypto 2006 in Santa Barbara. Input for the workshop is welcome, particularly suggestions for subjects to be discussed at that workshop. A call for participation will be posted.