National Institute of Standards December 18, 2002
Gaithersburg, MD

Comments on the RMAC algorithm

I write to comment on NIST's recent consideration of the RMAC algorithm as a data authentication mechanism.

First, let me join the others in applauding NIST for taking steps to evaluate and recommend a Message Authentication Code. These objects have long been integral in the implementation of security protocols on networks and in other settings. However, I would like to make several comments regarding the current choice of the RMAC algorithm.

Several technical comments have already been posted on the NIST Modes website, and I agree with most of these, however I have some additional comments. I will strive not to be too redundant with those already made by Rogaway and Wagner.

1. The Ideal Cipher Model for MACs??

First, regarding the ideal cipher model used in the proof of the (published) RMAC paper: as a researcher who has used both the ideal cipher model and the standard reduction-based proof modes, I must agree that the use of the ideal cipher model is inappropriate where it really is not mandated (because the reduction-based paradigm is sufficient as has been amply demonstrated by several other researchers producing similar algorithms to RMAC). Use of the ideal cipher model can often produce proofs which may give assurances of security where none exist simply because the model is too strong (ie, assumes too much) of the actual object which will be inserted (eg, AES) when an algorithm is implemented. To take a concrete example, a recent paper of mine used the ideal cipher model to show the security of some cryptographic hash functions; however it would be easy to construct objects which would be considered "good" block ciphers (in the sense that they are pseudorandom permutation families, the normal definition of "good" for block ciphers) and yet which are completely insecure when used to build cryptographic hash functions.

The main point is this: why work in a model where you model **very** strong properties of your building blocks when another model where weaker assumptions are perfectly adequate to achieve the same goals? To my knowledge, no other MAC has been designed using this strong model.

2. A "Model" versus an "Assumption"

Another comment: the RMAC authors keep referring to the ideal cipher model as "as an assumption." It is not. The ideal cipher model is a mathematical setting in which proofs are carried out. As has been pointed out repeatedly in the analogous debates surrounding the Random-Oracle-Model, a proof of security carried out in a model gives no guarantees about the security of an algorithm once some actual object (like AES) is inserted in place of the ideal cipher used in the model. On the other hand, an "assumption" is something that is either true or false. It is theoretically possible to even ascertain this; for example, a common assumption made of AES is that, for a random fixed key, an attacker cannot distinguish it from a randomly-chosen permutation, provided the attacker's computational parameters are constrained to some generously large values. This may or may not be true. However, in the ideal cipher model, we can never say things like "assume AES is an ideal cipher". It makes no sense. It's like saying "assume that 147 is a random number."

3. Do we really need added security?

Regarding the security bounds of RMAC, while it is true that the claimed bound goes beyond the normal quadratic degradation in security seen with non-randomized MACs, the security of the latter MACs is almost certainly adequate for virtually every computing setting it is likely to be found in. Assuming AES with its 128-bit blocksize, a non-randomized MAC typically allows up to, say, 2^{50} message blocks to be processed before the key should be changed. This is an enormous amount of data. Even if one million message blocks per second were MACed under the same key, it would take more than 30 years before the key would have to change. To take another numerical example, say our maximum message length is 10Kb and an attacker watches 1,000 MAC tags per second fly by for a month. Then his chances at producing a forgery are less than 1 in a trillion.

This 2^{-64} forgery probability is more than 4 billion times more secure than the security levels used by the banking industry for decades. Do we really need to worry that it's not strong enough?

4. On the Wisdom of Changing RMAC in Subtle Ways

Finally, I would like to agree that changing RMAC by adding a "salt" is a very dangerous and ill-advised step by NIST. Creating cryptographic primitives and modes is a very touchy business and it is extraordinarily easy to go astray. Often even the most innocent looking adjustment can result in a complete loss of security. The normal standard for security in the modern community is that we give both a precise definition and a rigorous proof of security under the weakest possible assumptions. Then this proof is subjected to peer review if it is submitted to a strong conference or journal. Even then there is no absolute guarantee: the proofs could still be flawed and the assumptions could prove to be

false, but the track record so far would seem to indicate that this process has produced a set of **far** more secure, robust, and trustworthy algorithms than has the old method of "intuitive" design.

5. A Suggestion

My suggestion would be to focus on a MAC which has a proof of security using the reduction-based approach. There are several existing algorithms available with various minor advantages and disadvantages. And if we truly believe that quadratic degradation in security must be overcome (eg, because we're using a short-blocksize cipher like TDES), there are very mature and standard methods for doing this as well, still remaining within the conservative reduction-based proof paradigm.

John Black
Assistant Professor
Department of Computer Science
University of Colorado at Boulder