UNIVERSITY OF CALIFORNIA, BERKELEY

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO　　SANTA BARBARA • SANTA CRUZ

Please Reply To:　　BERKELEY, CALIFORNIA 94720

David Wagner
Assistant Professor
EECS Department
University of California, Berkeley
Berkeley, CA 94720-1776 USA
+1-510-642-2758

daw@cs.berkeley.edu

December 5, 2002

Subject: Comments on RMAC

To whom it may concern:

I'd like to submit some brief comments on the NIST draft of the NIST standard, RMAC.

Most importantly, I wanted to pass on my endorsement of Phil Rogaway's analysis of Dec 2, 2002, titled "Comments NIST's RMAC Proposal." After reading Rogaway's work carefully, I concur with his analysis, and I agree with his conclusions. I urge you to give his comments careful consideration.

After looking at these issues, my feeling is that, from a technical point of view, RMAC is probably not the best choice for standardization. Don't get me wrong—RMAC is interesting research—but the goals of practical deployment are sometimes a little different from the goals of research. In this case, I believe we can do better than AES-RMAC for practical systems.

In particular, let me emphasize the value of reduction-based provable security. As Rogaway explains, other CBC-MAC based schemes, such as XCBC, EMAC, TMAC or OMAC, are probably safer from a security point of view, not least because they allow reduction-based provable security.

Perhaps I should clarify my meaning. As I see it, there are two independent axes we can use to evaluate the security provided by a cryptosystem:

Claimed security level:
: What is the claimed cost of defeating the security mechanism? How many chosen plaintexts, steps of computation, bytes of memory do we expect that an adversary would need to break the scheme?

Assurance:
: How much confidence do we have that the claimed security level accurately represents the true security level of the scheme? What are the odds that someone will find an unexpected attack on the scheme?

: Often schemes come with a proof of their security claims, under certain assumptions. However, this is not the end of the story; it is only the beginning of the story. In such cases, we need to evaluate several aspects of these proofs: How likely is the proof to be correct? How simple is the proof? How well-studied is the proof? What theoretical model is the proof done in? How much confidence do we have in the assumptions made in the proof? (After all, a proof of security under assumption X is not worth much if X turns out to be false.)

Note that these two aspects must be evaluated separately. With this background, we can now compare the security of RMAC to its natural competitors (XCBC, EMAC, TMAC, OMAC, etc.).

Claimed security level:

AES-RMAC has a higher claimed level of security than AES-XCBC. Loosely speaking, AES-XCBC claims security for only up to around 2^{64} messages, after which one must change keys; AES-RMAC's main claim to fame is that it is claimed to remain secure even if we change keys much less frequently.

But wait! Don't be overly distracted by this difference. Both AES-RMAC and AES-XCBC offer security levels that are more than adequate for all practical purposes[1]. Our MAC's don't need to be secure for 2^{64} messages, let alone more than that; it's a good idea to change keys long before that, no matter how secure the underlying primitive may be.

Hence, both AES-RMAC and AES-XCBC are adequate in this respect, and there is not much reason to prefer one over the other in this area.

Assurance:

XCBC offers higher assurance than RMAC. Both come with a proof of security, under some assumptions, but my feeling is that we can probably have more confidence in the assumptions in XCBC's proof than we can have in the assumptions in RMAC's proof.

RMAC's proofs requires stronger assumptions. For instance, RMAC's security proof requires the cipher to be secure against related-key attacks, while XCBC does not. Note that the AES has not been evaluated very carefully for security against related-key attacks, and what analysis has been done suggests that AES may have less margin of security against related-key attacks than against non-related-key attacks. In contrast, the assumptions required for XCBC's proof of security are better-studied—they align directly with what cryptanalysts study when they examine the security of AES—and so we can have greater confidence in the correctness of the assumptions found in XCBC's proof of security.

This is no accident. Indeed, XCBC was designed to achieve reduction-based provable security, and the notion of reduction-based security was formulated to model what security properties we usually expect from a block cipher and to exclude any assumptions about, for instance, security against related-key attacks. Consequently, reduction-based proofs of security (like XCBC's security proofs) offer higher assurance than proofs in the ideal cipher model (like RMAC's security proofs), and this explains why XCBC has better assurance than RMAC.

In summary, AES-XCBC offers higher assurance than AES-RMAC.

[1] The story is a little different for Triple-DES-RMAC and Triple-DES-XCBC, because Triple-DES has a shorter block length and hence Triple-DES-XCBC is only good for up to around 2^{32} messages. However, it seems that Triple-DES-RMAC has other issues (issues that I don't want to get into here, because they would take us far afield), and I think Triple-DES-XCBC is probably good enough in practice as long as the standard includes some cautions about how often to change keys.

With this background, it should become clear why I am mildly concerned about NIST's choice of RMAC. XCBC (or its cousins, EMAC, TMAC, OMAC, and so on) seem to have better security properties.

My advice would be to re-consider the choice of RMAC and give serious thought to standardizing on some CBC-MAC-based scheme with provable security in the reduction-based setting. I believe this would improve the confidence we can have in the standard. If you follow this direction, my preference would be for XCBC (and I think XCBC is ready for standardization without any need to impose further delays), but there are other reasonable choices, too.

In closing, let me praise you for taking on the challenge of choosing a standard AES-based MAC. I greatly appreciate NIST's efforts to standardize on a secure MAC transform, and I remain confident that this effort will make a significant contribution to computer security. Thank you for the chance to comment on the draft standard. I apologize that my comments were not submitted earlier, but I hope they will be helpful in your evaluation of the draft standard.

Yours Sincerely,

David Wagner
(affiliation & title used for identification purposes only; I do not speak for the University, and my opinions are solely my own)