Comments on NIST's RMAC Proposal

Phillip Rogaway
UC Davis and Chiang Mai Univ
rogaway@cs.ucdavis.edu
http://www.cs.ucdavis.edu/∼rogaway

2 December 2002

1 Summary

I would like to applaud NIST for moving forward with a MAC proposal [Draft]. But the mechanism that NIST has proposed is not an appropriate one to move forward with:

(a) *Poor assurance.* NIST's seeded-RMAC algorithm does not enjoy reduction-based provable security. Quite the opposite: it is demonstrably impossible to prove it secure, in the sense that one would like, under the usual assumption that our community makes about block ciphers (namely, security in the sense of a pseudorandom permutation, or PRP).

(b) *New design.* Though the NIST draft is based on a submission to them [JJV], their adaptation of this submission has resulted in a fundamentally new algorithm that aims to solve a fundamentally different cryptographic problem. (Is it NIST's intent to undertake the design of new cryptographic objects?)

(c) *Poor efficiency.* RMAC uses one or two more block-cipher calls than the natural alternative (the CBC MAC variant known as XCBC [BlRo]). It effectively re-keys with every message that is MACed, adding further overhead. It is unnecessarily stateful or probabilistic, yet another source of overhead and a practical drawback.

(d) *Wrong motivation.* Fundamentally, RMAC addresses the wrong problem for a contemporary cryptographic standard: with the emergence of the AES, beyond-birthday-attack security is just not an issue of much present-day concern.

In standardizing a new mode of operation the first two goals are security and efficiency. Security should be demonstrated in the reduction-based provable-security paradigm: the belief that AES (say) is a good PRP should be enough to conclude that some MAC based on it is secure. This has become the generally-accepted way of demonstrating security. One might even say that a MAC design that fails to do at least this much fails to meet the accepted professional standard for the design of a new mode of operation. Efficiency is another central goal, and for an object as simple as RMAC this is rather easy to gauge. In terms of both demonstrated security and efficiency, NIST's algorithm does not fare well.

The remainder of this note assumes familiarity with the NIST draft and adopts the notations used there.

2 Security

The NIST draft [Draft] is adapted from a proposal submitted to them [JJV]. The NIST draft inherits problems already present in the submission, and then it adds new problems of its own.

2.1 Problems Lifting the Submission to the NIST Draft

A major difference between the NIST draft and the submission it is based on is that the submission provides a construction for a randomized MAC and uses the conventional definition for a randomized MAC, while the NIST draft constructs a new kind of object and provides no corresponding notion of security for it. We call the new kind of object constructed in the NIST draft a *salted MAC*.

A number of factors make clear that the NIST draft is not specifying a randomized MAC. Section B of the NIST draft makes explicit that the salt R may be any nonce. Even before then, the syntax of a MAC is modified to include a salt-value R. Finally, it is untenable to think that the salt value R is a random but negligibly-often-repeating value when recommended parameter values include $r = |R| = 16$ bits (with $r = 16$ a random salt value R would repeat within a few hundred uses).

From our point of view, the NIST draft invents a new mode of operation. We refer to the algorithm of the NIST draft as *salted-RMAC* and the algorithm of the original submission as *randomized-RMAC*. We let the term *RMAC* encompass both mechanisms. Randomized-RMAC corresponds to salted-RMAC using parameter set II/case $b = 64$ or parameter set V, and where the salt R is required to be a random value, and where the security notion, as given below, is modified so that there is no expectation that the salt R is authenticated.

A salted MAC is not an object that has been defined in the literature. What should the definition of security for such an object be? Truly it is not the obligation of the analyst of a mode to reverse-engineer its proper goal, but the desirable definition of security would work as follows:

> Consider an adversary A that asks queries of the form (R, M) to a MAC oracle (R is the salt and M is the message). In response to a query (R, M) the oracle returns the pair $(R, T) = \mathrm{MAC}_K(R, M)$ where K is a random key chosen and fixed at the beginning of the experiment. No salt value R may be repeated by the adversary during its queries. When done asking queries the adversary A outputs a forgery attempt $(M^*, (R^*, T^*))$. The adversary is said to *forge* if (R^*, M^*) is *new*, meaning that there was no earlier query (R^*, M^*), and $(R^*, T^*) = \mathrm{MAC}_K(R^*, M^*)$. The *advantage* of an adversary A is the probability that it forges. Informally, a salted MAC is said to be *secure* if "reasonable" adversaries can have only "small" advantage.

Weaker notions of security for a salted MAC are possible, though they would seem to be less useful. (In particular, experience from using MACs in other domains, particularly entity authentication, has made clear that relaxing the notion of a forgery to M^* being new is not desirable for a general-purpose tool.) In the remainder of this note, we assume that the goal of salted-RMAC is (or at least should be) security in the sense that is described above (when $r > 0$).

It is a major problem with the NIST draft that it provides an object that isn't what is considered in nor provided by the randomized-RMAC proposal [JJV]. One can't change the syntax of an object, the attack model, and the notion of adversarial success without having a huge impact! Because randomized-RMAC is fundamentally different from salted-RMAC, theorems about randomized-RMAC [JJV] say little or nothing about salted-RMAC [Draft].

2.2 No Reduction-Based Provable-Security Result is Known for RMAC

Theorems from the randomized-RMAC paper [JJV] are in the ideal-cipher model—they are not reduction-based provable-security results.[1] Furthermore, salted-RMAC is a construction for which a reduction-based provable-security result, under the traditional (PRP) assumption, is impossible: salted-RMAC need not be secure when its block cipher is secure (as a PRP). We show this in Section 2.3. Thus good assurance for salted-RMAC is intrinsically out-of-reach.

The preceding paragraph has a lot to understand; let me back off and give a bit of background.

Basically, there are two different approaches for doing a security proof about a block-cipher-based construction: *reduction-based provable security* (also called *provable security* or *the standard model*) and a proof in the *ideal-cipher model*.

In the reduction-based provable-security approach one would show that *if* there exists a reasonable adversary A that breaks a given MAC that is built from a block cipher E, *then* there exists a reasonable adversary B that breaks E. Such a proof is called a *reduction*. Breaking E has usually come to mean that B can do a good job at distinguishing a black-box for E_K (for a random secret key K) from a black-box permutation π (for a random permutation π having the same domain and range as E_K). Using reduction-based provable security cryptographers have proven the security of MACs such as the CBC MAC, DMAC, HMAC, PMAC, UMAC, and the XOR MACs. (We have proven the security of numerous other kinds of objects, too.) The approach has become vastly popular because it lets scientists translate our assurance about a primitives (say AES) into assurance about a higher-level construct that uses the primitive.

The ideal-cipher model is completely different. There is no reduction. Instead one adopts a model of computation where the block cipher is treated as a family of random permutations. One makes claims solely within this model—you never pass back to the "real" world. Though I myself have used the ideal-cipher model in some of my work, the model is fairly criticized and is typically used only when there exists no viable alternative. The ideal-cipher model is the approach taken for justifying the security of the DESX construction, double encryption, and block-cipher-based hash-function constructions. In these cases, reduction-based provable security is believed to be impossible, and so one is forced by the setting to move to a less-well-founded alternative.

To the best of my knowledge, the ideal-cipher model was never before [JJV] used to argue security for a MAC; it simply isn't needed. The randomized-RMAC paper [JJV] resorts to the ideal-cipher model when the underlying goal—constructing a "beyond the birthday bound" MAC—is something that we understand how to do in the standard model.

The classical approach for a beyond-the-birthday-bound MAC is the version of the Carter-Wegman construction where one encrypts $H_{K1}(M)$ (the universal-2 hash of M) by xoring it with $F_{K2}(N)$ where F is a pseudorandom function and N is a nonce. This gives a reduction-based provably-secure MAC with a bound better still than that of RMAC.[2] The UMAC algorithm is an instance of this paradigm. Another example for a practical, reduction-based provably secure, beyond-birthday-bound MAC is given in [BGH].

We appreciate moving to the ideal-cipher model when one is trying to achieve a goal that is difficult or impossible to achieve in the standard model. But that is not this situation.

[1] Section 4.1 of the randomized-RMAC paper [JJV] might look like it gives a reduction-base provable-security result. But, first of all, Section 4.1 assumes the use of a block cipher with a $2b$-bit key (which is not mandated in the NIST draft). Regardless, the relevant portion of Section 4.1 is incorrect in suggesting that it is providing a standard-model proof: the authors provide no reduction (they continue to appeal to the ideal-cipher model) and it would appear instead that no reduction is possible.

[2] If F is instantiated by a block cipher one gets birthday bounds in q, the number of queries, but not in L, the total number of blocks queried. This quadratic loss in q can be reduced, if desired, by using any method from the body of literature on doing good PRP to PRF conversions.

Results in the ideal-cipher model are far less desirable than reduction-based provable security. Ideal-cipher results are particularly dangerous when dealing with DES and TDES, because the key-complementation property of these primitives is completely contrary to the ideal-cipher model. The ideal-cipher model is not without danger when applied to AES. The AES is a new algorithm whose security analysis has focused on, at most, distinguishing AES from a random permutation. Saying that AES is well-modeled by an ideal-cipher suggests that it has excellent key-scheduling, for example, and no related-key attacks. This is well beyond what people can or should believe about AES. Assuming the security of AES as a PRP is, instead, an accepted assumption.

2.3 Reduction-Based Provable Security is Impossible for salted-RMAC

Let $E : \{0,1\}^k \times \{0,1\}^b \to \{0,1\}^b$ be a good block cipher: one that is secure in the sense of a PRP. Consider the block cipher $E' : \{0,1\}^{2k} \times \{0,1\}^b \to \{0,1\}^b$ defined by $E'_{K'K}(X) = E_K(X)$. The first half of the key bits of E' are not used. Still E' is as secure as a PRP as E is. Apply the salted-RMAC construction from the NIST draft to E' and observe that it is trivial to forge RMAC-E' according to the notion sketched in Section 2.1. This shows that E being a secure PRP does not imply that salted-RMAC-E is a good MAC.

Of course examples like the above do not say that it is trivial to forge under salted-RMAC-AES, for example. That's not the point. The point is that any reduction-based provable-security result is out of reach (under the traditional assumption about the underlying block cipher). And without that, our community has nothing that we are comfortable to fall back on.

(If the above is too formal, understand, quite informally, that the fundamental problem with salted-RMAC is that one shouldn't be messing with a block-cipher key by xoring in a salt. Because what cryptographers like to assume about block ciphers is silent about what happens when you do such a strange thing as that.)

3 Efficiency

In this section we compare the efficiency of salted-RMAC and the efficiency of XCBC [BlRo], a CBC MAC variant that was designed to be efficient and have reduction-based provable security.

1. *Block-cipher calls.* Salted-RMAC uses $1 + \lceil(|M| + 1)/b\rceil$ block-cipher calls, while XCBC uses $\lceil|M|/b\rceil$ block-cipher calls. That is, salted-RMAC uses one more block-cipher call than XCBC does when $|M|$ is a non-multiple of the block length, and salted-RMAC uses two more block-cipher calls than XCBC when $|M|$ is a multiple of the block length. In many contexts, the spending of an extra one or two block-cipher calls can is significant (because the messages being MACed are often or always short).

2. *Block-cipher keying.* With XCBC all block-cipher calls use the same key; with RMAC the block cipher is effectively re-keyed with every message. Typical block ciphers, like AES and TDES, have a non-trivial cost for key setup. This cost is normally incurred in both software and hardware implementations. XCBC was specifically designed to avoid paying this cost.

3. *Tag length.* The length of an XCBC tag is shorter (for parameter sets III, IV, and V) because there is no need to choose and to communicate the salt.

4. *Statefulness.* For salted-RMAC (parameter sets III, IV, V) one must maintain state or generate pseudorandom values; XCBC, instead, is deterministic. Generating pseudorandom values is normally done by maintaining state and using the block cipher yet again (with a key that

should be used only for this context). This is an additional computational cost as well as an error-prone process.

5. *Key length.* This is the only efficiency measure we know where salted-RMAC "wins": XCBC uses a $3k$-bit key while salted-RMAC uses a $2k$-bit key. (However, a salted-RMAC implementation that generates pseudorandom salt using the same underlying block cipher would use a $3k$-bit key and yet one further block-cipher call.)

If there were a demonstrated security benefit for paying the extra costs enumerated above one would have to examine that benefit and see if it was worth the price. In this case, however, there would seem to be inferior efficiency *and* inferior demonstrated security.

We mention that there is an IETF Internet Draft (with one author from NIST, in fact) defining XCBC and including test vectors [FrHe].

4 Addressing the Wrong Problem

The basic CBC MAC has problems that a modern standard should fix. The worst problems with the basic CBC MAC are:

(1) *Insecurity.* The basic CBC MAC is completely insecure across messages of varying lengths (assume you output all b bits).

(2) *Limited domain.* The basic CBC MAC is only defined on $(\{0,1\}^b)^+$ (whereas one would like a message space of $\{0,1\}^*$).

It is desirable to address these problems without losing key properties of the basic CBC MAC: (a) it is a pseudorandom function (PRF); (b) it enjoys reduction-based provable security; (c) it uses $\lceil |M|/b \rceil$ block-cipher calls; and (d) all of those calls employ the same key. The significance of properties (b)–(d) have been discussed. The significance of (a) is that the MAC tag is shorter, there is no reliance on randomness or state, and the utility of the constructed object goes beyond message authentication and extends to goals like pseudorandom generation and key-separation.

RMAC manages to address issues (1) and (2) at the expense of losing all of (a)–(d). But then RMAC was designed to address a very different issue:

(3) *Quadratic security bounds.* The CBC MAC has proven security that degrades in $L^2/2^b$ where L is the total number of blocks queried by the adversary. Some other constructions (e.g., Carter-Wegman MACs, UMAC, XOR/ctr) do better in this regard.

We view issue (3) an issue of relatively minor importance to current practice:

(i) Every well-known block-cipher-based encryption mode likewise has a privacy guarantee that degrades in $L^2/2^b$ (are you going to "fix" CBC mode and CTR mode encryption?) And in terms of relative importance, an encryption-mode's $L^2/2^b$ security degradation is arguably a worse problem than a MAC's security degradation of the same amount, because it makes no sense to "break a MAC" once the communications session is torn down.

(ii) Regardless, for 128-bit blocksize block ciphers, which are to be the norm, the bound of $L^2/2^b$ is numerically satisfying.

We are not saying that it is not worthwhile to address issue (3) when elegant, provably-secure, improved-bounds methods become known. We simply don't see it as being a major issue today, and we don't see it as a desirable tradeoff to solve (3) and the expense of (a)–(d).[3] Moreover, improved security bounds for symmetric encryption schemes and MACs is a highly unsettled area of cryptographic research—reason enough for avoiding early work from this domain.

We will mention a final drawback with the CBC MAC (one that is probably more significant than issue (3) to emerging cryptographic practice):

(4) *Not parallelizable.* The CBC MAC is inherently sequential, limiting its applicability for ultra-high-speed applications.

Of course RMAC does nothing to address this issue.

5 Concluding Remarks

Cryptography is not where it was 20 years ago. Back then, a block-cipher mode of operation could only be designed and justified by attack-centric analysis and informal arguments. We knew no other way. But nowadays our community has a widely-accepted methodology that leads to much better assurance: reduction-based provable security. Because of this development, informal and error-prone arguments (like those given in Appendix A of the NIST draft) are no longer seen as particularly necessary or credible in this domain. We have developed better ways.

MACs, in particular, have become a well-understood primitive. We have a wide variety of constructions that are simple, efficient, and enjoy reduction-based provable security with standard, acceptable bounds. But, for whatever reasons, NIST did not select such a construction. Instead, they built on a recent paper [JJV] that makes claims in a non-standard model in order to try to get improved concrete security bounds. Starting with this paper [JJV], NIST did a lot of further, independent, design. They ended up with a kind of object that isn't even a conventional MAC, and isn't supported by any published scientific work. We don't think this is a right way to go. We recommend abandoning RMAC and choosing a more mature construction.

6 Acknowledgments

The author shared earlier versions of this note with Mihir Bellare, John Black, and David Wagner, obtaining insightful comments from all of them. Many thanks.

My research is funded by the National Science Foundation NSF CCR-0208842 and a gift from CISCO Systems.

[3] Issue (3) should have been regarded as more important a few years ago, when applying the CBC construction to a 64-bit block-cipher, DES or TDES, was the norm. Such applications must understand that well fewer than 2^{32} blocks should be CBC MACed under a single key. Similarly, a 128-bit block-cipher should MAC well fewer than 2^{64} blocks under a single key.

References

[BGH] M. Bellare, O. Goldreich and H. Krawczyk. Stateless Evaluation of Pseudorandom Functions: Security Beyond the Birthday Barrier. CRYPTO '99, pp. 270–287, 1999. www.cs.ucsd.edu/users/mihir

[BHK+] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, UMAC: Fast and Secure Message Authentication. CRYPTO '99, pp. 216–233, 1999. www.cs.ucdavis.edu/∼rogaway

[BlRo] J. Black and P. Rogaway. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. CRYPTO '00. pp. 197–215, 2000. Also see "XCBC" at csrc.nist.gov/encryption/modes/proposedmodes

[Draft] M. Dworkin. DRAFT Recommendation for Block Cipher Modes of Operation: The RMAC Authentication Mode. Publication 800-38B, 4 November 2002. csrc.nist.gov/publications/drafts.html

[FrHe] S. Frankel and H. Herbert. The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec. Internet Draft draft-ietf-ipsec-ciph-aes-xcbc-mac-02.txt. June 2002. www.ietf.org/internet-drafts/draft-ietf-ipsec-ciph-aes-xcbc-mac-02.txt

[JJV] E. Jaulmes, A. Joux, and F. Valette. On the Security of Randomized CBC-MAC Beyond the Birthday Paradox Limit A New Construction. FSE 2002. See also: eprint.iacr.org/2001/074/ (posted 31 Aug 2001, revised 28 Nov 2002) and csrc.nist.gov/encryption/modes/proposedmodes/

[PeRa] E. Petrank and C. Rackoff. CBC MAC for Real-Time Data Sources. *J. of Cryptology*, vol. 13, num 3, pp. 315–338, 2000. www.cs.technion.ac.il/∼erez/

[WeCa] M. Wegman and L. Carter, New Hash Functions and their use in Authentication and Set Equality. *J. of Comp. and System Sciences*, vol. 22, pp. 265–279, 1981.