Update on UMAC Fast Message Authentication

J. Black S. Halevi H. Krawczyk T. Krovetz P. Rogaway

May 12, 2000

The UMAC message authentication code (MAC) proposed by us at CRYPTO '99 combined a software-optimized hash-function family, NH, with a pseudorandom function (CBC-RC6 or HMAC-SHA1) [1]. For a MAC with forging probability of 2^{-60} we reported peak speeds of **1.0** Pentium II cycles-per-byte (cpb) using Pentium MMX SIMD parallelism, and about **1.9** cpb without. Since CRYPTO '99 we have continued to refine UMAC. An Internet Working Draft is now ready [2]. Here we summarize a few of the refinements to UMAC embodied by that spec, plus our most recent timings. The refinements were aimed at achieving three main goals:

• ***Faster MACing of short messages.*** Recent discussions with David McGrew and Scott Fluhrer (Cisco Systems) has reminded us that much traffic requiring MACs, particularly IP flows, is heavily geared towards short messages. Thus we have done more to achieve the best performance we can for very short messages. According to these Cisco folks, a fair rule-of-thumb for the distribution on message-sizes on an Internet backbone is that roughly one-third of messages are 43 bytes (TCP ACKs), one-third are about 256 bytes (common PPP dialup MTU), and one-third are 1,500 bytes (common Ethernet MTU). The following table gives current timings for the original UMAC (UMAC-STD and UMAC-MMX) and their corresponding replacements (UMAC32 and UMAC16). Timings are in cycles-per-byte and gathered on a 700 MHz Pentium III under gcc 2.95, mixed C/assembly. Both UMAC16 and UMAC32 give 64-bit tags with forging probability of approximately 2^{-60}.

	43 bytes	256 bytes	1500 bytes	256 kbytes
UMAC32	**16.3**	**3.8**	**2.1**	**1.9**
UMAC-STD	52.9	12.3	3.8	1.9
UMAC16	**14.0**	**2.7**	**1.2**	**1.0**
UMAC-MMX	35.9	4.5	1.7	1.0

• ***Minimizing "cryptography."*** After compressing a message with NH, we now hash the result to a fixed length (for use in a Wegman-Carter construction [3]) using a polynomial-based hash function. This is in contrast to our former approach of processing the compressed (but still unbounded in length) message with a PRF. Our new approach has the heuristic benefit that it minimizes the use of cryptography, and the provable-security benefit that the security analysis is now completely independent of the length of the messages being MACed.

• ***Selective-assurance verifiability.*** The current UMAC is constructed so that if one computes a 64-bit MAC, say, then one can verify the first 32 bits at nearly twice the speed of verifying the whole thing, and (with UMAC16) one can verify the first 16 bits at nearly four times the speed. Most MACs, including the earlier version of UMAC, do not have this have this property. Thanks to David Balenson and David Carman (NAI Labs) for this suggestion.

References

[1] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC: Fast and secure message authentication. In *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 216–233. Springer-Verlag, 1999.

[2] T. Krovetz, J. Black, S. Halevi, A. Hevia, H. Krawczyk, and P. Rogaway. UMAC — Message authentication code using universal hashing. IETF Internet Draft, draft-krovetz-umac-00.txt, `www.cs.ucdavis.edu/~rogaway/umac`, 2000.

[3] M. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *J. of Computer and System Sciences*, 22:265–279, 1981.