Patent-Free Authenticated-Encryption As Fast As OCB

Ted Krovetz
Computer Science Department
California State University
Sacramento, California, 95819 USA
tdk@acm.org

***Abstract*—This paper presents an efficient authenticated encryption construction based on a universal hash function and block cipher. Encryption is achieved via counter-mode while authentication uses the Wegman-Carter paradigm. A single block-cipher key is used for both operations. The construction is instantiated using the hash functions of UMAC and VMAC, resulting in authenticated encryption with peak performance about ten percent slower than encryption alone.**

Keywords- Authenticated encryption, block-cipher mode-of-operation, AEAD, UMAC, VMAC.

I. INTRODUCTION

Traditionally when one wanted to both encrypt and authenticate communications, one would encrypt the message under one key and authenticate the resulting ciphertext under a separate key. Encryption in such a scenario would often use a block-cipher mode of operation, while authentication would usually use another mode or HMAC [6]. If the block cipher encrypted blocks at a rate of x processor cycles per byte (cpb), then the combined process of encryption plus authentication would require at least $2x$ cpb and the management of two separate keys.

Recently proposed modes of operation combine encryption and authentication under a single key. Some of the modes also switch to faster Wegman-Carter authentication based on universal hashing [8,9]. This switch can bring authenticated encryption down to nearly x cpb because recent Wegman-Carter schemes are as fast as 0.5 cpb—much faster than any known block cipher. One other method of authenticated encryption, typified by OCB mode, authenticates a message as a byproduct of its encryption. These modes are very efficient, but are proprietary, require licenses and cannot be used until patent disputes are resolved. With the exception of OCB, all algorithms examined in this paper are patent-free and can be used freely without securing any license.

This paper examines a general method for converting a universal hash function into an authenticated encryption scheme that uses a single key for both encryption and authentication. The resulting construction is provably secure and has peak efficiency close to the sum of counter-mode encryption and the peak speed of the chosen universal hash function. As an example, the construction is applied to the AES block cipher and VHASH hash family [4]. The resulting authenticated encryption scheme peaks at 12.8 cpb, while OCB peaks at 13.9 cpb in our experiments. The paper closes with a performance comparison of several well-known authenticated encryption algorithms [6].

II. SECURITY DEFINITIONS

We adopt the notions of security from [7], and summarize them less formally here. An authenticated encryption with associated data (AEAD) scheme is a triple $S = (K,E,D)$, where K is a set of keys, and E and D are encryption and decryption functions. Encryption occurs by computing $E(k,n,h,p,f)$, which returns (c,t), for key k, nonce n, header h, plaintext m and footer f. Ciphertext c is the encryption of p, and tag t authenticates h, c and f. Decryption occurs by computing $D(k,n,h,c,f,t)$, which returns p only if (c,t) is a legitimate result for $E(k,n,h,p,f)$ and "invalid" otherwise.

AEAD scheme S is secure if Adv(S, PRIV) and Adv(S, AUTH) are both small given an adversary with reasonably limited resources. Adv(S, PRIV) is defined to be the maximum probability that an adversary could distinguish whether an oracle O has been instantiated as $E(k,-,-,-,-)$ for a randomly chosen k or if O simply returns (an appropriate number of) random bits instead of a legitimate (c,t) pair. For the definition of Adv(S, AUTH), let the adversary have an oracle O instantiated as $E(k,-,-,-,-)$ for a randomly chosen k. A forgery occurs if the adversary can produce an (n,h,c,t) for which $D(k,n,h,c,f,t)$ is valid, and c was never returned by the oracle. Adv(S, AUTH) is the maximum probability an adversary is able to create a forgery. In both the encryption and authentication cases, it is assumed the adversary never repeats a nonce to its oracle.

III. WC–AE CONSTRUCTION

Let H be an ε-almost-delta-universal hash family with all member functions having the domain of arbitrary strings and co-domain of L-bit strings. We will not describe delta-universal hash families in this paper, except to say that they can be used in Wegman-Carter authentication schemes [8,9]. Assume that a random j-bit string can be used to select a random element of H, and that the function indicated by string b is H_b. Let $\langle i \rangle_n$ represent the n-bit binary encoding of integer i, and $b[a \ldots c]$ repre-

sent the substring of b including bit indices a through c. Let $\|$ be string concatenation and $|b|$ the bit-length of string b.

We now define AEAD scheme WC-AE. Let K be the set of all functions from L bits to L bits. Choosing a random g from K then defines the following functions (where n is an $L/2$-bit string and h, p and f are arbitrary strings):

$E_g(n,h,p,f)$:
  $b = g(\langle 1\rangle_1 \| \langle 0\rangle_{L-1}) \| g(\langle 1\rangle_1 \| \langle 1\rangle_{L-1}) \| g(\langle 1\rangle_1 \| \langle 2\rangle_{L-1}) \| \ldots [1\ldots j]$
  $epad = g(n \| \langle 1\rangle_{L/2}) \| g(n \| \langle 2\rangle_{L/2}) \| g(n \| \langle 3\rangle_{L/2}) \| \ldots [1\ldots|p|]$
  $c = p \oplus epad$
  $tpad = g(n \| \langle 0\rangle_{L/2})$
  $t = H_b(h \| c \| f \| \langle|h|\rangle_{64} \| \langle|c|\rangle_{64}) + tpad \bmod 2^L$
  return (c,t)

$D_g(n,h,f,c,t)$:
  $b = g(\langle 1\rangle_1 \| \langle 0\rangle_{L-1}) \| g(\langle 1\rangle_1 \| \langle 1\rangle_{L-1}) \| g(\langle 1\rangle_1 \| \langle 2\rangle_{L-1}) \| \ldots [1\ldots j]$
  $tpad = g(n \| \langle 0\rangle_{L/2})$
  $t' = H_b(h \| c \| f \| \langle|h|\rangle_{64} \| \langle|c|\rangle_{64}) + tpad \bmod 2^L$
  if $t \neq t'$ return "invalid"
  $epad = g(n \| \langle 1\rangle_{L/2}) \| g(n \| \langle 2\rangle_{L/2}) \| g(n \| \langle 3\rangle_{L/2}) \| \ldots [1\ldots|p|]$
  $p = c \oplus epad$
  return p

Theorem: Adv(WC-AE, PRIV) = 0 and Adv(WC-AE, AUTH) $\leq \varepsilon$ when all nonces begin with a zero bit.

Proof: Because g is chosen from all possible L-bit functions, each invocation on different inputs returns a uniformly distributed L-bit string. This means b, and thus the choice H_b, is uniformly distributed. All other inputs to g are distinct over all invocations of E so long as n is unique for each and always begins with a zero bit. This means *tpad* and *epad* will be independent uniformly distributed strings for each invocation of E. This results in both c and t being uniformly distributed, and so Adv(WC-AE, PRIV) = 0. The value t is computed using a standard Wegman-Carter MAC construction, and so Adv(WC-AE, AUTH) $\leq \varepsilon$. ♦

For a more thorough examination of counter-based encryption and Wegman-Carter message authentication see [1,8,9].

The set of all L-bit functions is not a practical key set, so instead we use a block cipher in a realization of WC-AE. Block ciphers are designed to resemble random permutations, which in turn can be used in the place of a random function. Let B be a block cipher from L bits to L bits. We use standard notions of block-cipher security. We say that B is (α,q,t)-secure if no adversary can distinguish an oracle instantiated as B_k, with random block-cipher key k, from an oracle instantiated as a random L-bit permutation with probability greater than α, given q oracle queries and t computational steps. We assume, for the remainder of the paper that every adversary is limited to no more than t steps. Using B instead of g in WC-AE is accomplished by defining the key set K of WC-AE to be the set of all block cipher B keys and replacing all occurrences of g with B_k. We call this version WC-AE[B]. An advantage WC-AE[B] has over other AEAD schemes is its use of a single block-cipher key for both authentication and encryption. As one can see in the definition and proof of WC-AE, a single function is carefully used for both authentication and encryption, ensuring that g never is computing on the same input twice. When we move from using a random function g to a block cipher, this careful avoidance of repeated inputs allows for the use of a single block-cipher key.

Proposition: Adv(WC-AE[B], PRIV) $\leq ((1 - q/2^L)^{-q/2} - 1) + \alpha$ and Adv(WC-AE[B], AUTH) $\leq \varepsilon(1 - q/2^L)^{-q/2} + \alpha$ when all nonces begin with a zero bit and B is invoked no more than q times.

The term $(1 - q/2^L)^{-q/2}$ comes from the perceptible difference between a random L-bit function and random L-bit permutation over q points [2]. If an adversary existed that achieved greater than either advantage in the proposition, standard reduction techniques would allow us to construct an adversary that could distinguish between B_k (for random k) and a random permutation with greater than α probability using q queries.

As an example, consider the use of WC-AE[AES] to encrypt and authenticate some combination of messages requiring 2^{50} block-cipher invocations. Then Adv(WC-AE[AES], PRIV) $< 1/2^{28} + \alpha$ and Adv(WC-AE[AES], AUTH) $< \varepsilon(1 + 1/2^{28}) + \alpha$ where α represents the maximum probability AES under a random key can be distinguished from a random permutation over 2^{50} invocations. Since ε and α are typically very small (think $1/2^{64}$ or smaller), this is significant security over so many AES invocations. If fewer block-cipher invocations are needed, say 2^{30}, then Adv(WC-AE[AES], PRIV) $< 1/2^{68} + \alpha$ and Adv(WC-AE[AES], AUTH) $< \varepsilon(1 + 1/2^{68}) + \alpha$.

IV. VMAC-AE, UMAC–AE

Highly efficient realizations of WC-AE can be made using VHASH and UHASH, the hash functions of VMAC and UMAC [4,5]. UMAC was developed as a Wegman-Carter MAC with exceptional speed on processors that multiply 32-bit operands efficiently, while VMAC was later developed following the same principles as UMAC, but focused on 64-bit architectures. VHASH achieves ε values as low as $1/2^{59.9}$ and $1/2^{118}$ using 0.5 and 1.0 cpb, respectively. UHASH achieves ε values of about $1/2^{30i}$ using $i/2$ cpb on both 32- and 64-bit architectures (depending on one's choice of $1 \leq i \leq 4$). Additional information and implementations are found at fastcrypto.org [4].

To compare performance of VMAC-AE and UMAC-AE with other authenticated encryption schemes, a commonly cited public implementation of each was used. Gladman's implementations were used for OMAC, CCM, CWC and EAX, and a reference implementation of OCB was retrieved from the OCB author's website. All implementations are written in C with OCB, UMAC-AE and VMAC-AE using small amounts of inline assembly. Implementations use Gladman's AES assembly code and a similar test setup. Tests were run on two processor architectures: A 2GHz AMD Athlon 64 "Manchester" in 64-bit mode and a 2.8 GHz Intel Xeon "Nacona" in 32-bit mode. The examination intends only to give a sense of relative performance.

TABLE I. PERFORMANCE ON TWO ARCHITECHTURES

	64-bit Athlon 64			32-bit Pentium 4		
	64B	***256B***	***2KB***	***64B***	***256B***	***2KB***
CTR	11.9	11.9	11.9	21.6	21.6	21.4
OMAC	23.8	16.7	14.3	36.6	25.8	22.3
CCM	38.2	28.3	25.0	74.9	54.9	48.5
CWC	52.4	41.1	37.4	106*	79*	65*
EAX	41.7	28.9	24.7	76.6	52.4	44.5
GCM	51.3	38.2	34.4	106.5	82.0	74.5
OCB	21.5	15.8	13.9	46.6	32.5	28.1
UMAC-AE-64	22.6	15.8	13.7	**41.6**	**27.5**	**23.3**
UMAC-AE-128	26.8	17.6	14.9	52.4	30.0	25.0
VMAC-AE-64	**17.9**	**14.0**	**12.8**	52.0	36.6	29.1
VMAC-AE-128	19.7	14.9	13.1	58.7	46.6	36.6

Table I shows performances of the various algorithms over short, medium and long message lengths using AES with 128-bit keys as the block cipher. For comparison, CTR-mode encryption and OMAC authentication (a NIST-approved block-cipher based CBC-MAC variant) are listed. All timings are generated using GCC 4.0 under similar conditions except (*) which is taken from Gladman's AES webpage [3].

Table II shows memory and code sizes on Athlon 64 using GCC 4.0. Memory is per encryption key and determined by the C sizeof function. Code size is the sum of the algorithm specific object files generated by GCC, after executing gnu strip –s (sum excludes the AES code).

One solution to authenticating encryption is to encrypt a message and authenticate the ciphertext, using separate keys for each operation. Such a solution using CTR and OMAC would perform approximately at the rate of the sum of the rates of the two algorithms, but at the cost of managing two separate keys. CCM and EAX do away with the need for two keys, but without any speed improvement. OCB integrates authentication operations into the encryption process very efficiently, at a cost slightly higher than encryption alone. The remaining algorithms in the table all encrypt in CTR mode and apply a Wegman-Carter scheme for authentication. Those using the fastest hash functions come out on top—VMAC-AE and UMAC-AE—at roughly the same speeds as OCB.

V. CONCLUSION AND FUTURE WORK

The schemes presented here represent the fastest patent-free AEAD schemes currently known to the author. The schemes, however, are tailored to specific architectures with fast multipliers. This makes them appropriate for computational environments from laptops to servers and workstations, but less so for constrained environments such as cell phones, PDAs and inexpensive networking hardware. Also, custom hardware becomes much more expensive in terms of latency and die area when large multiplications are required. Future work could investigate the use of smaller moduli for multiplication, perhaps as little as just a few bits, and increasing parallelism. At the practical level, implementations could be developed that integrate VHASH and UHASH calculations more closely, reducing the register-to-memory overhead that a loosely coupled implementation may have.

TABLE II. MEMORY REQUIRMENTS

	Memory per key (bytes)	Code size (kilobytes)
CTR	248	—
OMAC	272	2.5
CCM	360	6.7
CWC	424	5.7
EAX	384	5.5
GCM	8552	13.0
OCB	516	4.6
UMAC-AE-64	1552	11.3
UMAC-AE-128	1704	11.8
VMAC-AE-64	624	7.2
VMAC-AE-128	608	8.1

REFERENCES

[1] Bellare M, Desai A, Jokipii E, Rogaway P. A concrete security treatment of symmetric encryption. In Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997.

[2] Bernstein D. Stronger security bounds for Wegman-Carter-Shoup authenticators. In Advances in Cryptology – EUROCRYPT 2005. Springer-Verlag, 2005.

[3] Gladman B. AES and Combined Encryption/Authentication Modes. Webpage: http://fp.gladman.plus.com/AES/.

[4] Krovetz T. Fast cryptography. Webpage: http://fastcrypto.org/.

[5] Krovetz T. Message authentication on 64-bit architectures. In Selected Areas in Cryptography: 13th International Workshop, SAC 2006. Springer-Verlag, 2006.

[6] NIST. Modes of operation. Webpage: http://www.nist.gov/modes/.

[7] Rogaway P. Authenticated-encryption with associated-data. In ACM Conference on Computer and Communications Security 2002 (CCS'02), ACM Press, 2002.

[8] Stinson D. Universal hashing and authentication codes. Designs, Codes and Cryptography 4, 1994.

[9] Wegman M, Carter L. New hash functions and their use in authentication and set equality. J. of Computer and System Sciences, 1979