GCM, GHASH and Weak Keys

Markku-Juhani O. Saarinen

REVERE SECURITY
4500 Westgrove Drive, Suite 335, Addison, TX 75001, USA.
mjos@reveresecurity.com

Abstract. The Galois/Counter Mode (GCM) of operation has been standardized by NIST to provide single-pass authenticated encryption. The GHASH authentication component of GCM belongs to a class of Wegman-Carter polynomial universal hashes that operate in the field $GF(2^{128})$. GCM uses the same block cipher key K to both encrypt data and to derive the generator H of the authentication polynomial. In present literature, only the trivial weak key $H = 0$ has been considered. In this note we show that GHASH has much wider classes of weak keys in its 512 multiplicative subgroups, analyze some of their properties, and give experimental results when GCM is used with the AES algorithm.

Keywords: Cryptanalysis, Galois/Counter Mode, AES-GCM, Weak Keys.

1 Introduction

Authenticated encryption modes and algorithms provide confidentiality and integrity protection in a single processing step. This results in performance and cost advantages as data paths can be shared.

The Galois/Counter Mode (GCM) has been standardized by NIST [9] to be used in conjunction with a 128-bit block cipher for providing authenticated encryption functionality. When paired with the AES [10] algorithm, the resulting AES-GCM combination has been used as a replacement to dedicated hash-based HMAC [1] in popular cryptographic protocols such as SSH [4], IPSec [6] and TLS [12].

In AES-GCM, data is encrypted using the Counter Mode (CTR). A single AES key K is used to both encrypt data and to derive authentication secrets. The component that is used by GCM to produce a message authentication code is called GHASH. GCM also supports Additional Authenticated Data (AAD) which is authenticated using GHASH but transmitted as plaintext.

The GHASH algorithm belongs to a widely studied class of Wegman-Carter polynomial universal hashes. The security bounds known for these algorithms indicate that a n-bit tag will give $2^{-\frac{n}{2}}$ security against forgery [2, 13].

In this paper we give further evidence that this is not only the security lower bound but an upper bound as well. It can be argued that universal hashes sacrifice some communication bandwidth for convenience as traditional hash-based MACs are designed to reach the information theoretic 2^{-n} bound against message forgery.

This paper is structured as follows. We give a description of GHASH in Section 2, followed by a key observation regarding collisions derived from cycles in Section 3. Section 4 contains an analysis of cycle lengths and group orders. In Section 5 we discuss the probability of successful forgery. Section 6 contains a test and experimental results related to cycle lengths. In Section 7 we analyze the impact of the attacks described in this paper, followed by conclusions in Section 8.

2 Description of GHASH

Let X be a concatenation of unencrypted authenticated data, CTR-encrypted ciphertext, and padding. This data is split into m 128-bit blocks X_i:

$$X = X_1 \mid\mid X_2 \mid\mid \cdots \mid\mid X_m.$$

AES is used to derive the root authentication key $H = E_K(0)$. The same AES key K is also used as the data encryption key. In the present work we assume that H is unknown to the attacker as the scheme would be otherwise trivially breakable.

GHASH is based on operations in the finite field $GF(2^{128})$. Horner's rule is used in this field to evaluate the polynomial Y.

$$Y_m = \sum_{i=1}^{m} X_i \times H^{m-i+1}. \tag{1}$$

The authentication tag is $T = Y_m + E_K(IV \mid\mid 0^{31} \mid\mid 1)$, assuming that a 96-bit Initialization Vector (IV) is used. The IV value must never be repeated as that would lead to the "forbidden attack" discussed by Joux in [5].

3 Collisions from Weak Keys

It has been observed that if $E_K(0) = H = 0$, the polynomial Y evaluates to zero and the security of GHASH breaks down. In fact, some sources assume that this pathological case is the only weak key [3]. AES keys K that produce

this fixed point are not known.[1] However, It is easy to see why such keys should exist for AES, especially when the size of K is more than 128 bits.

Our main observation is that sometimes the powers of H will repeat in a relatively short cycle. A trivial example occurs when H is equal to the identity element 1, which will lead to all powers being equal. Due to the commutativity of addition in Equation 1, a GHASH collision can be achieved by swapping any two ciphertext blocks X_i and X_j. This amounts to message forgery.

More generally, if we know that $H^{m-i+1} = H^{m-j+1}$ with $i \neq j$, we may simply swap X_i and X_j and the resulting authentication tag stays the same, as can be easily observed from Equation 1. From number theory we know that the powers of H will repeat in cycles which are determined by $n = \mathrm{ord}(H)$, the multiplicative order of H. Hence we may produce collisions by swapping X_i and X_{i+nm} for arbitrary i and m.

4 Cycle Lengths and Group Orders

From Lagrange's theorem in group theory we know that all subgroups divide the group of order $2^{128} - 1$. Numbers of this type factor into Fermat numbers

$$2^{2^n} - 1 = \prod_{i=1}^{n} 2^{2^{i-1}} + 1. \tag{2}$$

We can easily obtain the full factorization of $2^{128} - 1$:

$$3 \times 5 \times 17 \times 257 \times 641 \times 65537 \times 274177 \times 6700417 \times 67280421310721. \tag{3}$$

As this is a "smooth number", we can see that there are classes of H and therefore K values that produce cycles of length $n = 1, 3, 5, 15, 17, 51, \ldots$; any one of the $2^9 = 512$ combinations the primes in Equation 3 is a valid group order.[2]

We will illustrate this with few trivial examples. Due to the way finite field arithmetic is defined in the GCM standard [9], the identity element with $\mathrm{ord}(H) = 1$ is:

```
H = 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Apparently this was considered as the "first bit" by those who originally implemented GCM. Otherwise standard polynomial arithmetic is used with the field representation defined by the reducing polynomial $x^{128} + x^7 + x^2 + x + 1$.

The following two elements will produce a cycle of length $\mathrm{ord}(H) = 3$ (the cycle obviously goes through the identity as well):

[1] Some block ciphers such as GOST allow such fixed-point keys to be very easily found.

[2] The term *smooth number* comes from factorization theory and indicates that a number factors into a large number of small primes.

```
H = 10 D0 4D 25 F9 35 56 E6 9F 58 CE 2F 8D 03 5A 94
H = 90 D0 4D 25 F9 35 56 E6 9F 58 CE 2F 8D 03 5A 94
```

And these four elements have $\mathrm{ord}(H) = 5$:

```
H = 46 36 BD BD 1C 76 43 D3 4E E4 BB 1B F9 CA 08 4F
H = 92 17 8D 40 26 DA 1D CA 42 96 77 87 30 EB 9A 9E
H = 82 C7 C0 65 DF EF 4B 2C DD CE B9 A8 BD E8 C0 0A
H = D6 E6 F0 98 E5 43 15 35 D1 BC 75 34 74 C9 52 DB
```

We do not know which actual AES keys produce these H values, nor do we recommend testing against these particular values as the probability of hitting them is exceedingly small.

5 Message Forgery

We know that the field $GF(2^{128})$ offers a generous serving of $2^9 = 512$ different multiplicative subgroups. Figure 1 shows that these are quite evenly distributed in the range due to the super-exponential progression of the factors.

In our attack the adversary does not know H but will simply attempt blind forgery by swapping two (or more) message blocks in transit as discussed in Section 3.

It is easy to show that it is sufficient that the group order divides the distance between swapped elements. Since each subgroup of size n has exactly n elements, we arrive at the following observation:

Theorem 1. *Let n be a number satisfying $\gcd(2^{128} - 1, n) = n$. Blindly swapping blocks X_i and X_j, where $i \equiv j \pmod{n}$ will result in a successful forgery with probability of at least $\frac{n}{2^{128}}$.*

Proof. The distance congruence implies that the distance between X_i and X_j is a multiple of n. The $\gcd(2^{128} - 1, n) = n$ condition implies that n is one of the $2^9 = 512$ possible multiplicative subgroup sizes in $GF(2^{128})$. If indeed $\mathrm{ord}(H) \mid n$ then $H^i = H^j$ and the forgery is successful due to commutativity of equation 1. We observe that the cycles are unique; there are n members in a subgroup of size n and the set of n elements is unique to each subgroup size. Hence the probability of hitting one of these cycle elements is $\frac{n}{2^{128}}$. □

If the condition given in Theorem 1 does not hold, there is no reason to expect that the forgery is successful with a probability higher than $\frac{1}{2^{128}}$.

Assuming that an oracle has indicated a successful message forgery, any number of consecutive forgeries can be produced with probability 1.

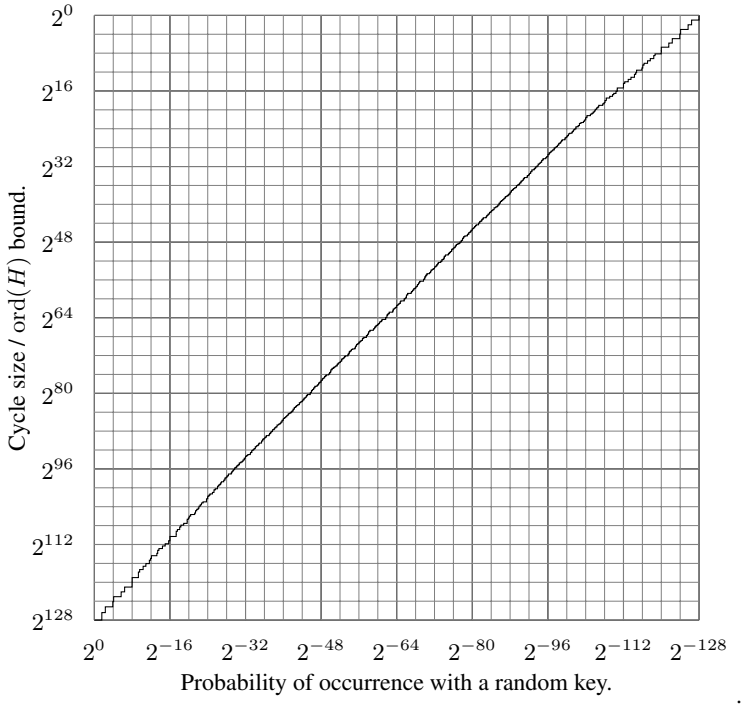

Fig. 1. Probability of hitting a cycle of given size (or smaller) with a random key.

5.1 Bit Forgeries

We note that more elaborate forgeries can be made. If $i - j$ is a multiple of $\mathrm{ord}(H)$ the authentication tag will remain valid as long as the equation

$$X_i \times H^{m-i+1} + X_j \times H^{m-j+1} = c \tag{4}$$

holds for some unknown constant c. Since $H^{m-i+1} = H^{m-j+1} = H_c$, this can be simplified to

$$X_i + X_j = c \times H_c^{-1}. \tag{5}$$

One may therefore flip *individual bits* in block X_i if the corresponding bit in X_j is also flipped. Any number of such modifications can be done to a message without affecting the probability of success (assuming that the same distance is used).

5.2 A Note on Finite Fields

The main observation of the previous sections does not hold for all polynomial hashes as finite fields exist that do not have a smooth multiplicative order.

As an example we could use a prime field $GF(p)$ with a Sophie Germain prime $p = 2^{128} + 12451$. Since $\frac{p-1}{2}$ is also a prime, all but three elements $\{0, 1, p-1\}$ have order in excess of 2^{127}. The swapping attacks described in this paper do not work in this field. Hence a binary field is probably not an ideal choice for authentication algorithms of this type.

6 Testing for Weak Keys

We know that finding weak H values is easy, so a natural question arises on how to determine weak AES keys K that produce these weak H roots.

To determine group order, we use a simple algorithm which is related to the Silver-Pohlig-Hellman algorithm for discrete logarithms [11]. Our algorithm is based on the following elementary observation:

Theorem 2. *Let p be one of the prime divisors given in Equation 3. If and only if p divides $\frac{2^{128}-1}{\mathrm{ord}(H)}$ we have*

$$H^{\frac{2^{128}-1}{p}} = 1. \tag{6}$$

Proof. Trivial. □

By performing the exponentiation test of Theorem 2 for each one of the nine prime divisors of $2^{128} - 1$ in Equation 3, we may completely determine the multiplicative order of H.

We have implemented a reasonably efficient exponentiation algorithm in GCM's $GF(2^{128})$, together with an AES-128 key setup and encryption function for deriving H values from K values. Our implementation is currently able to fully determine the order of 25000 AES keys per second on a Linux laptop that has a single 1.7 gHz AMD V140 processor.

Over couple of days we tested 2^{32} AES-128 keys and found progressively smaller subgroups:

$n \approx 2^{126.4}$ $K =$ `00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02`

$n \approx 2^{125.6}$ $K =$ `00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03`

...

$n \approx 2^{96.52}$ $K =$ `00 00 00 00 00 00 00 00 00 00 00 00 24 3E 8B 40`

$n \approx 2^{96.00}$ $K =$ `00 00 00 00 00 00 00 00 00 00 00 00 37 48 CF CE`

$n \approx 2^{93.93}$ $K =$ `00 00 00 00 00 00 00 00 00 00 00 00 42 87 3C C8`

$n \approx 2^{93.41}$ $K =$ `00 00 00 00 00 00 00 00 00 00 00 00 EC 69 7A A8`

As indicated by Figure 1, a significantly smaller group than $2^{128-32} = 2^{96}$ was found with 2^{32} effort, due to the large number of multiplicative subgroup sizes available in $GF(2^{128})$.

There is clearly room for improvement. The search is fully parallelizable, and hence a massively parallel FPGA or GPU-based search could be performed to find subgroups of magnitude $n \approx 2^{64}$ or less.

7 Risk Analysis for Cryptographic Protocols

The probability of randomly hitting an exploitable weak key with a real-world AES-GCM cryptographic protocol such as SSH [4], IPSec [6] or TLS [12] is acceptably small.

However, malicious players may exploit subtle weaknesses in more complicated cryptographic protocols in surprising ways. One feature of cycle attacks is that an attacker may first test for short cycles and then force a re-keying event if the test fails; once a long-term key with a short cycle is found, she may exploit it any number of times.

It is clear that risks rise quadratically when GCM is used with a 64-bit block cipher as suggested in Appendix A of [8]. There is a substantial risk of hitting a bad long-term key and therefore we recommend against using the 64-bit GCM.

8 Conclusions

We have shown that the GHASH algorithm has other weak key classes besides the trivial $H = 0$ case considered in current literature [3]. This is a result of the multiplicative group of $GF(2^{128})$ having a particularly smooth order. We suggest that Sophie Germain prime fields are used in similar future constructions as this minimizes the total number of weak keys to three ($H \in \{0, 1, p-1\}$).

We have also described a straightforward method of detecting GHASH weak keys. We performed an exhaustive experiment that found many AES-128 keys that produce H with order below $n \approx 2^{96}$.

One interesting future research direction and open question is the feasibility of mapping the weak H values to K symmetric keys with various block ciphers other than AES.

References

1. M. Bellare, R. Canetti and H. Krawczyk: "Keying hash functions for message authentication." CRYPTO '96. LNCS 1109, Springer-Verlag, pp. 1 – 55, 1996.
2. D. J. Bernstein: "Stronger Security Bounds for Wegman-Carter-Shoup Authenticators." EUROCRYPT 2005. LNCS 3494, Springer-Verlag, pp. 164–180, 2005.

3. H. Handschuh and B. Preneel: "Key-Recovery Attacks on Universal Hash Function based MAC Algorithms." CRYPTO 2008. LNCS 5157, Springer-Verlag, pp. 144–161, 2008.
4. K. Igoe and J. Solinas: "AES Galois Counter Mode for the Secure Shell Transport Layer Protocol." IETF Request for Comments 5647, August 2009.
5. A. Joux: "Authentication Failures in NIST version of GCM." NIST Comment, 2006. Available from `http://csrc.nist.gov/CryptoToolkit/modes/`
6. L. Law and J. Solinas: "Suite B Cryptographic Suites for IPsec." IETF Request for Comments 4869, May 2007.
7. D. A. McGrew and J. Viega: "The Security and Performance of the Galois/Counter Mode (GCM) of Operation." INDOCRYPT 2004. LNCS 3348, Springer-Verlag, pp. 343–355, 2004.
8. D. A. McGrew and J. Viega: "The Galois/Counter Mode of Operation (GCM)" Submission to NIST Modes of Operation Process. 2005
9. National Institute of Standards and Technology: "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC." NIST Special Publication 800-38D, November, 2007.
10. National Institute of Standards and Technology: "The Advanced Encryption Standard (AES)." FIPS Publication 197, U.S. DoC/NIST, November 26, 2001.
11. S. Pohlig and M. Hellman: "An Improved Algorithm for Computing Logarithms over $GF(p)$ and its Cryptographic Significance." IEEE Transactions on Information Theory (24): 106–110. 1978.
12. M. Salter, E. Rescorla and R. Housley: "Suite B Profile for Transport Layer Security (TLS)." IETF Request for Comments 5430, March 2009.
13. P. Sarkar: "A trade-off between collision probability and key size in universal hashing using polynomials." Designs, Codes and Cryptography. Vol 58, No 3, Springer-Verlag, pp. 271–278, 2011.

Document version 20120413091900.