SGCM: The Sophie Germain Counter Mode

Markku-Juhani O. Saarinen

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

Abstract. Sophie Germain Counter Mode (SGCM) is an authenticated encryption mode of operation, to be used with 128-bit block ciphers such as AES. SGCM is a variant of the NIST standardized Galois / Counter Mode (GCM) which has been found to be susceptible to weak key / short cycle forgery attacks. The GCM attacks are made possible by its extremely smooth-order multiplicative group which splits into 512 subgroups. Instead of GCM's $GF(2^{128})$, we use $GF(p)$ with $p = 2^{128} + 12451$, where $\frac{p-1}{2}$ is also a prime. SGCM is intended for those who want a concrete, largely technically compatible alternative to GCM. In this memo we give a technical specification of SGCM, together with some elements of its implementation, security and performance analysis. Test vectors are also included.

Keywords: Authenticated Encryption, GCM, Sophie Germain Counter Mode.

1 Introduction

The Galois/Counter Mode (GCM) [8] is a NIST standardized authenticated encryption mode, often used with the AES block cipher. AES-GCM has been proposed as a replacement to HMAC [1] in cryptographic protocols such as SSH [5], IPSec [7] and TLS [10].

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 is a Wegman-Carter polynomial universal hash which has relatively well understood security properties [2, 11]. Despite this, recently it has been shown that there are large classes of weak (AES) keys that make message forgery much easier than what is expected from a MAC function [12].

Section 2 describes GHASH and it's vulnerability with GCM. Section 3 describes the SGCM, followed by some aspects of its implementation in Section 4. We conclude in Section 5. Appendix A contains an example of full SGCM computation.

2 GHASH and the Cycle Swapping Attack

We will first give a concise description of the GHASH component of GCM defined in [8], followed by a discussion of the attacks described in [12].

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

$$X = X_1 \mid\mid X_2 \mid\mid \cdots \mid\mid X_n. \tag{1}$$

A 128-bit block cipher such as AES is used to derive the hash subkey $H = E_K(0)$. The same AES key K is also used as the data encryption key. GHASH is based on arithmetic operations in a finite field. Horner's rule is used to evaluate the polynomial Y, given m 128-bit message blocks X_i with padding.

$$Y_m = \sum_{i=1}^{m} X_i \times H^{m-i+1}. \tag{2}$$

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. Other options exist.

The attack described in [12] is based on the observation that powers of H sometimes repeat in a short cycle when the the arithmetic of Equation 2 is performed in $GF(2^{128})$. 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. The powers of H repeat in cycles which are determined by $n = \mathrm{ord}(H)$, the multiplicative order of H. We may therefore produce collisions by swapping any two ciphertext blocks X_i and X_j if $i \equiv j \mod n$. Note that this swapping attack can be also applied to any number of individual pairs of bits in corresponding positions of blocks separated by n positions or its multiple.

Lagrange's theorem in group theory tells us that each multiplicative subgroup order divides the main multiplicative group order, which for GCM is $2^{128}-1 = 3\times5\times17\times257\times641\times65537\times274177\times6700417\times67280421310721$. As there are 9 prime factors, there is a unique subgroup for each one of the $2^9 = 512$ different subsets of these primes. We may use this observation to increase the probability of successful message forgery.

Let n be a number satisfying $\gcd(2^{128}-1, n) = n$. There are 512 different options for n, ranging almost log-uniformly in the 128-bit range. Blindly swapping X_i and X_j, where $i \equiv j \pmod n$ will result in a successful forgery with probability of at least $\frac{n}{2^{128}}$, rather than the expected $\frac{1}{2^{128}}$.

To illustrate this, consider a protocol which exchanges messages that are larger than 1M (65537 blocks). It then has roughly 112-bit security in its randomly keyed 128-bit GCM MAC against a blind ciphertext block swap with offset 65537.

3 The Sophie Germain Counter Mode SGCM

Mathematically SGCM differs from GCM inly in the underlying field where GHASH's arithmetic operations are performed. While GCM uses the binary field $GF(2^{128})$, SGCM uses traditional modular arithmetic in $GF(p)$, where

$$p = 2^{128} + 12451 = 340282366920938463463374607431768223907. \quad (3)$$

Here $\frac{p-1}{2}$ is also a prime, a Sophie Germain prime. [1]

3.1 Technical Specification of SGCM

All other aspects of SGCM are equivalent to GCM, apart those described in Sections 6.3 "Multiplication Operation on Blocks" and 6.4 "GHASH Function" of NIST Special Publication 800-38D [8].

The bytes of 128-bit blocks of data are accessed in little-endian fashion. We give an example of computing the product of two elements in $GF(p)$ and the arrangement of the bytes in computer memory:

$$\begin{aligned} X = \lfloor 2^{126}\pi \rfloor &= 267257146016241686964920093290467695825 \\ X &= \texttt{D1 1C DC 80 8B 62 C6 C4 34 C2 68 21 A2 DA 0F C9} \\ Y = \lfloor 2^{126}e \rfloor &= 231245843636555084287727758960834198769 \\ Y &= \texttt{F1 3C 3D 27 20 56 DC AF 9A 4A BB A2 58 54 F8 AD} \\ Z = XY \bmod p &= 92057282056387974665238950822035710352 \\ Z &= \texttt{90 E1 BD 2C 96 07 A3 63 19 D9 D9 AE 6D 96 41 45} \end{aligned}$$

Now let X_i denote the sequence of blocks as defined in Equation 1 and let $H = E_K(0) + 2$ be the hash subkey. We start with $Y_0 = 0$ and iterate for $i = 1, \ldots, n$ the following:

$$Y_i = (Y_{i-1} + X_i)\, H \mod p. \quad (4)$$

The final iteration satisfies $Y_n = SGHASH_H(X)$. Should the value be equal to 2^{128} or larger and hence require more than 16 bytes of storage, the result should be truncated $\bmod 2^{128}$. This special case is exceedingly rare ($P \approx 2^{-114.396}$). This value is then used in equal fashion as $GHASH_H(X)$ is used in the GCM specification.

Appendix A contains an example of a full SGCM computation.

[1] Primes of this type are named after the French mathematician Marie-Sophie Germain (April 1, 1776 – June 27, 1831) who used these strong primes in her investigations.

3.2 SGCM Cycle Properties

We will now characterize the cycle properties of SGCM. Here $\left(\frac{x}{p}\right)$ is the Legendre symbol. From elementary number theory we know that the multiplicative order of a $GF(p)$ element $H > 0$ always satisfies one of the following four cases when $\frac{p-1}{2}$ is also a prime:

A. If $H = 1$ then $\mathrm{ord}(H) = 1$.
B. If $H = p - 1$ then $\mathrm{ord}(H) = 2$.
C. If $1 < H < p - 1$ and $\left(\frac{H}{p}\right) = 1$ then $\mathrm{ord}(H) = \frac{p-1}{2}$.
D. If $1 < H < p - 1$ and $\left(\frac{H}{p}\right) = -1$ then $\mathrm{ord}(H) = p - 1$.

Due to the start point rule $H = E_K(0) + 2$, which puts H in the range $2 \leq H < 2^{128} + 2$, we may dismiss cases A, B, and the pathological case $H = 0$.

One can compare this behavior to that of GCM with a n-bit key, which has roughly 2^{n-96} weak keys that produce cycles shorter than 2^{32} blocks. SGCM cycles are always about 2^{127}.

3.3 Resistance to Other Attacks

In [3] Ferguson notes, using a more complicated technique, that an n-bit GCM tag provides only n - k bits of authentication security when messages are 2^k blocks long. His attacks were based on bitwise linearity of constant multiplication and squaring in $GF(2^{128})$, and hence these attacks do not apply to SGCM.

Joux [6] discusses chosen-IV attacks against GCM. SGCM is susceptible to these attacks and the IVs should never be repeated.

4 Implementation

SGCM may be implemented in software either by using large tables derived from the H value (a common method for GCM) or by using regular integer multiplication instructions. Table-based implementations have similar structure as those for GCM, with the obvious difference that XOR operations are replaced with ADD operations and carry propagation and overflow logic must be implemented.

Division-free modular reduction can be achieved by noting that

$$2^{128}x \equiv -12451x \pmod{p}. \tag{5}$$

Multiplication of the 128-bit overflow with this small negative constant may also be tabulated (this table is not dependent on the H value).

Some Intel processors support the PCLMULQDQ "carry-less multiplication" instruction. This instruction was apparently included in response to GCM performance issues [4]. We note that SGCM does not require special instructions to achieve similar speed.

Overall, we expect SGCM to have similar or superior performance to GCM on most software platforms. Hardware implementations of SGCM may require a somewhat larger gate count.

5 Conclusions

We have described the Sophie Germain Counter Mode (SGCM), which is a plug-in compatible variant of the Galois/Counter Mode (GCM) in terms of data paths and implementation logic. We have shown that SGCM is resistant to weak key / short-cycle attacks and has similar, or better performance features to GCM. We encourage the cryptographic community to study and comment on SGCM.

References

1. M. Bellare, R. Canetti and H. Krawczyk, Keying hash functions for message authentication, Proc. CRYPTO '96, LNCS 1109 (1996) 1–55.
2. D. J. Bernstein, "Stronger Security Bounds for Wegman-Carter-Shoup Authenticators." Proc. EUROCRYPT 2005, LNCS 3494 (2005) 164–180.
3. N. Ferguson, Authentication weaknesses in GCM, Available from `http://csrc.nist.gov/CryptoToolkit/modes/`, Official NIST Comment (2005).
4. Intel, Intel Carry-Less Multiplication Instruction and its Usage for Computing the GCM Mode, White Paper, (2010).
5. K. Igoe and J. Solinas, AES Galois Counter Mode for the Secure Shell Transport Layer Protocol, IETF Request for Comments 5647 (2009).
6. A. Joux, Authentication Failures in NIST version of GCM, Official NIST Comment (2006).
7. L. Law and J. Solinas, Suite B Cryptographic Suites for IPsec, IETF Request for Comments 4869 (2007).
8. NIST, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, NIST Special Publication 800-38D (2007).
9. NIST, The Advanced Encryption Standard (AES), FIPS Publication 197 (2001).
10. M. Salter, E. Rescorla and R. Housley: "Suite B Profile for Transport Layer Security (TLS)." IETF Request for Comments 5430 (2009).
11. P. Sarkar, A trade-off between collision probability and key size in universal hashing using polynomials, Designs, Codes and Cryptography. Vol 58, No 3, (2011) 271–278.
12. M.-J. O. Saarinen, GCM, GHASH and Weak Keys, Submitted for publication, IACR ePrint 2011/202 (2011).

A Example of Full SGCM Computation

In this example we will encrypt 48 bytes (an increasing byte sequence) using AES-SGCM with a 128-bit key and a 96-bit IV. Three rounds and a finalization round is required. The authentication tag can be found at the end.

```
      KEY  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
       IV  10 11 12 13 14 15 16 17 18 19 1A 1B
  (+2) H   C8 A1 3B 37 87 8F 5B 82 6F 4F 81 62 A1 C8 D8 79

i = 1
       PT  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
       CB  10 11 12 13 14 15 16 17 18 19 1A 1B 00 00 00 02
  CT & X   C4 2F 01 AC 0B 4A B0 E8 1F D4 57 FE CB 2A E5 31
        Y  6C 33 4B FF 88 81 60 66 2B C9 D5 5A D6 2E 15 AB

i = 2
       PT  10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
       CB  10 11 12 13 14 15 16 17 18 19 1A 1B 00 00 00 03
  CT & X   2A AD 66 94 22 E1 7D A8 9D D2 33 0A 7B 18 0F B2
        Y  64 EE 69 40 EF 74 DC 6E 34 E2 C8 1F B5 17 C0 F4

i = 3
       PT  20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
       CB  10 11 12 13 14 15 16 17 18 19 1A 1B 00 00 00 04
  CT & X   F2 F8 03 1C A5 83 DD 3B CB 89 FF E3 F6 FD 7F 34
        Y  86 64 80 7B 55 0F 65 96 1E D6 CF C5 CD E1 17 CE

Fin
        X  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 80
        Y  05 89 56 EC 77 7A B2 1A 17 76 29 17 56 DF 8C D1
       CB  10 11 12 13 14 15 16 17 18 19 1A 1B 00 00 00 01
      TAG  0B 5E 73 76 AA 6A A3 FB 4E A6 27 76 E7 4D D8 C1
```

We note that with the exactly same input data, AES-GCM will produce the following authentication tag:

```
      TAG  BB 50 08 DB A5 F7 4C E1 6F BC 92 5F 78 C7 45 76
```

Document version 20111104192300.