Comparative Study of RKC and GCM Mode of Operation

Puneet Kumar Kaushal
Lovely Professional University
Jalandhar Delhi G.T. Road
Phagwara, Punjab (India)

Rajeev Sobti
Lovely Professional University
Jalandhar Delhi G.T. Road
Phagwara, Punjab (India)

Ranjana Singh Rathore
Lovely Professional University
Jalandhar Delhi G.T. Road
Phagwara, Punjab (India)

ABSTRACT

In the race of designing an efficient ‘Authenticated Encryption (AE) mode of operation’ several efforts have been done over past few years. In practical, when we have to send any data, we also need integrity and authentication along with the encrypted message. These requirements in real world prioritize *Authenticated Encryption*, in spite of using encryption and authentication schemes separately. In an AE scheme the parts responsible for authentication and privacy are tightly coupled so that they can work efficiently. The other reason for using AE scheme is that an AE scheme is less likely to be used incorrectly than the combination of two different schemes. In this paper comparative study of two prominent Authenticated Encryption modes (RKC and GCM) is done that are free from any intellectual property claims.

General Terms

Security, Authenticated Encryption modes.

Keywords

Random Key Chainingmode (RKC), Galois/Counter mode(GCM), Comparison, Authenticated encryption mode.

1. INTRODUCTION

Among authenticated encryption modes, OCB [1] comes out to be the fastest mode of operation [2] [3]. But OCB is covered with intellectual property claims. So, the second most efficient, **Galois Counter Mode**[4], and recently emerged **Random Key Chaining Mode**[5] is chosen for the analysis. Both GCM and RKC are nonce based, meaning that, on encryption and decryption a nonce value is required. In RKC nonce is termed as Seed. In GCM while encryption, nonce must be changed with each message however there is no such need in RKC. Both AE scheme takes as input key, nonce and plaintext and in GCM one extra parameter called associated data (AD) is required. AD is authenticity parameter protected by the ciphertext. It is neither included as a part in ciphertext nor can be recovered from it.

GCM has many favorable characteristics. It has provable security guarantee provided the tags are sufficiently long, has found its way in TLS [6] IPsec [7] and MACSec [8]. GCM is also optimized on Intel processors by Vinodh Gopal et al. and achieved performance of 2.8 cycles/byte [9]. However Furguson [10] described two weaknesses in the authentication functionality of GCM when it is used with a short authentication tag. The first weakness was construction of targeted cipher text forgery and second weakness reveals authentication key if attacker manages to create successful forgeries [11]. But these issues are not fatal. RKC is a new candidate in this field. It is simple to understand and implement. Not so much research has been done with it. This paper presents the first software implementation of RKC. It makes use of Deterministic Random Bit Generator (DRBG) along with secret key for generating keystream and makes use of SHA-256 for authentication.

2. MODE SPECIFICATION

GCM and RKC modes are briefly described below:

2.1 Galois/Counter Mode

Following NIST SP 800-38D [12], the underlying blockcipher in GCM must be AES, because the blockcipher must be NIST-approved and must have 128-bit block size. The other NIST approved block-ciphers operate on 64-bit blocks (like TDEA, Skipjack).

Let us consider plaintext sequence is $P_1, P_2 \ldots P_{n-1}, P_n^*$ciphertext sequence is$C_1, C_2 \ldots C_{n-1}, C_n^*$.

Where, | P_i | = | C_i | = 128 bits, P^*_n& C^*_n are the last block of plaintext and ciphertext respectively.Let say | P^*_n | = u, and total no. of plaintext blocks = n, then plaintext can be represented as:

$$(n-1)128 + u \text{, where } 1 \le u \le 128$$

Additional authentication data AAD is denoted as A_1, A_2… A_{m-1}, A^*_m

Where, | A_i | = 128 bits, A^*_m is the last block of AAD. Let say | A^*_m | = v, and total no. of blocks in AAD = m, then AAD can be represented as:

$$(m-1)128 + v\text{, and } 1 \le v \le 128$$

The authenticated encryption operation is:

$$H = E(K, 0^{128})$$

$$Y_0 = (IV||0^{31}1)\text{, if len (IV) = 96}$$

$$\text{Otherwise,} Y_0 = GHASH(H, \{\}, IV)$$

$$Y_i = incr(Y_{i-1})\text{for i = 1… n}$$

$$C_i = P_i \oplus E(K, Y_i)\text{, for i = 1. . . n−1}$$

$$C_n^* = P_n^* \oplus MSB_u(E(K, Y_n))$$

$$T = MSB_t(GHASH(H, A, C) \oplus E(K, Y_0))$$

incr() function is used to generate counter values. It takes rightmost 32 bits of its argument with least significant bit on right and increment it modulo 2^{32}.

Function GHASH is defined by $GHASH(H, A, C) = X_{m+n+1}$. The variable X_i for i = 0… m + n + 1 are defined as:

$X_i = 0$,for i = 0

$X_i = (X_i - 1 \oplus A_i).H$, for i = 1. . . m−1

$X_i = X_{m-1} \oplus A_m^*|| \, 0^{128-v})).H$, for i = m

$X_i = (X_{i-1} \oplus C_{i-m}).H$, for i = m +1. . . m+n−1

$X_i = X_{m+n-1} \oplus (C_n^*|| \, 0^{128-u})).H$, for i = m+n

$X_i = X_{m+n} \oplus (len(A)||len(C))).H$, for i=m+n+1

Multiplication in GF (2128) is performed as described below:

Z = X.Y, where X, Y and Z ∈ GF (2^{128})

```
$Z \leftarrow 0, V \rightarrow X$

For i = 0 to 127 do
        if $Y_i = 1$ , then
        $Z \leftarrow Z \oplus V$
        end if
        if $V_{127} = 0$,then
            $V \leftarrow rightshift(V)$
        else
        $V \leftarrow rightshift\,(V \oplus R)$
        end if
end For
return Z
```

The authenticated decryption operation is:

$H = E(K, 0^{128})$

$Y_0 = (IV||0^{31}1)$, if len (IV) = 96

Otherwise,$Y_0 = GHASH(H, \{\}, IV)$

$T' = MSB_t(GHASH(H, A, C) \oplus E(K, Y_0))$

$Y_i = incr(Y_{i-1})$for i = 1… n

$P_i = C_i \oplus E(K, Y_i)$, for i = 1. . . n

$P_n^* = C_n^* \oplus MSB_u(E(K, Y_n))$

$T = MSB_t(GHASH(H, A, C) \oplus E(K, Y_0))$

Tag T' is computed by decryption operation and compared with tag T that arrives with ciphertext. If both the tag matches in length and value then the plaintext is returned.

2.2 Random Key Chaining Mode

RKC is defined to use only with AES-256 [5] as the underlying blockcipher. Deterministic Random Bit Generator (DRBG) function is used for generating pseudorandom number. RKC makes use of Hash_DRBG as specified in Section 10.1.1 of NIST SP800-90 [13]. SHA-256[14] is used as underlying hash-function in Hash_DRBG because word size in SHA-256 is 32-bit and RKC is optimized for 32-bit architecture.

For encryption inputs are:

- An Encryption key K_0, 256-bits
- Seed S = 440-bits (for instantiating DRBG)
- Message M, where M = 128(n-1) + y bits, Where n = total number of blocks of size 128-bits and y = number of bits in last block of message.

Initially,

$K_1 = K_0 \oplus R_1$, where K_0 is shared secret key between sender and receiver.

The encryption keystream is calculated as:

$K_i = K_{i-1} \oplus R_i$, For i = 1… n

After the keystream is generated (pre-processed), the encryption can be executed in parallel.

$C_i = E(K_i, P_i)$

X_i as computed below is used to generate authentication tag.

$X_i = C_i \oplus P_i$

At the receiving end, X_i is calculated as shown above after decrypting the message. The decryption process is:

$P_i = D(K_i, C_i)$

Authentication Tag at both the end is calculated as:

T = $HASH(X_1 \,||\, X_2 \,||\, \ldots\ldots\ldots\, X_n\,)$

For calculating authentication tag, parallelization is feasible up to some extent. Calculation of X_i can be executed in parallel, the only part where authentication data bits are put into hash function can be executed only after X_i is calculated till i=n.

3. SUMMARY OF THE PROPERTIES

Security Strength: Security strength of GCM relies on the random permutation of the underlying block cipher. In RKC $2^{256+440}$ steps are required to perform brute force attack.

Security Function: Authenticated encryption in both the schemes.

Error Propagation: None in both the schemes.

Parallelizability: GCM can have block-level parallelizability while encryption/decryption and bit-level parallelizability while authentication. In RKC encryption and decryption can be executed fully parallel after the pre-processing (keystream generation). Authentication is also parallelizable upto greater extent.

Pre-processing: Keystream can be pre-computed in both the schemes. In GCM fixed parts of IV can also be processed in advance.

Message Length: GCM can have arbitrary message length up to 239-256 bits. RKC can have message length up to 264 bits when authentication is required otherwise there is no restriction on message length. This is because input to hash function in this scheme is the modified ciphertext. Length of plaintext and ciphertext is equal and SHA-256 has input restriction of 264 bits.

Ciphertext Expansion: Ciphertext length is identical to plaintext length in both the schemes.

Key Material: 1 key and 1 IV is required in both schemes. The length of IV (Seed) in RKC is fixed i.e. 440-bits.

Memory Requirement: Memory for plaintext, ciphertext, pre-processed keystream and authentication data is required in both the scheme. So, there is no significant difference in the memory requirements.

4. EXPERIMENTAL RESULTS

Both AE schemes are implemented in C# .net framework 4.0 using windows forms. The AES encryption algorithm used in both the modes are inherited from *System.Security.Cryptography.AesManaged* class (inbuilt in .NET framework). Hash based Deterministic Random Bit Generator has been coded and .net framework inbuilt hash generator class *SHA256Managed* is used for getting hash value. Average time of 5 different plaintexts has been taken. It is assumed under the same environment time required for decryption of ciphertext will be same as that of encrypting plaintext, so only time for encryption is taken for the analysis. In GCM length of IV is taken as 96-bit for getting best results. Results are as shown below:

Table 1. Time taken for the encryption process by RKC and GCM scheme and their differences in milliseconds.

Plaintext Size	RKC	GCM	RKC-GCM
128 byte	10	12.5	-2.5
256 byte	12	17.5	-5.5
512 byte	14	23	-9.0
1 KB	24	31	-7.0
2 KB	52	50.67	+1.33
3 KB	106	65.75	+40.25
4 KB	148.6	81.75	+66.85
5 KB	260.75	132	+128.75
6KB	334.67	170.75	+163.92
7 KB	354.33	173.25	+181.08
8KB	466.75	304.25	+161.75
9KB	600	414.5	+185.5
10KB	659	406.67	+252.33

Figure 1 below shows relative time taken by RKC and GCM.

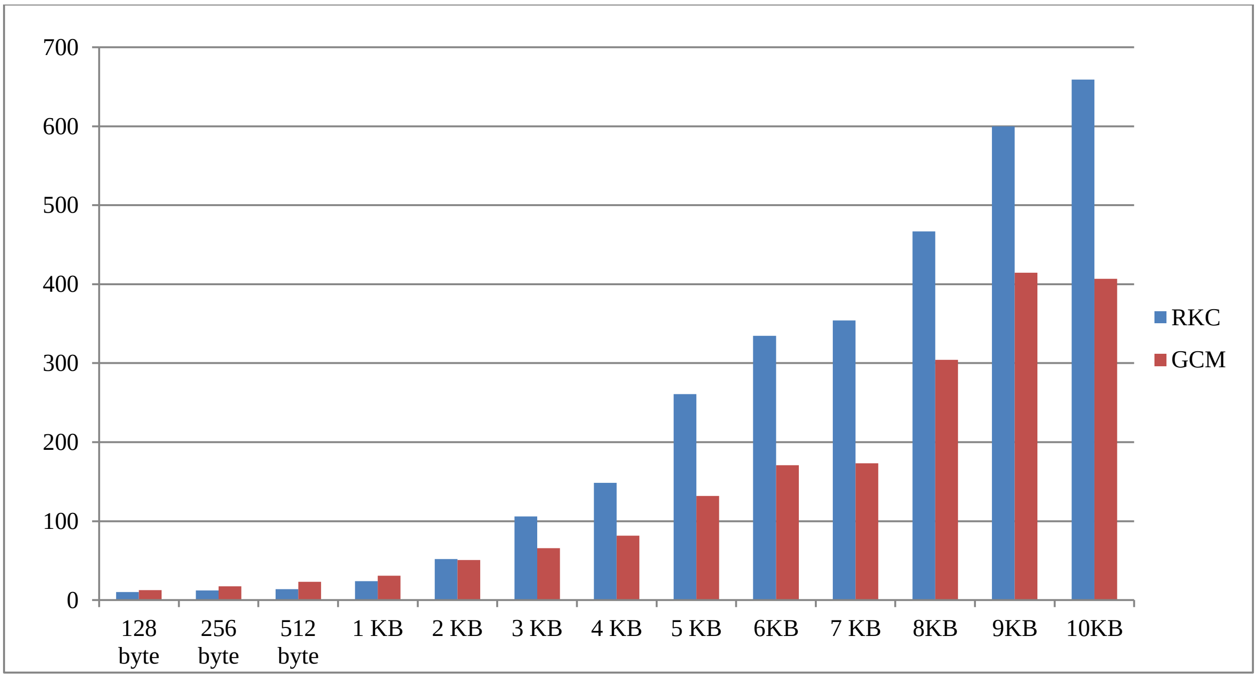

Figure 1: Encryption time taken by RKC and GCM for different plaintext sizes.

5. CONCLUSION

The result shows that RKC is faster than GCM for data up to 1KB in size. The timing difference for 2KB data is also not significant. But from 3KB onwards GCM performs better than RKC (highlighted). So, it can be concluded that when size of transmitting data is less than 2KB RKC is suitable (like in sensor nodes) and for the rest GCM still holds the position. As far as security is concerned,no serious flaw has been found till date against any of the two. However, to be on a safer side GCM should not be used with a shorter authentication tag.

6. ABBREVIATIONS

X || Y: Concatenation of two strings X and Y. Where X and Y are both bit-strings.

X ⊕ Y: Bitwise exclusive-or of two bit-strings X and Y of same length.

E(X, Y): Encryption using AES-256 over bit-string Y using key X. Where X is 256-bits in length and Y is 128-bits in length.

D(X, Y): Decryption using AES-256 over bit-string Y using key X. Where X is 256-bits in length and Y is 128-bits in length.

| P |: Length of string P, represented in number of bits.

7. REFERENCES

[1] Rogaway, P., Bellare, M., Black, J., Krovetz, T., Aug 3, 2001, OCB: A Block Cipher Mode of Operation for Efficient Authenticated Encryption.

[2] McGrew, D. A., Viega, J., June 2005, Galois/CTR Mode of Operation, Table 3, pp. 21.

[3] Krovetz T., Rogaway P., March 21, 2011, the Software Performance of Authenticated Encryption Modes, pp. 9-11.

[4] McGrew, D. A., Viega, J., June 2005, Galois/CTR Mode of Operation.

[5] Kaushal, P., Sobti, R., Geetha, G., Feb 2012, Random Key Chaining (RKC): AES Mode of Operation, International Journal of Applied Information Systems, Volume 1, Number 5.

[6] Salowey, J., Choudhury, A., McGrew, D., August 2008, AES Galois Counter Mode (GCM) Cipher Suites for TLS, RFC 5288.

[7] McGrew, D., Viega, J., May 2006, Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH, McAfee, Inc., RFC 4543.

[8] Institute of Electrical and Electronics Engineers. IEEE Std. 802.1AE-2006, IEEE Standard for local and metropolitan area networks – Media Access Control (MAC) security. IEEE Press, 2006.

[9] Gopal, V., Ozturk, E., Feghali, W., Guilford, J., Wolrich, G., Dixon, M., August 2010, *Optimized Galois-Counter-Mode Implementation on Intel Architecture Processors,*Intel Corporation.

[10] Furguson, N., May 20, 2005, Authentication Weakness in GCM.

[11] Furguson, N., May 20, 2005, Authentication Weakness in GCM, pp. 7-8.

[12] Dworkin, M. NIST Special Publication 800-38D. Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM and GMAC), Nov. 2007.

[13] Barker, E., Kelsey, J., Recommendation for Random Number Generation Using Deterministic Random Bit Generators, March 2007, revised May 2011, NIST Special Publication 800-90, pp. 34-38.

[14] Announcing Secure Hash Standard, Federal Information Processing Standards Publication, FIPS 180-2, Aug 1, 2002, pp. 18-19, 33-40.