International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 1– No.5, February 2012 – www.ijais.org

Random Key Chaining (RKC): AES 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)

Dr. G. Geetha
Lovely Professional University
Jalandhar-Delhi G.T. Road
Phagwara, Punjab (India)

ABSTRACT
There is a compelling need for a mode of operation that can efficiently provide authenticated encryption at a higher data rate, and is capable of making use of pipelining and parallel processing. This paper describes **Random Key Chaining** (**RKC)** block cipher mode of operation that fills this need. RKC mode makes use of **Deterministic Random Bit Generator (DRBG)** andwith the application of DRBG every block of plaintext is being encrypted with a different key bringing it closer to one-time pad approach. The slight variation of RKC mode can be used as a confidentiality mode that can be used in application like hard-disk compression with reduced computational cost.

General Terms
Security, Authenticated Encryption mode.

Keywords
Random Key chaining RKC, Deterministic Random Bit Generator DRBG, Confidentiality mode, Authenticated Encryption Mode of Operation, Advanced Encryption Standard AES.

1. INTRODUCTION
Among the encryption modes, **Counter mode**[1] has emerged as a best method for high speed encryption but provides no protection against bit-flip attacks (also called malleability[2]) because there is no suitable message authentication in this mode. David A. McGrew and John Viega proposed **Galois/Counter Mode of Operation**[3] that combines counter mode of encryption with Galois Mode of authentication and was an improvement to **Carter-Wegman Counter CWC**[4] mode. The key feature in GCM was: the Galois Field multiplication used for authentication can be computed in parallel, thus increasing the throughput in comparison to authentication algorithms that uses chaining mode (like CBC).

GCM comes out to be a good candidate for authenticated-encryption mode of operation and is also implemented in TLS1.2. Vinodh Gopal et al. describes an optimized implementation of GCM that combines functional stitching with novel polynomial multiplication methods, and achieved performance of 2.8 Cycles/byte. This is a new record for GCM performance on Intel processors [5]. However Furguson[6] 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 [7]. **CBC-MAC**[8] and the modes that use it to provide authentication like **OMAC**[9], **EAX**[10] and **CCM**[11] cannot be pipelined or parallelized, and thus cannot provide higher data rates. In CWC mode of operation, message authentication has circuit depth longer because of additional application of block cipher due to additional post processing step[12]. This additional delay would certainly impact its performance on short packets or frames. This leaves behind CWC mode for high speed implementation.Rogaway et al. in his paper[13] described software performance of authenticated-encryption modes (**CCM, GCM** &different version of**OCB**) across variety of platform, and **OCB**[14]was found to be faster than other modes. However,OCB is covered with multiple intellectual property claims.

Random Key Chaining (RKC) block cipher mode is designed to provide both data authenticity and confidentiality. Confidentiality is provided on plaintext, and authentication is provided on plaintext and cipher text both.

RKC mode has additional useful properties. For every block of plaintext, different key is used for encryption. And these keys are obtained with no extra computational cost (since preprocessing can be done for key stream generation). Variation of encryption-keys for each plaintext block plays an important role. Sometimes what happens, if an attacker successfully breaks one of the cipher text blocks, breaking rest of them becomes relatively easy. Variation in encryption-key for every block of encryption makes differential cryptanalysis more difficult.

Differential cryptanalysis is a form of chosen plaintext attack. What makes it differential is that attacker choses a family of plaintexts P_0, P_1, P_2… where each P_i is equal to P_0 plus some small difference. For many differential attacks difference is taken as 1-bit. So, encrypting plaintexts of 'small bit-differences' with different key will result in cipher text blocks with 'large hamming distance' in comparison to cipher text blocks that are generatedusing 'same key' again and again.

Another useful property of RKC is that its slight variation can be used to provide only the confidentiality (*section 7, paragraph 3*), i.e., it can perform as a confidentiality mode efficiently.

2. TERMS & DEFINITIONS
Deterministic Random Bit Generator (DRBG): DRBG produces a pseudorandom sequence of bits from a secret initial value called *seed* with other possible (optional) inputs.

Entropy: Measure of randomness or disorder in a closed system. Entropy of X is a mathematical measure of the amount of information provided by an observation of X.

Hamming Distance: It is the difference between two equal length strings. Its value equals to the number of places at which the two bit-strings differ i.e., have different bits.

Hash Function:A mathematical function that maps values from a large domain into a smaller range known as message

digest or hash value. The function satisfies the following properties:

- It is computationally infeasible to find any input that map to any predefined output.
  (A)
- It is computationally infeasible any two distinct input that map to the same output.
  (B).

Pseudorandom: A process is said to be pseudorandom when the outcome is deterministic, yet also effectively random as long as the internal action of the process is hidden from the observation.

Security Strength: A number associated with the amount of work (no. of operations) required to break a cryptographic algorithm or a system. Security strength i.e. the maximum amount of work required to break the encryption algorithm of x bits key, using the brute force is 2^x. For Hash function with message digest size of x bits, the max amount of work required to break property A (as defined under the heading Hash Function) is 2^x and to break property B is $2^{x/2}$.

Seed: A string of bits that is used as input to a DRBG. The seed determines the internal state of the DRBG and its entropy must be sufficient to support the security strength of DRBG.

Circuit depth: Depth of a circuit is defined as the length of the longest path from the input to the output.

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.

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

3. MODE SPECIFICATION

RKC is generic encryption and authentication block cipher mode. In this paper it is defined with AES encryption algorithm. RKC is defined for only use with **128-bit block cipher**, with **AES-256** [15] as the encryption algorithm. The RKC idea can easily be extended to other block cipher which will require further definitions.

RKC uses Deterministic Random Bit Generator (DRBG) function for generation pseudorandom bits. DRBG generates same sequence of bits if the same seed is used again. Our DRBG function uses **Hash_DRBG** as specified in Section 10.1.1 of NIST SP800-90[16] with **SHA-256**[17] as the underlying hash-function. That has:

- Seed length for DRBG = 440 bits

  (Seed S must have sufficient entropy for generating pseudorandom number).
- Requested number of bits = 256 bits.
- Maximum number of pseudorandom bits that can be generated without reseeding $\leq 2^{48}$.

Initially, let's say pseudorandom number generated from DRBG is R_1. Then encryption key $K_1$for the 1st block is obtained as:

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

3.1 Inputs

To send a message sender must provide the following information:

- An Encryption key $\mathbf{K_0}$, 32 bytes (256-bits) in length.
- A Seed **S**, 55 bytes (440-bits) in length for instantiating DRBG
- Message M, that has 128(n-1) + y bits. Where n is the total number of blocks each of size 128-bits and y are the number of bits in last block.

For better performance, the total number of plaintext blocks to be encrypted (**n**), must be provided as an input so that appropriate number of key streams can be generated beforehand.

3.2 Encryption

The Encryption process is shown in Figure 1.

Let's say P_i denotes the plaintext block, whereas $1 \leq i \leq n$, and n is total no. of blocks of plaintext. To encrypt a message M of 128(n-1) + y bits, we first determine encryption-key stream by:

$$K_i = K_{i-1} \oplus R_i \quad \text{Fori} = 1,\ldots.., n \qquad (1)$$

The step performed so far is a part of preprocessing. Each key generated(K_i), is used for the encryption of plaintext blocks(P_i), for i = 1 to n respectively and this can be executed in parallel.

$$C_i = E(K_i, P_i) \qquad (2)$$

The cipher text generated (C_i) is XOR-ed with P_i to produce X_i. This operation is also parallelizable. X_i as computed below is used to generate authentication tag.

$$X_i = C_i \oplus P_i \qquad (3)$$

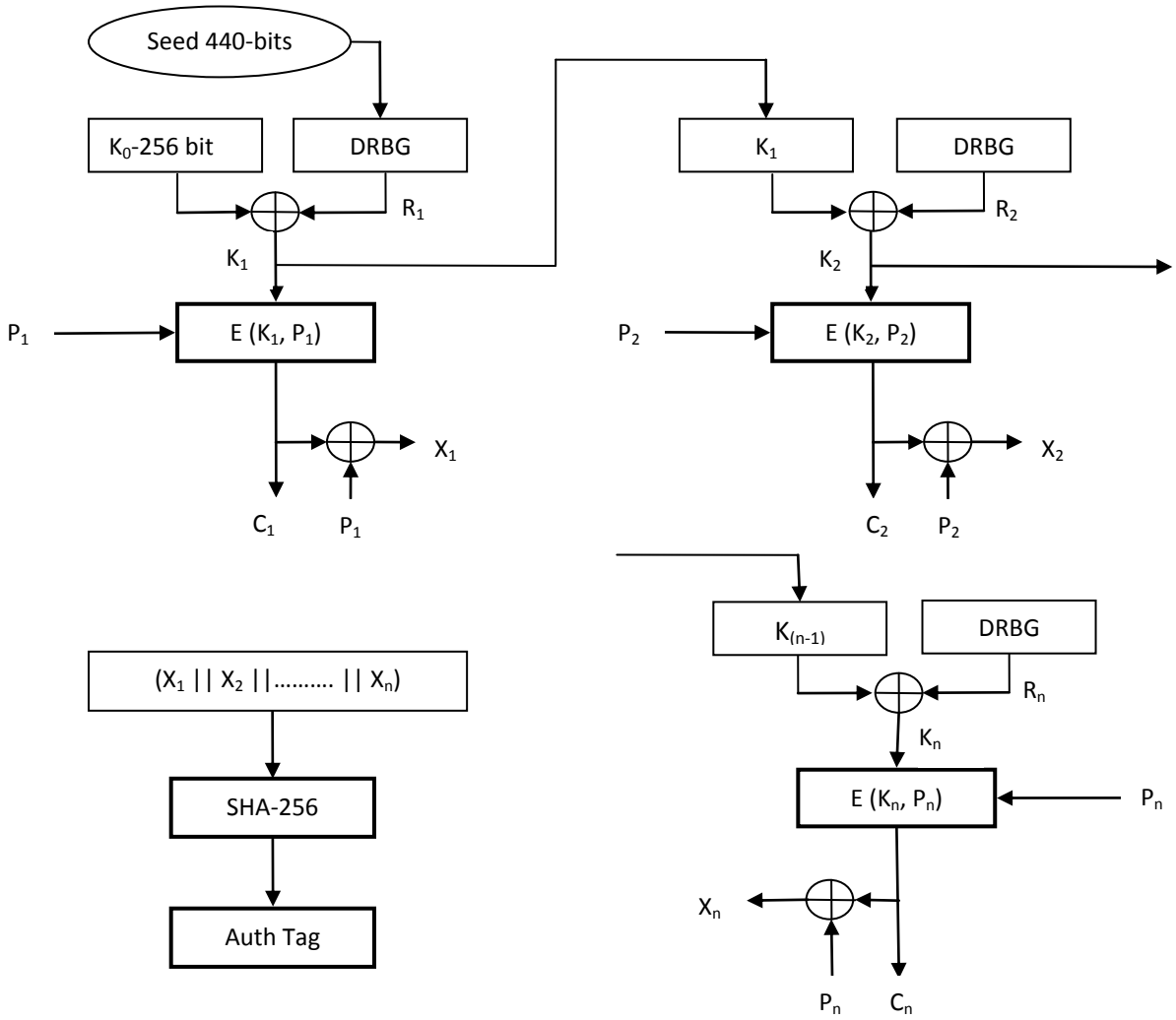

Fig 1: The authenticated encryption operation. For simplicity, first two and then last block of plaintexts are shown.

3.3 Decryption

Decryption process is shown in figure 2.

For decryption, receiver needs Secret Key (K_0), Seed S for DRBG instantiation and cipher text. For decryption, key stream (K_i) is generated, same as in equation 1. Then decryption can be executed in parallel for recovering P_i. Whereas,

$$P_i = D(K_i, C_i) \qquad (4)$$

At the receiving end, $X_i$is also calculated, same as in equation 3.

3.4 Authentication

Authentication part includes:

- **Origin:** Whether sender is legitimate or not
- **Integrity:** There is no alteration in cipher text received.

In RKC block cipher mode the authentication is provided on plaintext and the cipher text. For authentication, 'Authentication Tag' T is used. The plaintext (P_i) and cipher text (C_i) are XOR-ed to produce 128-bit string X_i, which are concatenated sequentially and processed by hash algorithm **SHA-256**[17] to give authentication tag T.

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

Where,

$$X_i = P_i \oplus C_i$$

The process of generating authentication data X_i can also be executed in parallel. However, for producing T, complete padding (X_1 to X_n) is required. So, authentication up to much extent is parallelizable.

At receiving end, equation 3 is repeated and authentication tag T' is computed. If both T and T' matches, then cipher text is authenticated.

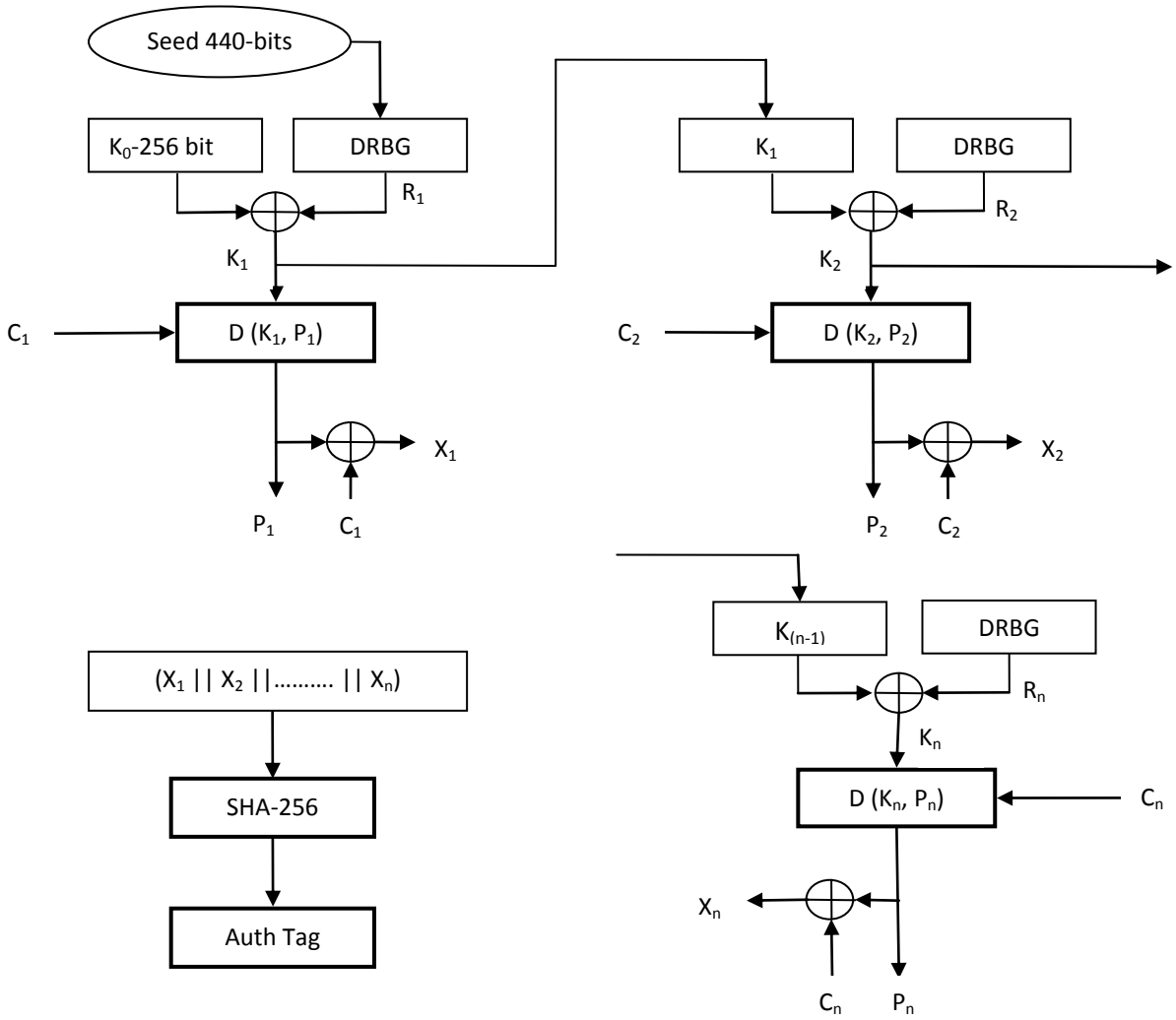

Fig 2: The authenticated decryption operation.First two and then last block of cipher texts are shown.

3.5 Output

The final result consists of:

- Cipher text**C** and
- The Authentication tag **T**

Cipher text ($C = C_1 \parallel C_2 \parallel \ldots \parallel C_n$) is padded with Authentication Tag T and send to receiver.

4. ALGORITHM

Sender and receiver share two keys: K_0(256-bits) and S (440-bits) using a secure channel. $K_0$is the secret key required by the underlying encryption algorithm (AES-256) and S is the seed required by HASH_DRBG.

4.1 Encryption Process

1. Pre-compute encryption key stream using shared Secret Key and Seed for HASH_DRBG for n (i.e. N/128) blocks (where N is the message length)
   - For i = 1 to n do

     $K_i = K_{i-1} \oplus R_i$
2. Divide plaintext of N-bits into equal length blocks ($P_1 \ldots P_n$). Where $|P_i| = 128$.
3. Encrypt each 128 bit P_i using AES-256 with K_i as the encryption key. This will be executed in parallel.
4. Perform XOR operation as shown below to generate authentication data X_i. This will also be executed in parallel
   - $X_i = P_i \oplus C_i$
5. Pad X_1 to X_n and input into SHA-256 to generate Authentication Tag T
   - $T = HASH(X_1 \parallel X_2 \parallel \ldots\ldots\ldots X_n)$

Send C and T to the receiving end, where $C = C_1 \parallel C_2 \parallel \ldots\ldots \parallel C_n$

4.2 Decryption Process

1. Precompute decryption key stream using shared Secret Key and Seed for HASH_DRBG for n (i.e. N'/128) blocks (where N' is the cipher text length)

- For i = 1 to n do

$$K_i = K_{i-1} \oplus R_i$$

2. Divide cipher text of N'-bits into equal length blocks ($C_1 \ldots C_n$). Where $|C_i| = 128$.
3. Decrypt each 128 bit C_i using AES-256 with K_i as the decryption key. This will be executed in parallel.
4. Perform XOR operation as shown below to generate authentication data X_i. This will also be executed in parallel
   - $X_i = P_i \oplus C_i$
5. Pad X_1 to X_n and input into SHA-256 to generate Authentication Tag T'
   - T' = HASH($X_1 \| X_2 \| \ldots\ldots\ldots X_n$)
6. Compare T' computed, with Tag T received.

   If T' = T, then authentication is complete.

5. SECURITY

Theorem 1: For encryption purposes RKC mode for AES is secure against the attacks of $2^{256+440}$ steps of operations provided AES and underlying HASH_DRBG is secure.

Theorem 2: For authentication purposes RKC mode is secure provided underlying SHA-256 is secure.

6. SUMMARY

The various parameters of RKC mode of operation is compared with other authenticated encryption modes like CCM, OMAC, EAX, GCM, OCB and CWC and can be found in Annexure I.

7. CONCLUSION

By making use of Deterministic Random Bit Generator in RKC mode, every block of plaintext is being encrypted with a different key. So we are closing to **One-time Pad (OTP)** [18] with practical limit equivalent to period of DRBG non-repetition (i.e., 2^{48} pseudorandom numbers without repetition) and thus there is significant improvement in overall security of the block cipher mode.

One-time pad is also called as the perfect cipher. The idea of OTP was dropped in real world application because it requires a very long key. Key in OTP cannot be used again and so, number of keys required was also very large, and was directly proportional to number of messages. In RKC mode this problem is resolved with the use of two keys, one for encryption and other for seeding. Also, reseeding is not required in this block cipher mode because when pseudorandom number more than 2^{48} is generated then shared secret key is expected to change and that would generate a different encryption key.

The slight variation of RKC mode can be used as a confidentiality mode that can be used in application like hard-disk compression. For that, we need to skip the process of generating Xi, which is done by XOR-ing Pi and Ci, and this will also reduce the overall computational cost of the mode. Also, there is no restriction in the plaintext message length when authentication is not required.

8. REFERENCES

[1] Lipmaa, H., Rogaway, P., Wagner, D., CTR-Mode Encryption, *Comments to NIST concerning AES Modes of Operations*, pp. 1-3. Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ctr/ctr-spec.pdf.

[2] McGrew, D. A., Nov 15, 2002, *Counter Mode Security: Analysis and Recommendations*, pp. 2.

[3] McGrew, D. A., Viega, J., June 2005, *Galois/CTR Mode of Operation*. Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf.

[4] Kohno, T., Viega, J., Whiting, D., May 27, 2003, *the CWC Authenticated Encryption (Associated Data) Mode*. Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/cwc/cwc-spec.pdf.

[5] 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.

[6] Furguson, N., May 20, 2005, *Authentication Weakness in GCM*. Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/CWC-CM/Ferguson2.pdf.

[7] Furguson, N.,May 20, 2005, *Authentication Weakness in GCM*, pp. 7-8. Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/CWC-CM/Ferguson2.pdf.

[8] Black, J., Rogaway, P., Comments to NIST concerning AES Mode of Operation: *A suggestion for handling Arbitrary-Length Messages with CBC-MAC*, pp. 1-2, Section 2.

[9] Iwata, T., Kurosawa, K.,Dec 20, 2002, *One Key CBC-MAC*, Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/omac/omac-spec.pdf.

[10] Bellare, M., Rogaway, P., Wagner, D., April 13, 2003, *A Conventional Authenticated-Encryption Mode*, Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/eax/eax-spec.pdf.

[11] Housley, R., Whiting, D., Furguson, N., June 3, 2002, Counter with CBC-MAC, *Submission to NIST Concerning AES mode of Operation*. Available online athttp://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ccm/ccm.pdf.

[12] http://www.zork.org/cwc/draft-irtf-cfrg-cwc-01.txt, Section 2.5.

[13] Krovetz T., Rogaway P., March 21, 2011, *the Software Performance of Authenticated Encryption Modes.*

[14] Rogaway, P., Bellare, M., Black, J., Krovetz, T.,Aug 3, 2001, *OCB: A Block Cipher Mode of Operation for Efficient Authenticated Encryption*. Available online at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ocb/ocb-spec.pdf.

[15] Announcing the Advanced Encryption Standard (AES), *Federal Information Processing Standards Publication, FIPS 197*, Nov 26, 2001.

[16] 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. Available online at: http://csrc.nist.gov/publications/drafts/800-90/Draft_SP800-90A-Rev1_May-2011.pdf.

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

[18] William Stallings, Cryptography and Network Security Principles and Practices, Pearson, Fourth Edition, pp. 48-49.

ANNEXURE I

Table 1: Comparison of RKC with other authenticated modes of operation

	CCM[11]	**CWC[4]**	**EAX[10]**	**GCM[3]**	**OCB[14]**	**OMAC[9]**	**RKC**
SECURITY STRENGTH	2^{128} for key length larger or equal to 256 bit.	Provably secure and its provable security depends only on the pseudo randomness of the underlying block cipher.	Provably secured. Reduction from block cipher pseudorandom permutation security.	Relies on random permutation of the underlying block cipher	Provably secured as long as the underlying block cipher is secured	OMAC is variable input length pseudorandom function. So, depends upon the input.	$2^{256+440}$ steps of operations.
ERROR PROPAGATION	None	None	None	None	Infinite	None	None
PARALLELIZABILITY	Encryption can be parallelized	Encryption parallelizable. Amount of parallelism for the hashing portion can be determined by the implementer.	None	Encryption: block level. Authentication: bit-level	Encryption and decryption are parallelizable	Sequential	Encryption and decryption can be executed in parallel. And authentication can be parallelized up to greater extent.
PRE-PROCESSING	Encryption key stream can be precomputed	CTR mode key stream can be precomputed	Key stream only	Key stream can be precomputed. Fixed parts of IV can be processed in advance	Limited	Limited	Key stream generation for encryption and decryption can be pre-computed.
MESSAGE LENGTH	Octet aligned message of arbitrary length, up to 2^{8L} octets, and octet aligned arbitrary additional authenticated data, up to 2^{64} octets.	Up to $128.2^{(32-1)}$ bits.	Arbitrary	Arbitrary message up to $2^{39} - 256$ bits. Arbitrary additional authenticated data up to 2^{64} bits.	Any bit string	Arbitrary length	Message of arbitrary length up to 2^{64} (because of message input limit of underlying SHA-2)when authentication is required. Otherwise, no restriction.
CIPHERTEXT EXPANSION	4, 6, 8, 10, 12, 14, or 16 octets depending on size of MAC selected.	Minimum possible.	Arbitrary	Cipher text length is identical to plaintext length 0 to 128 bits required for the authentication tag.	Minimal possible	Not applicable	None. Would require further definitions.

KEY MATERIAL / NONCE / IV	1 key + Nonce/IV	1 key + Nonce/IV	1 key + Nonce/IV	1 key + Nonce/IV	1 key + Nonce/IV	1 key + Nonce/IV	1 key+ IV(Seed)
MEMORY REQUIREMENT	Requires memory for encrypt operation of the underlying block cipher,plaintext, cipher text (expanded for CBC-MAC), and a per-packet counter.	Depends upon underlying block cipher.	Small constants	Small constants	Modest	Modest	Require memory for plaintext, cipher text and key stream generated for n blocks, and also for the authentication data (X_i).