Korea Communications Standard

방송통신표준

KCS.KO-12.0131 제정일: 2013년 12월 31일

기밀성과 메시지 인증을 제공하는 128 비트 블록 암호 운영 모드

Modes of Operation of 128-bit Block Cipher for Confidentiality and Message Authentication

미래창조과학부

국립전파연구원

방송통신표준

KCS.KO-12.0131 제정일: 2013년 12월 31일

기밀성과 메시지 인증을 제공하는 128 비트 블록 암호 운영 모드

Modes of Operation of 128-bit Block Cipher for Confidentiality and Message Authentication

미래창조과학부

국립전파연구원

서 문

1. 표준의 목적

본 표준은 정보처리 시스템 및 정보통신망 환경에 사용되는, 입출력 데이터 블록의 길이가 128 비트인 블록 암호 알고리즘의 기밀성과 메시지 인증 기능을 동시에 제공하는 운영 모드 CCM(Counter with CBC-MAC), GCM(Galois/Counter Mode)에 대해 규정한다.

2. 주요 내용 요약

본 표준은 128 비트 블록 암호 알고리즘에 대한 CCM(Counter with CBC-MAC), GCM(Galois/Counter Mode)운영 모드의 암복화·복호화 과정 및 각각의 운영 모드에서 사용되는 내부 연산 함수들을 규정하고 있으며, 구현 적합성을 확인하기 위한 참조 구현값이 부록에 명시되어 있다.

3. 표준 적용 산업 분야 및 산업에 미치는 영향

본 표준은 기밀성 기능을 제공하는 정보 보호 시스템 및 암호 제품에 다양하게 활용되어 국내 정보통신망의 안전성· 신뢰성을 제고할 수 있다.

4. 참조 표준(권고)

4.1 국외 표준(권고)

- NIST SP 800-38C, ‘Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality’, 2004.
- NIST SP 800-38D, ‘Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC’, 2007.

4.2 국내 표준

해당 사항 없음.

5. 참조 표준(권고)과의 비교

5.1 참조 표준(권고)과의 관련성

본 표준은 'NIST SP 800-38C'과 'NIST SP 800-38D' 표준에서 CCM과 GCM 운영 모드에 대한 기본적인 개념 및 설계 기준만을 참조하고 있다.

5.2 참조한 표준(권고)과 본 표준의 비교표

KCS.KO-12.0131	참조 표준		비고
	NIST SP 800-38C	NIST SP 800-38D	
1. 개요	-	-	추가
2. 표준의 구성 및 범위	-	-	추가
3. 용어 정의 및 약어	-	-	추가
4. CCM 운영 모드	5. PRELIMINARIES 6. CCM SPECIFICATION	-	수정
5. GCM 운영 모드	-	5. ELEMEMTS OF GCM 6. MATHEMATICAL COMPONENTS OF GCM 7. GCM SPECIFICATION	수정
부록 I. CCM 운영 모드 참조 구현값(SEED)	-	-	추가
부록 II. GCM 운영 모드 참조 구현값(SEED)	-	-	추가
부록 III. 관련 문헌	-	-	추가

6. 지식 재산권 관련 사항

본 표준의 ‘지적 재산권 확약서’ 제출 현황은 국립전파연구원 웹사이트에서 확인할 수 있다.

※본 표준을 이용하는 자는 이용함에 있어 지식 재산권이 포함되어 있을 수 있으므로, 확인 후 이용한다.

※본 표준과 관련하여 접수된 확약서 이외에도 지식 재산권이 존재할 수 있다.

7. 시험 인증 관련 사항

7.1. 시험 인증 대상 여부

해당 사항 없음.

7.2. 시험 표준 제정 현황

해당 사항 없음.

8. 표준의 이력 정보

8.1. 표준의 이력

판 수	제정·개정일	제정·개정 내역
제 1 판	2013.12.31.	제정 KCS.KO-12.0131

8.2. 주요 개정 사항

해당 사항 없음.

Preface

1. Purpose of Standard

This standard specifies CCM(Counter with CBC-MAC) and GCM(Galois/Counter Mode) mode of operation for 128-bit block cipher(block size of Input/output data is 128bit) in the information systems and communication environments.

2. Summary of Contents

This standard explains the structure and functions of CCM and GCM mode of operation for 128-bit block cipher.

3. Applicable Fields of Industry and its Effect

This standard facilitates to implement CCM and GCM mode of operation for 128-bit block cipher that has been applied to many cryptographic applications to provide secure communication environment.

4. Reference Standards(Recommendations)

4.1. International Standards(Recommendations)

- NIST SP 800-38C, “Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality”, 2004.
- NIST SP 800-38D, “Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC”, 2007.

4.2. Domestic Standards

None

5. Comparison between Reference Standards(Recommendations) and this Standard

5.1. Relevance of this Standard with Reference Standards(Recommendations)

This standard refers to the basic concept and design principle of CCM and GCM mode of operation in NIST SP 800-38C and SP 800-38D.

5.2. A Comparative Table of Reference Standard(Recommendation) and this Standard

KCS.KO-12.0131	Reference Standards		Remarks
	NIST SP 800-38C	NIST SP 800-38D	
1. Introduction	-	-	Added
2. Constitution and Scope	-	-	Added
3. Terms Definition and Abbreviations	-	-	Added
4. Mode of Operation CCM (Counter with CBC-MAC)	5. PRELIMINARIES 6. CCM SPECIFICATION	-	Modified
5. Mode of Operation GCM(Galois/Counter Mode)	-	5. ELEMEMTS OF GCM 6. MATHEMATICAL COMPONENTS OF GCM 7. GCM SPECIFICATION	Modified
Appendix Ⅰ. CCM Test Vectors(SEED)	-	-	Added
Appendix Ⅱ. GCM Test Vectors(SEED)	-	-	Added
Appendix Ⅲ. Related Documents	-	-	Added

6. Statement of Intellectual Property Rights

"Written Confirmation of Intellectual Property Rights" for this standard can be referenced to the website of the National Radio Research Agency.

Those using this standard must confirm that whether intellectual property rights are included in this standard.

Other intellectual property rights may exist in relation to written confirmation received for this standard.

7. Statement of Testing and Certification

7.1. Object of Testing and Certification

None

7.2. Standards of Testing and Certification

None

8. History of Standard

8.1. Change History

Edition	Issued date	History
The 1st edition	2013.12.31.	Established KCS.KO-12.0131

8.2. Revisions Related Details

None

목 차

Contents

기밀성과 메시지 인증을 제공하는 128 비트 블록 암호 운영 모드 (Modes of Operation of 128-bit Block Cipher for Confidentiality and Message Authentication)

1. 개요

본 표준은 128 비트 블록 암호 알고리즘(입출력 데이터 블록의 길이가 128 비트인 블록 암호 알고리즘)을 이용하여 128 비트 블록보다 큰 길이의 데이터를 암호화·복호화할 때, 해당 데이터를 블록 단위로 분할하여 암호화·복호화하고 기밀성과 메시지 인증 기능을 동시에 제공하는 CCM(Counter with CBC-MAC), GCM(Galois/Counter Mode) 운영 모드 방법에 대해 규정한다.

2. 표준의 구성 및 범위

본 표준에서는 기밀성과 메시지 인증 기능 제공을 위한 128 비트 블록 암호 알고리즘의 운영 모드로 CCM(Counter with CBC-MAC), GCM(Galois/Counter Mode)모드를 규정하고, 각 운영 모드의 암호화·복호화 과정의 상세 명세와 운영 모드를 사용하기 위한 매개변수 값을 규정한다.

부가적으로, 표준 적합성 테스트를 위해 128 비트 블록 암호 알고리즘 SEED의 참조 구현값을 부록에서 기술한다. 그 외 다른 128 비트 블록 암호 알고리즘 ARIA에 대한 참조 구현값은 부록 III의 [2]에서, AES(Advanced Encryption Standard)의 참조 구현값은 부록 III의 [3]에서 참조할 수 있다.

3. 용어 정의 및 약어

3.1. 용어 정의

평문 **또는 평문 데이터**	암호화 대상 또는 암호문을 복호화한 원래의 데이터
암호문 **또는 암호문 데이터**	평문을 암호 알고리즘을 이용하여 변환한 데이터
암호화	평문을 암호문으로 변환하는 과정
복호화	암호문을 평문으로 변환하는 과정
비밀 키	평문 또는 암호문의 암호화·복호화에 사용되는 비밀 정보

연관 데이터 (Associated Data)	인증만을 수행하는 CCM 암호화 단계의 입력값
추가 인증 데이터 (Additional Authenticated Data)	인증만을 수행하는 GCM 암호화 단계의 입력값
최하위 비트 (Least Significant Bit)	비트 스트링의 가장 오른쪽 비트
최상위 비트 (Most Significant Bit)	비트 스트링의 가장 왼쪽 비트
난스(Nonce)	구체적으로 명시된 맥락(context)에서 한번만 사용되는 값
128 비트 블록 암호 알고리즘	입/출력 데이터 블록의 길이가 128 비트인 블록 암호 알고리즘

3.2. 약어

CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
GCM	Galois/Counter Mode
MAC	Message Authentication Code

3.3. 기호

A	연관 데이터, 추가 인증 데이터
a	*A*의 바이트 길이
Alen	*A*의 비트 길이
C	암호문
Clen	*C*의 비트 길이
$CIPH_K(X)$	비밀 키 *K*를 이용하여 데이터 *X*를 암호화하는 128 비트 블록 암호 알고리즘
H	해시 서브 키
ICB	초기 카운터 블록
inc(*X*)	비트열 X의 정수 표현
IV	초기 벡터
K	비밀 키
Klen	*K*의 비트 길이
Len(*X*)	비트열 *X*의 비트 길이

$LSB_Y(X)$	비트열 X의 최하위 비트에서부터 Y 비트 길이만큼의 비트열
$MSB_Y(X)$	비트열 X의 최상위 비트에서부터 Y 비트 길이만큼의 비트들
N	난스
n	N의 바이트 길이
$Nlen$	N의 비트 길이
P	평문
$Plen$	P의 비트 길이
Q	P의 바이트 길이에 대한 비트 표현
q	Q의 바이트 길이
T	인증 태그
t	T의 바이트 길이
$Tlen$	T의 비트 길이

4. CCM 운영 모드

CCM 운영 모드는 카운터 모드(Counter mode)와 CBC-MAC(Cipher Block Chaining-Message Authentication Code)의 기능을 결합한 운영 모드로서 메시지의 기밀성과 인증 기능을 동시에 제공한다.

4.1. 입출력 데이터의 요구 사항 및 유효성

4.1.1. 입출력 데이터의 요구 사항

CCM 운영 모드의 입력 데이터 N, P, A는 완전한 데이터 블록 B_0, B_1,…, B_r로 변환되어야만 한다. 양의 정수인 r은 입력 데이터의 포맷 함수와 입력 데이터들에 의해 결정된다. 주어진 키에 대해, 다음 3 가지 요구 사항들은 입력 데이터의 포맷 함수에 대하여 반드시 만족되어야 한다.

요구 사항 1) 첫 번째 블록 B_0는 N을 유일하게 결정해야 한다.

요구 사항 2) 포맷화된 데이터는 P와 A를 유일하게 결정해야 한다.

만약 (N, P, A)와 (N', P', A')가 각각 B_0, B_1, …, B_r과 B'_0, B'_1,…, B'_r로 입력 데이터의 포맷 함수에 대한 서로 다른 입력 데이터라면, i에 대하여 B_i와 B'_i는 서로 달라야만 한다.

요구 사항 3) 첫 번째 블록 B_0는 CCM 운영 모드에서 이용되는 임의의 카운터 블록과 달라야만 한다.

4.1.2. 유효한 (입출력) 데이터

입력 데이터 N, A, P 는 8 비트의 배수이어야 한다. 따라서 입력 비트열은 바이트열로 표현될 수 있으며, 입력 비트열 N, A, P의 바이트 길이는 각각 n, a, p로 표기한다. 이와 마찬가지로, t는 T의 바이트 길이를 표기한다. P의 바이트 길이는 포맷화된 데이터의 첫 번째 블록에서 Q에 명시하고, Q의 바이트 길이는 q로 표기한다. 따라서, Q는 q 바이트에 p를 이진법으로 표현한 $[p]_{8q}$와 같다. 예를 들어, 만약 q = 3이고 P가 4096 비트로 구성된 비트열이라면, p = 512가 되어 Q는 다음과 같다.

$$Q = 00000000\ 00000010\ 00000000$$

입력 데이터에 대한 포맷 함수는 다음과 같이 N, P, A의 값과 길이를 제한한다. 다음의 제한 조건을 만족하는 값은 유효하다고 규정한다.

제한 조건 1) $t \in \{4, 6, 8, 10, 12, 14, 16\}$
제한 조건 2) $q \in \{2, 3, 4, 5, 6, 7, 8\}$
제한 조건 3) $n \in \{7, 8, 9, 10, 11, 12, 13\}$
제한 조건 4) $n + q = 15$
제한 조건 5) $a < 2^{64}$

4.2. 입력 데이터에 대한 포맷 함수

입력 데이터 (N, A, P)를 블록열 B_0, B_1,…, B_r로 인코딩하는 포맷 함수는 3 가지 단계로 구성된다. 4.2.1 절에서는 N과 제어 정보의 포맷화를 명시하고, 5.2.2 절에서는 A 의 포맷화를 명시한다. 4.2.3 절에서는 P 의 포맷화를 명시한다.

4.2.1. 난스와 제어 정보의 포맷화

포맷화된 블록열의 첫 번째 블록 B_0의 첫 번째 바이트는 제어 정보에 대한 4 개의 플래그를 담고 있다. 플래그는 예약된 한 비트, $Adata$ 한 비트, t의 값에 의하여 인코딩된 3 비트, q의 값에 의하여 인코딩된 3 비트로 구성된다. t의 인코딩은 $[(t - 2)/2]_3$으로 규정하며 q의 인코딩은 $[q - 1]_3$으로 규정한다. 예를 들어, 만약 MAC의 길이가 8 바이트라면 t는 011로 인코딩된다. 정당한 t나 q 의 값은 000로 대응되지 않는다. 만약 a = 0이면 $Adata$ 비트는 0이고, 그렇지 않으면 1 이다. B_0의 첫 번째 바이트에서 플래그들의 위치는 표 4.1에 주어진다.

표 4.1 B_0의 첫 번째 바이트에서 플래그들의 위치

비트 번호	7	6	5	4	3	2	1	0
내용	예약됨	*Adata*	$[(t-2)/2]_3$			$[q-1]_3$		

포맷화된 첫 번째 블록의 나머지 15 바이트는 표 4.2 와 같이 n 바이트의 N 과 q 바이트 P 의 길이 정보로 구조화된다.

표 4.2 B_0의 포맷화

바이트 번호	0	1…15-q	16-q…15
내용	플래그	N	Q

예를들어, B_0가 다음과 같다고 하자.

01101110 00010011 11010100 10100011 01011101 01110001 10100101 00000000
00000000 00000000 00000000 00000000 00000000 00000000 01000100 00000001

- o *Adata* = 1이기 때문에 A 의 길이는 0이 아니다.
- o $[(t-2)/2]_3$ = 101이기 때문에 인증 코드는 12 바이트로 구성된다.
- o Q의 바이트 길이가 7이기 때문에 Q는 00000000 00000000 00000000 00000000 00000000 01000100 00000001 이다.
- o $Q = [17409]_{56}$ 이기 때문에 P 는 17409 바이트로 구성된다.
- o N의 바이트 길이가 8이기 때문에 N 은 00010011 11010100 10100011 01011101 01110001 10100101 00000000 00000000 이다.

4.2.1. A의 포맷화

만약 $a = 0$ 이면 B_0의 첫 번째 바이트의 *Adata* 필드를 통하여 확인할 수 있으며, 포맷화된 데이터에 A 가 할당되지 않았음을 확인할 수 있다. 만약 $a > 0$ 이면, a 는 a 의 값에 따라 인코딩되고, a 의 인코딩된 결과값에 A 가 덧붙여진다. 그리고 포맷화된 A 가 16 바이트의 배수가 될 수 있도록 최소 개수의 0 을 덧붙인다. A 에 의하여 인코딩된 데이터 블록을 B_0, B_1,…, B_u으로 표기하며, u 는 a 에 의하여 결정된다.

a 에 따라 인코딩되는 방법은 다음과 같다.

- o 만약 $0 < a < 2^{16}-2^8$이면, a 는 $[a]_{16}$으로 인코딩된다.
- o 만약 $2^{16}-2^8 \le a < 2^{32}$이면, a 는 0xff ‖ 0xfe ‖ $[a]_{32}$로 인코딩된다.

o 만약 $2^{32} \leq a < 2^{64}$이면, a는 0xff ‖ 0xff ‖ $[a]_{64}$로 인코딩된다.

예를 들어, 만약 $a = 2^{16}$이면 a의 인코딩 결과는 다음과 같다.

11111111 11111110 00000000 00000001 00000000 00000000

4.2.3. *P*의 포맷화

포맷화된 A 블록열에 P의 블록열이 덧붙는다. P의 블록열이 16 바이트의 배수가 될 수 있도록 최소 개수의 0을 덧붙인다. P에 의하여 인코딩된 데이터 블록을 B_{u+1}, B_{u+2},…, B_r로 표기하며, $r = u + \lceil p/16 \rceil$ 이다.

4.3. 카운터 블록에 대한 포맷 함수

카운터 블록에 대한 포맷 함수는 완전한 데이터 블록에 카운터 색인 i를 포맷화하는 것과 동일하다. 카운터 블록 Ctr_i는 표 4.3과 같이 포맷화된다.

표 4.3 카운터 블록 Ctr_i의 포맷화

바이트 번호	0	1…15−q	16−q…15
내용	플래그	N	$[i]_{8q}$

각각의 카운터 블록 Ctr_i에서 이용되는 플래그의 표 4.4와 같이 포맷화된다.

표 4.4 카운터 블록 Ctr_i의 첫 번째 바이트에서 플래그들의 위치

비트 번호	7	6	5	4	3	2	1	0
내용	예약됨	예약됨	0	0	0	$[q-1]_3$		

예약된 비트는 향후 확장된 필드로 이용되기 위하여 예약되었으며, 0으로 설정된다. 카운터 블록 Ctr_i가 모든 B_0블록과 구별되기 위하여 3, 4, 5 번째 비트는 0으로 설정되며, 0, 1, 2 번째 비트는 B_0와 동일하게 인코딩된다.

4.4. CCM 모드 암호화

o 전제 조건
- 128 비트 블록 암호 알고리즘 CIPH
- 비밀 키 K
- 입력 데이터에 대한 포맷 함수
- 카운터 블록에 대한 포맷 함수
- 인증 태그의 길이 $Tlen$

o 입력
- 유효한 난스 N
- 비트 길이 $Plen$을 갖는 유효한 평문 P
- 유효한 연관 데이터 A

o 출력
- 암호문 C

o 처리 단계 :

① 블록 B_0, B_1,…, B_r을 생성하기 위하여 (N, P, A)를 입력 데이터에 대한 포맷 함수에 적용한다.

② $Y_0 = \mathrm{CIPH}_K(B_0)$를 계산한다.

③ i= 1에서 r까지 다음을 수행한다.

$$Y_i = \mathrm{CIPH}_K(B_i \oplus Y_{i-1})$$

④ $T = \mathrm{MSB}_{Tlen}(Y_r)$을 계산한다.

⑤ 카운터 블록 Ctr_0, Ctr_1,…, Ctr_m($m = \lceil Plen/128 \rceil$)을 생성하기 위하여 카운터 블록에 대한 포맷 함수를 적용한다.

⑥ j = 0 에서 m 까지 다음을 수행한다.

$$S_j = \mathrm{CIPH}_K(Ctr_j)$$

⑦ $S = S_1 \| S_2 \| \cdots \| S_m$을 계산한다.

⑧ $C = (P \oplus \mathrm{MSB}_{Plen}(S)) \| (T \oplus \mathrm{MSB}_{Tlen}(S_0))$

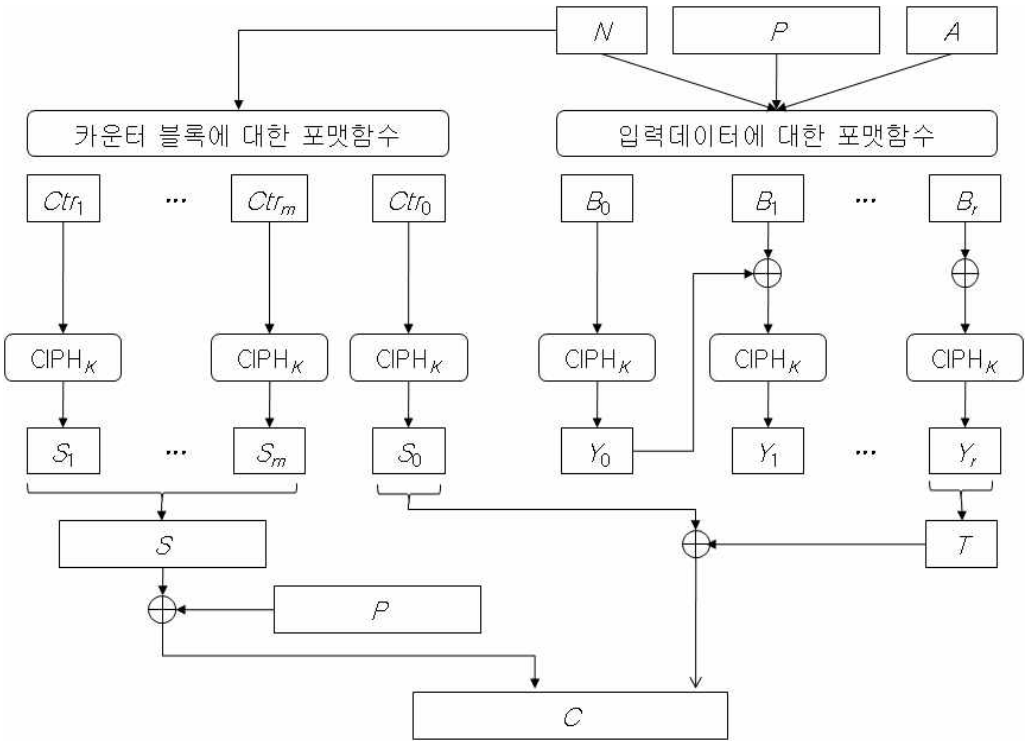

그림 4.1 CCM 모드 암호화

4.5. CCM 모드 복호화

o 전제 조건
- 128 비트 블록 암호 알고리즘 CIPH
- 비밀 키 *K*
- 입력 데이터에 대한 포맷 함수
- 카운터 블록에 대한 포맷 함수
- 인증 태그의 길이 *Tlen*

o 입력
- 난스 *N*
- 연관 데이터 *A*
- 비트 길이 *Clen*을 갖는 암호문 *C*

o 출력
- 평문 *P* 또는 *FAIL*

o 처리 단계 :
① 만약 *Clen* ≤ *Tlen*이면 *FAIL*을 출력한다.

② 카운터 블록 Ctr_0, Ctr_1,…, Ctr_m(m= ⌈ $Clen-Tlen$/128 ⌉)을 생성하기 위하여 카운터 블록에 대한 포맷 함수를 적용한다.

③ j = 0 에서 m 까지 다음을 수행한다.

$$S_j = \mathrm{CIPH}_K(Ctr_j)$$

④ $S = S_1 \| S_2 \| \cdots \| S_m$을 계산한다.

⑤ $C = \mathrm{MSB}_{Clen-Tlen}(C) \oplus \mathrm{MSB}_{Clen-Tlen}(S)$을 계산한다.

⑥ $T = \mathrm{LSB}_{Tlen}(C) \oplus \mathrm{MSB}_{Tlen}(S_0)$을 계산한다.

⑦ N, P, A가 입력 데이터의 요구 사항을 만족한다면, 블록 B_0, B_1,…, B_r을 생성하기 위하여 (N, P, A)를 입력 데이터에 대한 포맷 함수에 적용한다.

⑧ $Y_0 = \mathrm{CIPH}_K(B_0)$를 계산한다.

⑨ i = 1 에서 r까지 다음을 수행한다.

$$Y_i = \mathrm{CIPH}_K(B_i \oplus Y_{i-1})$$

⑩ 만약 $T \neq \mathrm{MSB}_{Tlen}(Y_r)$이면 $FAIL$을 출력하고 그렇지 않으면 P를 출력한다.

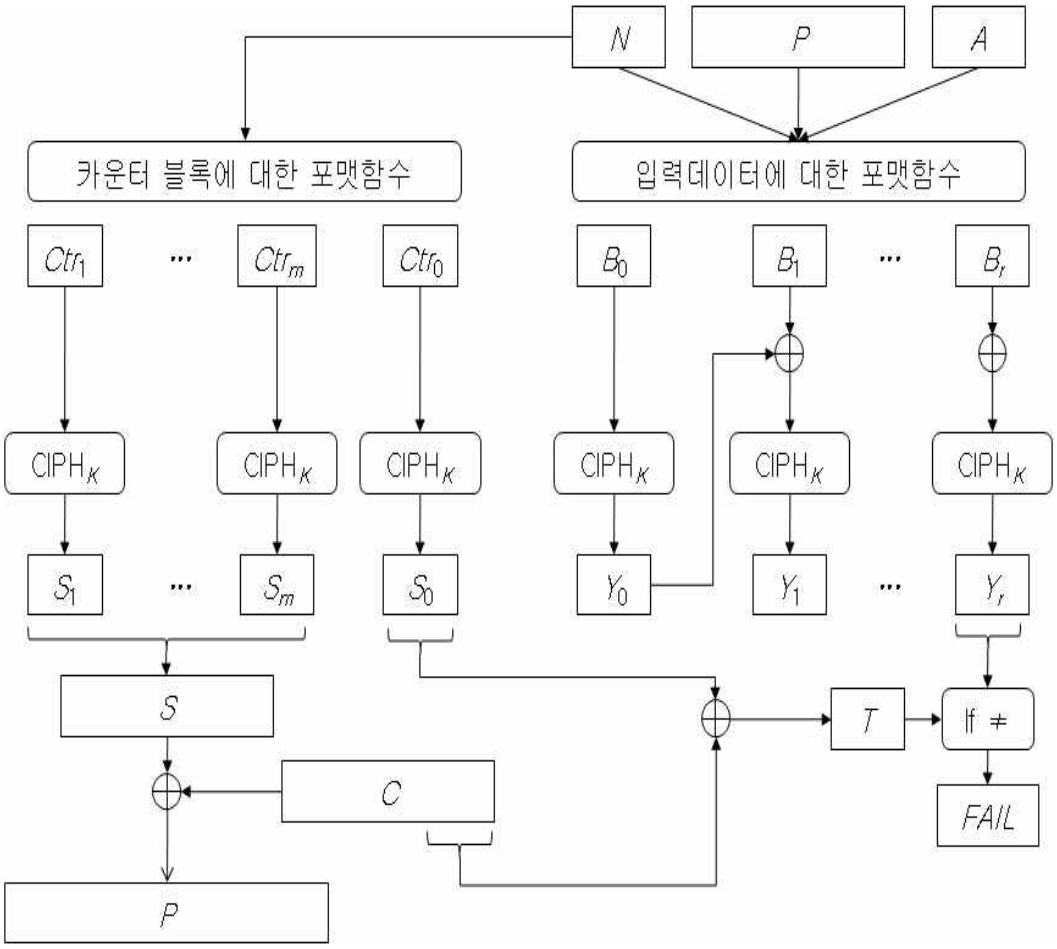

그림 4.2 CCM 모드 복호화

5. GCM 운영 모드

GCM 운영 모드는 이진 Galois체상에서 규정된 GHASH 함수를 이용하여 기밀 데이터의 인증을 보장하는 운영 모드로서 메시지의 기밀성과 인증 기능을 동시에 제공한다.

5.1. 입출력 데이터의 요구 사항

GCM 암호화 함수의 입력 데이터에 대한 비트 길이는 다음을 만족하여야만 한다.

- o $\text{len}(P) \leq 2^{39}-256$
- o $\text{len}(A) \leq 2^{64}-1$
- o $1 \leq \text{len}(IV) \leq 2^{64}-1$

평문, 추가적인 데이터, 초기 벡터의 비트 길이는 8의 배수여야만 하며, 따라서 바이트열로 표현될 수 있다.

인증 태그의 비트 길이는 *Tlen*로 표기되며 인증의 강도를 나타내는 변수이다. 일반적으로 *Tlen*는 128, 120, 112, 104, 96 중에서 선택된다. 특별한 경우, *Tlen*가 64와 32가 선택될 수 있다.

5.2. 증가 함수

비트열 X와 $\text{len}(X) \geq s$를 만족하는 양수 s에 대하여, s-비트 증가 함수는 $\text{inc}_s(X)$로 표기하며 다음과 같이 규정된다.

$$\text{inc}_s(X) = \text{MSB}_{\text{len}(X)-s}(X) \parallel [\text{int}(\text{LSB}_s(X)) + 1 \bmod 2^s]_s$$

다시 말하면, 증가 함수는 비트열의 오른쪽 상위 s비트를 1만큼 증가시키는 함수이다. 왼쪽 상위 $\text{len}(X)-s$비트는 변하지 않는다.

5.3. 블록 간의 곱셈 연산

R을 $11100001 \parallel 0^{120}$이라 하자. 두 개의 블록 X와 Y가 주어지면 다음에 의하여 X와 Y의 “곱셈” $X \bullet Y$을 계산할 수 있다.

- o 입력 : 블록 X와 Y
- o 출력 : 블록 $X \bullet Y$

o 처리 단계 :

① 블록 X를 비트열 $x_0x_1...x_{127}$로 표기한다.

② $Z_0 = 0^{128}$와 $V_0 = Y$을 초기화한다.

③ $i = 0$부터 127까지 블록 Z_{i+1}와 V_{i+1}을 다음과 같이 계산한다.

$$Z_{i+1} = \begin{cases} Z_i & \text{if } x_i = 0; \\ Z_i \oplus V_i & \text{if } x_i = 1. \end{cases}$$

$$V_{i+1} = \begin{cases} V_i \gg 1 & \text{if } \mathrm{LSB}_1(V_i) = 0; \\ (V_i \gg 1) \oplus R & \text{if } \mathrm{LSB}_1(V_i) = 1. \end{cases}$$

④ Z_{128}을 출력한다.

5.4. GHASH 함수

다음은 GHASH 함수를 명시한다.

o 전제 조건

- 해시 서브 키 블록 H

o 입력

- 양수 m 에 대하여 $\mathrm{len}(X) = 128m$을 만족하는 비트열 X

o 출력

- 블록 $\mathrm{GHASH}_H(X)$

o 처리 단계 :

① 비트열 $X=X_1 \| X_2 \| X_3 \| \cdots \| X_{m-1} \| X_m$의 블록열로 $X_1, X_2, X_3, \cdots, X_{m-1}, X_m$으로 표기한다.

② Y_0를 0^{128}로 초기화한다.

③ $i = 1$ 부터 m 까지 다음을 계산한다.

$$Y_i = (Y_{i-1} \oplus X_i) \bullet H$$

④ Y_m을 출력한다.

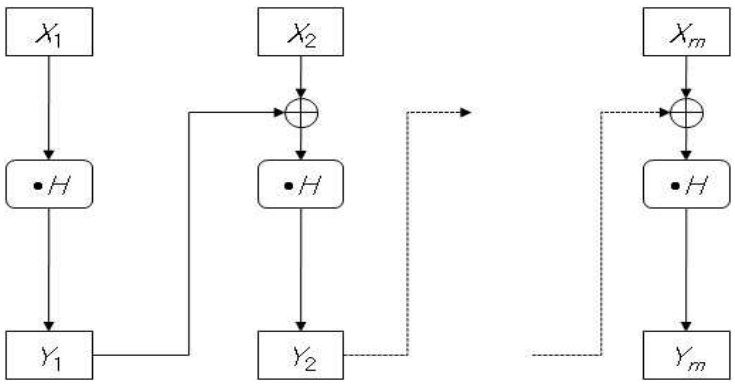

그림 5.1 GHASH 함수

5.5. GCTR 함수

다음은 GCTR 함수를 명시한다.

o 전제 조건
- 128 비트 블록 암호 알고리즘 CIPH
- 비밀 키 K

o 입력
- 초기 카운터 블록 ICB
- 임의 길이의 비트열 X

o 출력
- 비트 길이 len(X)를 만족하는 비트열 Y

o 처리 단계 :
① 만약 X가 빈 비트열이라면, Y로서 빈 비트열을 출력한다.
② $n = \lceil \text{len}(X)/128 \rceil$ 을 계산한다.
③ 비트열 $X = X_1 \| X_2 \| X_3 \| \cdots \| X_{n-1} \| X^*_n$의 블록열로 $X_1, X_2, X_3, \cdots, X_{n-1}, X^*_n$로 표기한다.
X_1, X_2, X_3, …, X_{n-1}은 128 비트로 구성된 완전한 데이터 블록이다.
④ $CB_1 = ICB$를 계산한다.
⑤ $i = 2$ 부터 n까지 다음을 계산한다.

$$CB_i = \text{inc}_{32}(CB_{i-1}).$$

⑥ $i = 1$ 부터 $n - 1$ 까지 다음을 계산한다.

$$Y_i = X_i \oplus \text{CIPH}_K(CB_i)$$

⑦ $Y^*_n = X^*_i \oplus \text{MSB}_{\text{len}(tmp)}(\text{CIPH}_K(CB_i))$, $tmp = X^*_n$를 계산한다.
⑧ $Y = Y_1 \| Y_2 \| \cdots \| Y^*_n$를 계산한다.
⑨ Y을 출력한다.

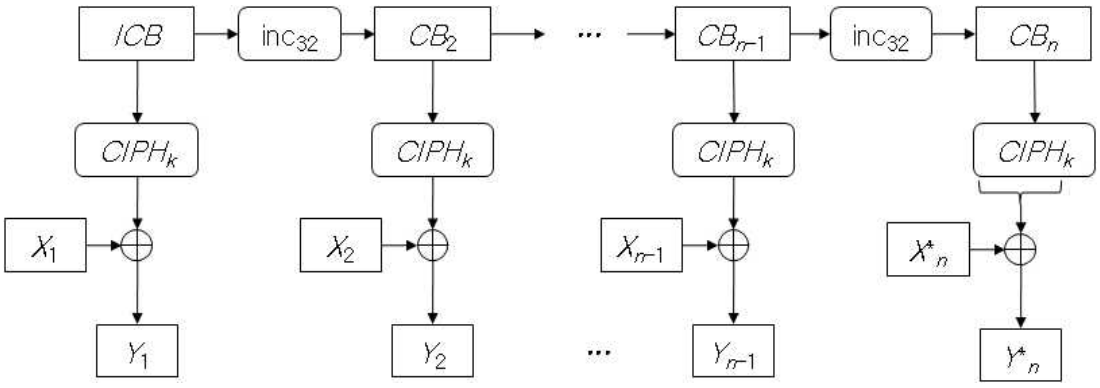

그림 5.2 GCTR 함수

5.6. GCM 모드 암호화

o 전제 조건
- 128 비트 블록 암호 알고리즘 CIPH
- 비밀 키 K
- 입·출력 데이터의 길이
- 인증 태그의 길이 $Tlen$

o 입력
- 초기 벡터 IV
- 평문 P
- 추가 인증 데이터 A

o 출력
- 암호문 C
- 인증 태그 T

o 처리 단계 :

① $H = \mathrm{CIPH}_K(0^{128})$을 계산한다.

② J_0를 다음과 같이 정의한다.
- 만약 $\mathrm{len}(IV) = 96$이면,
  $J_0 = IV \| 0^{31} \| 1$로 정의한다.
- 만약 $\mathrm{len}(IV) \neq 96$이면,
  $s = 128 \lceil \mathrm{len}(IV)/128 \rceil - \mathrm{len}(IV)$라 할 때,
  $J_0 = \mathrm{GHASH}_H(IV \| 0^{s+64} \| [\mathrm{len}(IV)]_{64})$로 정의한다.

③ $C = \mathrm{GCTR}_K(\mathrm{inc}_{32}(J_0), P)$를 계산한다.

④ $u = 128 \lceil \mathrm{len}(C)/128 \rceil - \mathrm{len}(C)$와 $v = 128 \lceil \mathrm{len}(A)/128 \rceil - \mathrm{len}(A)$를 계산한다.

⑤ 블록 S를 다음과 같이 계산한다.

$$S = \mathrm{GHASH}_H (A \| 0^v \| C \| 0^u \| [\mathrm{len}(A)]_{64} \| [\mathrm{len}(C)]_{64})$$

⑥ T를 다음과 같이 계산한다.

$$T = \mathrm{MSB}_{Tlen}(\mathrm{GCTR}_K(J_0, S))$$

⑦ (C, T)를 출력한다.

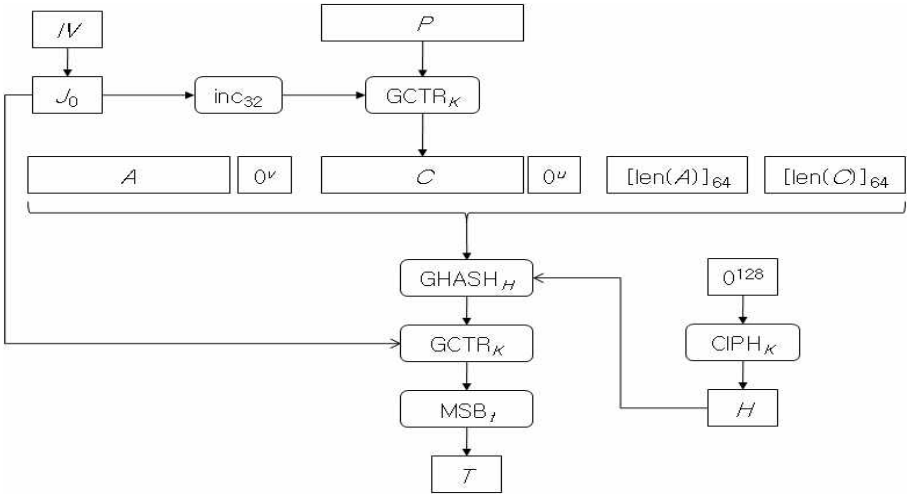

그림 5.3 GCM 모드 암호화

5.7. GCM 모드 복호화

o 전제 조건
- 128 비트 블록 암호 알고리즘 CIPH
- 비밀 키 K
- 입출력 데이터의 길이
- 인증 태그의 길이 $Tlen$

o 입력
- 초기 벡터 IV
- 암호문 C
- 추가 인증 데이터 A
- 인증 태그 T

o 출력
- 평문 P 혹은 $FAIL$

o 처리 단계 :
① IV, A, C가 정당한 길이를 갖지 않으면 $FAIL$을 출력한다.
② $H = CIPH_K(0^{128})$을 계산한다.
③ J_0를 다음과 같이 정의한다.
- 만약 $len(IV) = 96$이면,
  $J_0 = IV \| 0^{31} \| 1$로 정의한다.
- 만약 $len(IV) \neq 96$이면,
  $s = 128 \lceil len(IV)/128 \rceil - len(IV)$라 할 때,
  $J_0 = GHASH_H(IV \| 0^{s+64} \| [len(IV)]_{64})$로 정의한다.

④ P = $GCTR_K(inc_{32}(J_0), C)$를 계산한다.

⑤ u = 128 ⌈ len(C)/128 ⌉ −len(C)와 v=128 ⌈ len(A)/128 ⌉ −len(A)를 계산한다.

⑥ 블록 S를 다음과 같이 계산한다.

$$S = GHASH_H (A \| 0^v \| C \| 0^u \| [len(A)]_{64} \| [len(C)]_{64})$$

⑦ T'를 다음과 같이 계산한다.

$$T' = MSB_{Tlen}(GCTR_K(J_0, S))$$

⑧ 만약 $T = T'$를 만족하면 P를 출력하고, 그렇지 않으면 $FAIL$을 출력한다.

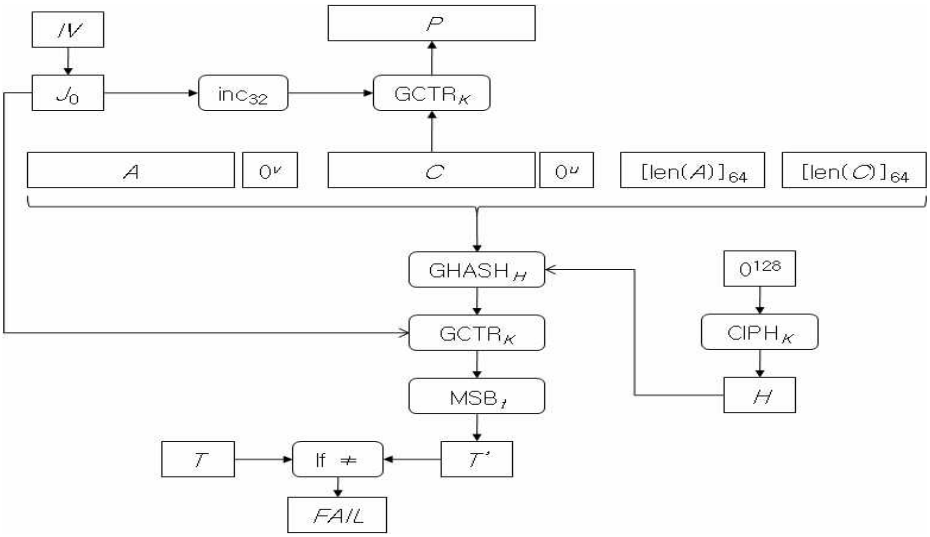

그림 5.4 GCM 모드 복호화

부 록 Ⅰ
CCM 운영 모드 참조 구현값(SEED)

본 부록에서는 기밀성과 메시지 인증을 제공하는 128 비트 블록 암호 알고리즘 운영 모드 표준의 CCM 운영 모드 구현 적합성 실험을 위한 참조 구현값(Test Vectors)이 제공된다.

CCM 모드의 기반이 되는 블록 암호는 128 비트 블록 암호 알고리즘 SEED이다. 인증 태그(*T*), 난스(*N*), 연관 데이터(*A*), 평문(*P*)은 각각의 예에서 다양한 길이를 갖는다.

Ⅰ.1. 참조 구현값 1 - Klen=128, Tlen=32, Nlen=56, Alen=64, Plen=32

Ⅰ.1.1. CCM 모드 암호화

Ⅰ.1.1.1. CCM 모드 암호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 141516
A	00010203 04050607
P	20212223

Ⅰ.1.1.2. 입력 데이터 포맷화 값

B	4f101112 13141516 00000000 00000004 00080001 02030405 06070000 00000000 20212223 00000000 00000000 00000000

Ⅰ.1.1.3. 데이터 블록 B를 이용한 인증 태그 값 계산 과정

B_0	4f101112 13141516 00000000 00000004
SEED입력	4f101112 13141516 00000000 00000004
SEED출력	2b9f7b76 3f7a75ff 2bd34f06 d4c8632a
B_1	00080001 02030405 06070000 00000000
SEED입력	2b977b77 3d7971fa 2dd44f06 d4c8632a
SEED출력	637d097c ad30975f 6ee2fc1b d0226a8f
B_2	20212223 00000000 00000000 00000000
SEED입력	435c2b5f ad30975f 6ee2fc1b d0226a8f
SEED출력	7e3ebe95 bbb6f93c 8fa3b616 3150c77c
T	7e3ebe95

Ⅰ.1.1.4. 평문과 인증 태그 값을 이용한 암호문 계산 과정

Ctr_0	07101112 13141516 00000000 00000000
SEED출력	b485a67b 7715b5ef 1279f66d d01cf1d8
Ctr_1	07101112 13141516 00000000 00000001
SEED출력	3c0c222a df28fec5 1ae2ce3b e208a149

Ⅰ.1.1.5. CCM 모드 암호화의 출력 데이터

C	1c2d0009 cabb18ee

Ⅰ.1.2. CCM 모드 복호화

Ⅰ.1.2.1. CCM 모드 복호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 141516
A	00010203 04050607
C	1c2d0009 cabb18ee

Ⅰ.1.2.2. 암호문을 이용한 평문 계산 과정

Ctr_0	07101112 13141516 00000000 00000000
SEED출력	b485a67b 7715b5ef 1279f66d d01cf1d8
Ctr_1	07101112 13141516 00000000 00000001
SEED출력	3c0c222a df28fec5 1ae2ce3b e208a149

Ⅰ.1.2.3. 암호문으로부터 복구한 평문을 이용한 인증 태그 값 계산

B	4f101112 13141516 00000000 00000004 00080001 02030405 06070000 00000000 20212223 00000000 00000000 00000000

B_0	4f101112 13141516 00000000 00000004
SEED입력	4f101112 13141516 00000000 00000004
SEED출력	2b9f7b76 3f7a75ff 2bd34f06 d4c8632a
B_1	00080001 02030405 06070000 00000000
SEED입력	2b977b77 3d7971fa 2dd44f06 d4c8632a
SEED출력	637d097c ad30975f 6ee2fc1b d0226a8f
B_2	20212223 00000000 00000000 00000000
SEED입력	435c2b5f ad30975f 6ee2fc1b d0226a8f
SEED출력	7e3ebe95 bbb6f93c 8fa3b616 3150c77c
T	7e3ebe95

Ⅰ.1.1.4 CCM 모드 복호화의 출력 데이터

P	20212223

Ⅰ.2. 참조 구현값 2 - Klen=128, Tlen=48, Nlen=64, Alen=128, Plen=128

Ⅰ.2.1. CCM 모드 암호화

Ⅰ.2.1.1. CCM 모드 암호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 14151617
A	00010203 04050607 08090a0b 0c0d0e0f
P	20212223 24252627 28292a2b 2c2d2e2f

Ⅰ.2.1.2. 입력 데이터 포맷화 값

B	56101112 13141516 17000000 00000010 00100001 02030405 06070809 0a0b0c0d 0e0f0000 00000000 00000000 00000000 20212223 24252627 28292a2b 2c2d2e2f

Ⅰ.2.1.3. 데이터 블록 B을 이용한 인증 태그 값 계산 과정

B_0	56101112 13141516 17000000 00000010
SEED입력	56101112 13141516 17000000 00000010
SEED출력	24d3583c bceba260 e31e726a fa2094b8
B_1	00100001 02030405 06070809 0a0b0c0d
SEED입력	24c3583d bee8a665 e5197a63 f02b98b5
SEED출력	4229cff0 04b283f1 52adc3cf ae770a06
B_2	0e0f0000 00000000 00000000 00000000
SEED입력	4c26cff0 04b283f1 52adc3cf ae770a06
SEED출력	863e8b8d a42f2753 81454e98 c962fa53
B_3	20212223 24252627 28292a2b 2c2d2e2f
SEED입력	a61fa9ae 800a0174 a96c64b3 e54fd47c
SEED출력	7c646686 ed1af0e0 d85f38ff c791d78a
T	7c646686 ed1a

Ⅰ.2.1.4. 평문과 인증 태그 값을 이용한 암호문 계산 과정

Ctr_0	06101112 13141516 17000000 00000000
SEED출력	6b6dece3 4576ae70 c6f3b52f c21ef254
Ctr_1	06101112 13141516 17000000 00000001
SEED출력	eb9089a9 2a87a2a9 9d0fb7f3 b83ded3a

Ⅰ.2.1.5. CCM 모드 암호화의 출력 데이터

C	cbb1ab8a 0ea2848e b5269dd8 9410c315 17098a65 a86c

Ⅰ.2.2. CCM 모드 복호화

Ⅰ.2.2.1. CCM 모드 복호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 14151617
A	00010203 04050607 08090a0b 0c0d0e0f
C	cbb1ab8a 0ea2848e b5269dd8 9410c315 17098a65 a86c

Ⅰ.2.2.2. 암호문을 이용한 평문 계산 과정

Ctr_0	06101112 13141516 17000000 00000000
SEED출력	6b6dece3 4576ae70 c6f3b52f c21ef254
Ctr_1	06101112 13141516 17000000 00000001
SEED출력	eb9089a9 2a87a2a9 9d0fb7f3 b83ded3a

Ⅰ.2.2.3. 암호문으로부터 복구한 평문을 이용한 인증 태그 값 계산

B	56101112 13141516 17000000 00000010 00100001 02030405 06070809 0a0b0c0d 0e0f0000 00000000 00000000 00000000 20212223 24252627 28292a2b 2c2d2e2f

B_0	56101112 13141516 17000000 00000010
SEED입력	56101112 13141516 17000000 00000010
SEED출력	24d3583c bceba260 e31e726a fa2094b8
B_1	00100001 02030405 06070809 0a0b0c0d
SEED입력	24c3583d bee8a665 e5197a63 f02b98b5
SEED출력	4229cff0 04b283f1 52adc3cf ae770a06
B_2	0e0f0000 00000000 00000000 00000000
SEED입력	4c26cff0 04b283f1 52adc3cf ae770a06
SEED출력	863e8b8d a42f2753 81454e98 c962fa53
B_3	20212223 24252627 28292a2b 2c2d2e2f
SEED입력	a61fa9ae 800a0174 a96c64b3 e54fd47c
SEED출력	7c646686 ed1af0e0 d85f38ff c791d78a
T	7c646686 ed1a

Ⅰ.2.1.4. CCM 모드 복호화의 출력 데이터

P	20212223 24252627 28292a2b 2c2d2e2f

Ⅰ.3. 참조 구현값 3 - Klen=128, Tlen=64, Nlen=96, Alen=160, Plen=192

Ⅰ.3.1. CCM 모드 암호화

Ⅰ.3.1.1. CCM 모드 암호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 14151617 18191a1b
A	00010203 04050607 08090a0b 0c0d0e0f 10111213
P	20212223 24252627 28292a2b 2c2d2e2f 30313233 34353600

Ⅰ.3.1.2. 입력 데이터 포맷화 값

B	5a101112 13141516 1718191a 1b000018 00140001 02030405 06070809 0a0b0c0d 0e0f1011 12130000 00000000 00000000 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353600 00000000 00000000

Ⅰ.3.1.3. 데이터 블록 B를 이용한 인증 태그 값 계산 과정

B_0	5a101112 13141516 1718191a 1b000018
SEED입력	5a101112 13141516 1718191a 1b000018
SEED출력	b60c31dd 0c1090a7 190fcd82 47ceb3c2
B_1	00140001 02030405 06070809 0a0b0c0d
SEED입력	b61831dc 0e1394a2 1f08c58b 4dc5bfcf
SEED출력	5d0996a2 13f6a0aa 22e77daf d4419eff
B_2	0e0f1011 12130000 00000000 00000000
SEED입력	530686b3 01e5a0aa 22e77daf d4419eff
SEED출력	ebfa0c15 771190f6 0b8d365c c093e097
B_3	20212223 24252627 28292a2b 2c2d2e2f
SEED입력	cbdb2e36 5334b6d1 23a41c77 ecbeceb8
SEED출력	27892df2 babdeda5 83fc0481 97b68309
B_4	30313233 34353600 00000000 00000000
SEED입력	17b81fc1 8e88dba5 83fc0481 97b68309
SEED출력	dcdcea6a b82c5dbe 56de3bfe 5631aa65
T	dcdcea6a b82c5dbe

Ⅰ.3.1.4. 평문과 인증 태그 값을 이용한 암호문 계산 과정

Ctr_0	02101112 13141516 1718191a 1b000000
SEED출력	682b1a22 ba0164d5 c4987a1a b7e087cb
Ctr_1	02101112 13141516 1718191a 1b000001
SEED출력	81318ba2 70d0a16f 3ec7c2cf 76f16b56
Ctr_2	02101112 13141516 1718191a 1b000002
SEED출력	ad4ac71d 7fcacbaf a0a6f9f7 fe3e2997

Ⅰ.3.1.5. CCM 모드 암호화의 출력 데이터

C	a110a981 54f58748 16eee8e4 5adc4579 9d7bf52e 4bfffdaf b4f7f048 022d396b

Ⅰ.3.2. CCM 모드 복호화

Ⅰ.3.2.1. CCM 모드 복호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 14151617 18191a1b
A	00010203 04050607 08090a0b 0c0d0e0f 10111213
C	a110a981 54f58748 16eee8e4 5adc4579 9d7bf52e 4bfffdaf b4f7f048 022d396b

Ⅰ.3.2.2. 암호문을 이용한 평문 계산 과정

Ctr_0	02101112 13141516 1718191a 1b000000
SEED출력	682b1a22 ba0164d5 c4987a1a b7e087cb
Ctr_1	02101112 13141516 1718191a 1b000001
SEED출력	81318ba2 70d0a16f 3ec7c2cf 76f16b56
Ctr_2	02101112 13141516 1718191a 1b000002
SEED출력	ad4ac71d 7fcacbaf a0a6f9f7 fe3e2997

Ⅰ.3.2.3. 암호문으로부터 복구한 평문을 이용한 인증 태그 값 계산

B	5a101112 13141516 1718191a 1b000018 00140001 02030405 06070809 0a0b0c0d 0e0f1011 12130000 00000000 00000000 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353600 00000000 00000000

B_0	5a101112 13141516 1718191a 1b000018
SEED입력	5a101112 13141516 1718191a 1b000018
SEED출력	b60c31dd 0c1090a7 190fcd82 47ceb3c2
B_1	00140001 02030405 06070809 0a0b0c0d
SEED입력	b61831dc 0e1394a2 1f08c58b 4dc5bfcf
SEED출력	5d0996a2 13f6a0aa 22e77daf d4419eff
B_2	0e0f1011 12130000 00000000 00000000
SEED입력	530686b3 01e5a0aa 22e77daf d4419eff
SEED출력	ebfa0c15 771190f6 0b8d365c c093e097
B_3	20212223 24252627 28292a2b 2c2d2e2f
SEED입력	cbdb2e36 5334b6d1 23a41c77 ecbeceb8
SEED출력	27892df2 babdeda5 83fc0481 97b68309
B_4	30313233 34353600 00000000 00000000
SEED입력	17b81fc1 8e88dba5 83fc0481 97b68309
SEED출력	dcdcea6a b82c5dbe 56de3bfe 5631aa65
T	dcdcea6a b82c5dbe

Ⅰ.3.1.4. CCM 모드 복호화의 출력 데이터

P	20212223 24252627 28292a2b 2c2d2e2f 30313233 34353600

Ⅰ.4. 참조 구현값 4 - Klen=128, Tlen=32, Nlen=56, Alen=0, Plen=512

Ⅰ.4.1. CCM 모드 암호화

Ⅰ.4.1.1. CCM 모드 암호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 141516
A	없음
P	20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f 40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f

Ⅰ.4.1.2. 입력 데이터 포맷화 값

B	0f101112 13141516 00000000 00000040 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f 40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f

Ⅰ.4.1.3. 데이터 블록 B를 이용한 인증 태그 값 계산 과정

B_0	0f101112 13141516 00000000 00000040
SEED입력	0f101112 13141516 00000000 00000040
SEED출력	75b1360e 406b45d2 ea927f84 3f33d3c6
B_1	20212223 24252627 28292a2b 2c2d2e2f
SEED입력	5590142d 644e63f5 c2bb55af 131efde9
SEED출력	b4de8df3 33271f91 b1f8ed31 8bf9e18c
B_2	30313233 34353637 38393a3b 3c3d3e3f
SEED입력	84efbfc0 071229a6 89c1d70a b7c4dfb3
SEED출력	15495955 98ed6335 4057ecc9 8e626b3f
B_3	40414243 44454647 48494a4b 4c4d4e4f
SEED입력	55081b16 dca82572 081ea682 c22f2570
SEED출력	cb7a9030 14899b06 e2f30f64 fc4cae45
B_4	50515253 54555657 58595a5b 5c5d5e5f
SEED입력	9b2bc263 40dccd51 baaa553f a011f01a
SEED출력	5ec6a539 380db2e0 0687f052 99eee65c
T	5ec6a539

Ⅰ.4.1.4. 평문과 인증 태그 값을 이용한 암호문 계산 과정

Ctr_0	07101112 13141516 00000000 00000000
SEED출력	b485a67b 7715b5ef 1279f66d d01cf1d8
Ctr_1	07101112 13141516 00000000 00000001
SEED출력	3c0c222a df28fec5 1ae2ce3b e208a149
Ctr_2	07101112 13141516 00000000 00000002
SEED출력	6b32fabe 6768d525 85f251b1 1ba8b898
Ctr_3	07101112 13141516 00000000 00000003
SEED출력	bed50cb5 ec0a92ba c5c163a7 70dd45bb
Ctr_4	07101112 13141516 00000000 00000004
SEED출력	1bd1a0b4 d4be0492 a4af19eb 59111d1c

Ⅰ.4.1.5. CCM 모드 암호화의 출력 데이터

C	1c2d0009 fb0dd8e2 32cbe410 ce258f66 5b03c88d 535de312 bdcb6b8a 279586a7 fe944ef6 a84fd4fd 8d8829ec 3c900bf4 4b80f2e7 80eb52c5 fcf643b0 054c4343 ea430342

Ⅰ.4.2. CCM 모드 복호화

Ⅰ.4.2.1. CCM 모드 복호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 141516
A	없음
C	1c2d0009 fb0dd8e2 32cbe410 ce258f66 5b03c88d 535de312 bdcb6b8a 279586a7 fe944ef6 a84fd4fd 8d8829ec 3c900bf4 4b80f2e7 80eb52c5 fcf643b0 054c4343 ea430342

Ⅰ.4.2.2. 암호문을 이용한 평문 계산 과정

Ctr_0	07101112 13141516 00000000 00000000
SEED출력	b485a67b 7715b5ef 1279f66d d01cf1d8
Ctr_1	07101112 13141516 00000000 00000001
SEED출력	3c0c222a df28fec5 1ae2ce3b e208a149
Ctr_2	07101112 13141516 00000000 00000002
SEED출력	6b32fabe 6768d525 85f251b1 1ba8b898
Ctr_3	07101112 13141516 00000000 00000003
SEED출력	bed50cb5 ec0a92ba c5c163a7 70dd45bb
Ctr_4	07101112 13141516 00000000 00000004
SEED출력	1bd1a0b4 d4be0492 a4af19eb 59111d1c

Ⅰ.4.2.3. 암호문으로부터 복구한 평문을 이용한 인증 태그 값 계산

B	0f101112 13141516 00000000 00000040 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f 40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f

B_0	0f101112 13141516 00000000 00000040
SEED입력	0f101112 13141516 00000000 00000040
SEED출력	75b1360e 406b45d2 ea927f84 3f33d3c6
B_1	20212223 24252627 28292a2b 2c2d2e2f
SEED입력	5590142d 644e63f5 c2bb55af 131efde9
SEED출력	b4de8df3 33271f91 b1f8ed31 8bf9e18c
B_2	30313233 34353637 38393a3b 3c3d3e3f
SEED입력	84efbfc0 071229a6 89c1d70a b7c4dfb3
SEED출력	15495955 98ed6335 4057ecc9 8e626b3f
B_3	40414243 44454647 48494a4b 4c4d4e4f
SEED입력	55081b16 dca82572 081ea682 c22f2570
SEED출력	cb7a9030 14899b06 e2f30f64 fc4cae45
B_4	50515253 54555657 58595a5b 5c5d5e5f
SEED입력	9b2bc263 40dccd51 baaa553f a011f01a
SEED출력	5ec6a539 380db2e0 0687f052 99eee65c
T	5ec6a539

Ⅰ.4.1.4. CCM 모드 복호화의 출력 데이터

P	20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f 40414243 44454647 48494a4b 4c4d4e4f 50515253 54555657 58595a5b 5c5d5e5f

Ⅰ.5. 참조 구현값 5 - Klen=128, Tlen=32, Nlen=56, Alen=512, Plen=0

Ⅰ.5.1. CCM 모드 암호화

Ⅰ.5.1.1. CCM 모드 암호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 141516
A	00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
P	없음

Ⅰ.5.1.2. 입력 데이터 포맷화 값

B	4f101112 13141516 00000000 00000000 00400001 02030405 06070809 0a0b0c0d 0e0f1011 12131415 16171819 1a1b1c1d 1e1f2021 22232425 26272829 2a2b2c2d 2e2f3031 32333435 36373839 3a3b3c3d 3e3f0000 00000000 00000000 00000000

Ⅰ.5.1.3. 데이터 블록 B를 이용한 인증 태그 값 계산 과정

B_0	4f101112 13141516 00000000 00000000
SEED입력	4f101112 13141516 00000000 00000000
SEED출력	f028bf35 7f5eb238 ffb0dcb1 dee79347
B_1	00400001 02030405 06070809 0a0b0c0d
SEED입력	f068bf34 7d5db63d f9b7d4b8 d4ec9f4a
SEED출력	e4d4f118 a791a7d2 4e68ab97 c2371365
B_2	0e0f1011 12131415 16171819 1a1b1c1d
SEED입력	eadbe109 b582b3c7 587fb38e d82c0f78
SEED출력	45d558f1 64632ecf b87131f9 91ef3181
B_3	1e1f2021 22232425 26272829 2a2b2c2d
SEED입력	5bca78d0 46400aea 9e5619d0 bbc41dac
SEED출력	6719b538 54e1a459 f6e7a9f0 606c37a8
B_4	2e2f3031 32333435 36373839 3a3b3c3d
SEED입력	49368509 66d2906c c0d091c9 5a570b95
SEED출력	88ef3566 b11a126c 24a64b90 1bfaf057
B_5	3e3f0000 00000000 00000000 00000000
SEED입력	b6d03566 b11a126c 24a64b90 1bfaf057
SEED출력	e73ba875 f3471403 3a7e086b 78765a23
T	e73ba875

Ⅰ.5.1.4. 평문과 인증 태그 값을 이용한 암호문 계산 과정

Ctr_0	07101112 13141516 00000000 00000000
SEED출력	b485a67b 7715b5ef 1279f66d d01cf1d8

Ⅰ.5.1.5. CCM 모드 암호화의 출력 데이터

C	53be0e0e

Ⅰ.5.2. CCM 모드 복호화

Ⅰ.5.2.1. CCM 모드 복호화의 입력 데이터

K	40414243 44454647 48494a4b 4c4d4e4f
N	10111213 141516
A	00010203 04050607 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637 38393a3b 3c3d3e3f
C	53be0e0e

Ⅰ.5.2.2. 암호문을 이용한 평문 계산 과정

Ctr_0	07101112 13141516 00000000 00000000
SEED출력	b485a67b 7715b5ef 1279f66d d01cf1d8

Ⅰ.5.2.3. 암호문으로부터 복구한 평문을 이용한 인증 태그 값 계산

B	4f101112 13141516 00000000 00000000 00400001 02030405 06070809 0a0b0c0d 0e0f1011 12131415 16171819 1a1b1c1d 1e1f2021 22232425 26272829 2a2b2c2d 2e2f3031 32333435 36373839 3a3b3c3d 3e3f0000 00000000 00000000 00000000

B_0	4f101112 13141516 00000000 00000000
SEED입력	4f101112 13141516 00000000 00000000
SEED출력	f028bf35 7f5eb238 ffb0dcb1 dee79347
B_1	00400001 02030405 06070809 0a0b0c0d
SEED입력	f068bf34 7d5db63d f9b7d4b8 d4ec9f4a
SEED출력	e4d4f118 a791a7d2 4e68ab97 c2371365
B_2	0e0f1011 12131415 16171819 1a1b1c1d
SEED입력	eadbe109 b582b3c7 587fb38e d82c0f78
SEED출력	45d558f1 64632ecf b87131f9 91ef3181
B_3	1e1f2021 22232425 26272829 2a2b2c2d
SEED입력	5bca78d0 46400aea 9e5619d0 bbc41dac
SEED출력	6719b538 54e1a459 f6e7a9f0 606c37a8
B_4	2e2f3031 32333435 36373839 3a3b3c3d
SEED입력	49368509 66d2906c c0d091c9 5a570b95
SEED출력	88ef3566 b11a126c 24a64b90 1bfaf057
B_5	3e3f0000 00000000 00000000 00000000
SEED입력	b6d03566 b11a126c 24a64b90 1bfaf057
SEED출력	e73ba875 f3471403 3a7e086b 78765a23
T	e73ba875

Ⅰ.5.1.4. CCM 모드 복호화의 출력 데이터

P	없음

부 록 II
GCM 운영 모드 참조 구현값(SEED)

본 부록에서는 기밀성과 메시지 인증을 제공하는 128 비트 블록 암호 알고리즘 운영 모드 표준의 GCM 운영 모드 구현 적합성 실험을 위한 참조 구현값(Test Vectors)이 제공된다.

GCM 모드의 기반이 되는 블록 암호는 128 비트 블록 암호 알고리즘 SEED이다. 인증 태그(T), 초기 벡터(IV), 추가 인증 데이터(A), 평문(P)은 각각의 예에서 다양한 길이를 갖는다.

II.1. 참조 구현값 1 - Klen=128, Tlen=128, IVlen=96, Alen=0, Plen=0

II.1.1. GCM 모드 암호화

II.1.1.1. GCM 모드 암호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	없음
P	없음

II.1.1.2. 평문을 이용한 암호문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

II.1.1.3. 키를 이용한 H값 계산 과정

H	addab0a6 958b6567 19702b91 73e3dbb4

II.1.1.3. 추가 인증 데이터와 암호문을 이용한 인증 태그 값 계산 과정

$GHASH_H$	00000000 00000000 00000000 00000000	
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000001
	SEED출력	cb99d743 4d4d1962 7026c832 4d5523f9
	T	cb99d743 4d4d1962 7026c832 4d5523f9

II.1.1.4. GCM 모드 암호화의 출력 데이터

C	없음
T	cb99d743 4d4d1962 7026c832 4d5523f9

II.1.2. GCM 모드 복호화

II.1.2.1. GCM 모드 복호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	없음
C	없음
T	cb99d743 4d4d1962 7026c832 4d5523f9

II.1.2.2. 암호문을 이용한 평문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

II.1.2.3. GCM 모드 복호화의 출력 데이터

P	없음

II.2. 참조 구현값 2 - Klen=128, Tlen=128, IVlen=96, Alen=0, Plen=512

II.2.1. GCM 모드 암호화

II.2.1.1. GCM 모드 암호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	없음
P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39 1aafd255

II.2.1.2. 평문을 이용한 암호문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	C_0	1a43abdb 0b01d1e7 d8003851 d6ecf7d6
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	C_1	7e3efb04 1ab192be 81e4fe67 b14e863f
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	C_2	779851bf d993c2a1 9f8cde02 1f6962a5
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	C_3	c4d27ad5 5e5d16e0 fd737170 fb8329ea

II.2.1.3. 키를 이용한 H값 계산 과정

H	addab0a6 958b6567 19702b91 73e3dbb4

II.2.1.3. 추가 인증 데이터와 암호문을 이용한 인증 태그 값 계산 과정

$GHASH_H$	9ec30bad c2ab68e2 307dd374 13b23ca1	
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000001
	SEED출력	cb99d743 4d4d1962 7026c832 4d5523f9
	T	555adcee 8fe67180 405b1b46 5ee71f58

II.2.1.4. GCM 모드 암호화의 출력 데이터

C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170 fb8329ea
T	555adcee 8fe67180 405b1b46 5ee71f58

II.2.2. GCM 모드 복호화

II.2.2.1. GCM 모드 복호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	없음
C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170 fb8329ea
T	555adcee 8fe67180 405b1b46 5ee71f58

II.2.2.2. 암호문을 이용한 평문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	P_0	d9313225 f88406e5 a55909c5 aff5269a
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	P_1	86a7a953 1534f7da 2e4c303d 8a318a72
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	P_2	1c3c0c95 95680953 2fcf0e24 49a6b525
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	P_3	b16aedf5 aa0de657 ba637b39 1aafd255

II.2.2.3. GCM 모드 복호화의 출력 데이터

P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39 1aafd255

II.3. 참조 구현값 3 - Klen=128, Tlen=128, IVlen=96, Alen=512, Plen=0

II.3.1. GCM 모드 암호화

II.3.1.1. GCM 모드 암호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585 03b9699d e785895a 96fdbaaf 43b1cd7f 598ece23 881b00e3 ed030688 7b0c785e 27e8ad3f 82232071 04725dd4
P	없음

II.3.1.2. 평문을 이용한 암호문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

II.3.1.3. 키를 이용한 H값 계산 과정

H	addab0a6 958b6567 19702b91 73e3dbb4

II.3.1.3. 추가 인증 데이터와 암호문을 이용한 인증 태그 값 계산 과정

$GHASH_H$	fda5da72 cc53c3cb 84e814b4 5c993ac7	
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000001
	SEED출력	cb99d743 4d4d1962 7026c832 4d5523f9
	T	363c0d31 811edaa9 f4cedc86 11cc193e

II.2.1.4. GCM 모드 암호화의 출력 데이터

C	없음
T	363c0d31 811edaa9 f4cedc86 11cc193e

II.3.2. GCM 모드 복호화

II.3.2.1. GCM 모드 복호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585 03b9699d e785895a 96fdbaaf 43b1cd7f 598ece23 881b00e3 ed030688 7b0c785e 27e8ad3f 82232071 04725dd4
C	없음
T	363c0d31 811edaa9 f4cedc86 11cc193e

II.3.2.2. 암호문을 이용한 평문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

II.3.2.3. GCM 모드 복호화의 출력 데이터

P	없음

II.4. 참조 구현값 4 - Klen=128, Tlen=128, IVlen=96, Alen=512, Plen=512

II.4.1. GCM 모드 암호화

II.4.1.1. GCM 모드 암호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585 03b9699d e785895a 96fdbaaf 43b1cd7f 598ece23 881b00e3 ed030688 7b0c785e 27e8ad3f 82232071 04725dd4
P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39 1aafd255

II.4.1.2. 평문을 이용한 암호문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	C_0	1a43abdb 0b01d1e7 d8003851 d6ecf7d6
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	C_1	7e3efb04 1ab192be 81e4fe67 b14e863f
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	C_2	779851bf d993c2a1 9f8cde02 1f6962a5
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	C_3	c4d27ad5 5e5d16e0 fd737170 fb8329ea

II.4.1.3. 키를 이용한 H값 계산 과정

H	addab0a6 958b6567 19702b91 73e3dbb4

II.4.1.3. 추가 인증 데이터와 암호문을 이용한 인증 태그 값 계산 과정

$GHASH_H$	5c7361e8 5d984fef b22335e0 9efc3129	
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000001
	SEED출력	cb99d743 4d4d1962 7026c832 4d5523f9
	T	97eab6ab 10d5568d c205fdd2 d3a912d0

II.4.1.4. GCM 모드 암호화의 출력 데이터

C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170 fb8329ea
T	97eab6ab 10d5568d c205fdd2 d3a912d0

Ⅱ.4.2. GCM 모드 복호화

Ⅱ.4.2.1. GCM 모드 복호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585 03b9699d e785895a 96fdbaaf 43b1cd7f 598ece23 881b00e3 ed030688 7b0c785e 27e8ad3f 82232071 04725dd4
C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170 fb8329ea
T	97eab6ab 10d5568d c205fdd2 d3a912d0

Ⅱ.4.2.2. 암호문을 이용한 평문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	P_0	d9313225 f88406e5 a55909c5 aff5269a
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	P_1	86a7a953 1534f7da 2e4c303d 8a318a72
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	P_2	1c3c0c95 95680953 2fcf0e24 49a6b525
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	P_3	b16aedf5 aa0de657 ba637b39 1aafd255

Ⅱ.4.2.3. GCM 모드 복호화의 출력 데이터

P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39 1aafd255

II.5. 참조 구현값 5 - Klen=128, Tlen=128, IVlen=96, Alen=160, Plen=512

II.5.1. GCM 모드 암호화

II.5.1.1. GCM 모드 암호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585
P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39

II.5.1.2. 평문을 이용한 암호문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	C_0	1a43abdb 0b01d1e7 d8003851 d6ecf7d6
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	C_1	7e3efb04 1ab192be 81e4fe67 b14e863f
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	C_2	779851bf d993c2a1 9f8cde02 1f6962a5
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	C_3	c4d27ad5 5e5d16e0 fd737170

II.5.1.3. 키를 이용한 H값 계산 과정

H	addab0a6 958b6567 19702b91 73e3dbb4

II.5.1.3. 추가 인증 데이터와 암호문을 이용한 인증 태그 값 계산 과정

$GHASH_H$	371f5691 eb6587df b91a5eef c7472e68	
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000001
	SEED출력	cb99d743 4d4d1962 7026c832 4d5523f9
	T	fc8681d2 a6289ebd c93c96dd 8a120d91

II.5.1.4. GCM 모드 암호화의 출력 데이터

C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170
T	fc8681d2 a6289ebd c93c96dd 8a120d91

II.5.2. GCM 모드 복호화

II.5.2.1. GCM 모드 복호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585
C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170
T	fc8681d2 a6289ebd c93c96dd 8a120d91

II.5.2.2. 암호문을 이용한 평문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	P_0	d9313225 f88406e5 a55909c5 aff5269a
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	P_1	86a7a953 1534f7da 2e4c303d 8a318a72
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	P_2	1c3c0c95 95680953 2fcf0e24 49a6b525
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	P_3	b16aedf5 aa0de657 ba637b39

II.5.2.3. GCM 모드 복호화의 출력 데이터

P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39

II.6. 참조 구현값 6 - Klen=128, Tlen=96, IVlen=96, Alen=160, Plen=512

II.6.1. GCM 모드 암호화

II.6.1.1. GCM 모드 암호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585
P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39

II.6.1.2. 평문을 이용한 암호문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	C_0	1a43abdb 0b01d1e7 d8003851 d6ecf7d6
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	C_1	7e3efb04 1ab192be 81e4fe67 b14e863f
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	C_2	779851bf d993c2a1 9f8cde02 1f6962a5
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	C_3	c4d27ad5 5e5d16e0 fd737170

II.5.1.3. 키를 이용한 H값 계산 과정

H	addab0a6 958b6567 19702b91 73e3dbb4

II.6.1.3. 추가 인증 데이터와 암호문을 이용한 인증 태그 값 계산 과정

$GHASH_H$	371f5691 eb6587df b91a5eef c7472e68	
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000001
	SEED출력	cb99d743 4d4d1962 7026c832 4d5523f9
	T	fc8681d2 a6289ebd c93c96dd

II.6.1.4. GCM 모드 암호화의 출력 데이터

C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170
T	fc8681d2 a6289ebd c93c96dd

II.6.2. GCM 모드 복호화

II.6.2.1. GCM 모드 복호화의 입력 데이터

K	feffe992 8665731c 6d6a8f94 67308308
IV	cafebabe facedbad decaf888
A	3ad77bb4 0d7a3660 a89ecaf3 2466ef97 f5d3d585
C	1a43abdb 0b01d1e7 d8003851 d6ecf7d6 7e3efb04 1ab192be 81e4fe67 b14e863f 779851bf d993c2a1 9f8cde02 1f6962a5 c4d27ad5 5e5d16e0 fd737170
T	fc8681d2 a6289ebd c93c96dd

II.6.2.2. 암호문을 이용한 평문 계산 과정

J_0	cafebabe facedbad decaf888 00000001

$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000002
	SEED출력	c37299fe f385d702 7d593194 7919d14c
	P_0	d9313225 f88406e5 a55909c5 aff5269a
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000003
	SEED출력	f8995257 0f856564 afa8ce5a 3b7f0c4d
	P_1	86a7a953 1534f7da 2e4c303d 8a318a72
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000004
	SEED출력	6ba45d2a 4cfbcbf2 b043d026 56cfd780
	P_2	1c3c0c95 95680953 2fcf0e24 49a6b525
$GCTR_K$	SEED입력	cafebabe facedbad decaf888 00000005
	SEED출력	75b89720 f450f0b7 47100a49 e12cfbbf
	P_3	b16aedf5 aa0de657 ba637b39

II.6.2.3. GCM 모드 복호화의 출력 데이터

P	d9313225 f88406e5 a55909c5 aff5269a 86a7a953 1534f7da 2e4c303d 8a318a72 1c3c0c95 95680953 2fcf0e24 49a6b525 b16aedf5 aa0de657 ba637b39

부 록 III

관련 문헌

다음 문서들은 본 표준의 이해를 돕기 위한 문서로서 특정 문서(발행일 및 판 번호 또는 개정 번호를 명시한 것)와 일반 문서로 구별된다.

- 특정 문서인 경우 해당 판본 이후의 개정판은 적용되지 않는다.
- 일반 문서인 경우 최신 판본이 적용된다.

[1] ISO/IEC 19772, 'Information technology -- Security techniques -- Authenticated encryption', 2009.
[2] KS X 1213-2, '128비트 블록 암호 알고리즘 ARIA - 제2부 : 운영 모드', 2009.
[3] TTAE.IF-RFC3610 , '블록 암호 운영 모드: Counter with CBC-MAC(CCM)', 2007.12.

방송통신표준

기밀성과 메시지 인증을 제공하는 128 비트 블록 암호 운영 모드 (Modes of Operation of 128-bit Block Cipher for Confidentiality and Message Authentication)

발행인 : 미래창조과학부 장관
발행처 : 미래창조과학부 국립전파연구원
140-848, 서울 용산구 원효로41길 29
발행일 : 2013.12.
국립전파연구원 고시 제 2013-20호