

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

---

# AUTOSAR HaeModule Crypto Driver User Manual

## HSM and CryptoLib Module



일반(Anyuser)/종원 본 문서는 HyundaiAutoever의 정보자산이므로 무단으로 전제 및 복제할 수 없으며, 이를 위반할 시에는 명사  
사규 및 관련 법규에 의해 제재를 받을 수 있습니다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

VERSION HISTORY

Version	작성일	변경내용	작성자	비고
1.00	2023.11.17	사용자 매뉴얼 최초 작성	-	
1.01	2023.12.18	Crypto_76_HaeModule v1.0.1 업데이트	-	
1.02	2023.01.19	Crypto_76_HaeModule v1.0.2 업데이트	-	
1.02a	2024.03.05	매뉴얼 오류 수정 (RSAES-OAEP input struct, curve ID)	-	
1.02b	2024.03.06	HSM API 사용 주의 사항 추가	-	
1.02c	2024.03.07	매뉴얼 오류 수정 (ED448 input struct, curve ID)	-	
1.02d	2024.03.19	서명 값 +1 byte 크기 허용 작성	-	
1.03	2024.03.27	Pbkdf2 사용방법 추가	JeonJM	
1.03a	2024.04.16	매뉴얼 오류 수정 (block cipher key length 단위)	-	
1.03b	2024.04.24	KeyId 주의사항 추가, HSM 사용자 주의 사항 추가	JeonJM	

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## HSM 사용자 주의 사항

- HSM 사용자의 경우 낮은 성능의 MCU의 경우 API 알고리즘에 따라 WDT Timeout이 발생할 수 있으니 Task Timeout 시간을 충분히 늘리도록 한다. 이때 설정 시간의 경우 API 수행 시간에 따라 사용자가 설정해야 하며, 제어기 환경, MCU 환경, 사용자 설계 마다 결과가 달라질 수 있으니 이는 **사용자 환경에서 직접 확인해야 한다**. 관련 내용은 [“3.2 주의 사항”](#)에 자세히 명시되어 있으니 참고한다.
- Autosar 표준 사양에 따라 CRYPTO\_E\_BUSY가 return될 수 있다. 이 경우는 **에러 상황이 아니며, HSM이 이미 다른 TASK를 수행 중이라는 의미이다**. 따라서 잠시후 API를 다시 호출할 수 있도록 해주어야 한다. 이때 while 등을 활용하여 API를 계속해서 재시도할 경우 낮은 성능의 MCU는 위와 동일하게 WDT Timeout이 발생할 수 있으니, TASK 종료 후, 다시 TASK에 진입하여 API를 호출할 수 있도록 시퀀스를 설계해야 한다.
- 위와 같은 이유로 HSM API는 Async - SingleCall로 수행하기를 권장한다.
- 모든 HSM 관련 API는 Autosar 표준 사양을 만족하며 HSM 특화 기능 사양을 포함하고 있다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

차례

HSM 사용자 주의 사항.....3

1. 문서 개요 ..... 12

1.1. 문서의 목적 ..... 12

1.2. 문서 범위..... 12

1.3. 대상 ..... 12

1.4. 용어 정의..... 12

1.5. 참조 문서..... 12

2. HAEMODULE CRYPTO DRIVER 개요 ..... 13

2.1. HaeModule Crypto Driver 아키텍처 ..... 13

2.2. HaeModule Crypto Driver 구조..... 14

2.3. 제공 파일 및 툴 ..... 15

2.3.1. Static HaeModule Crypto Driver 소스 파일 및 헤더 파일.....15

2.3.2. AUTOSAR\_Crypto\_76\_HaeModule\_ECU\_Configuration\_PDF.arxml .....15

2.3.3. Ecud\_Crypto\_76\_HaeModule.arxml .....15

2.3.4. Bswmd\_Crypto\_76\_HaeModule.template .....15

2.3.5. Crypto Generator 툴 .....15

3. GENERAL 사양..... 16

3.1. 동작 조건..... 16

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

3.2.	주의 사항.....	16
3.3.	Interface 지원.....	17
3.4.	AUTOSAR Classic Release R19-11 적용 사항.....	18
4.	<b>CRYPTO GENERATOR 툴.....</b>	<b>20</b>
4.1	사용 방법 및 입력 파라미터.....	20
4.2	소스 및 헤더 파일 생성.....	20
4.3	BSWMD 파일 생성.....	20
5.	<b>CRYPTO DRIVER OBJECTS .....</b>	<b>21</b>
5.1.	지원 모듈.....	21
5.2.	모듈 별 Object와 Primitives .....	21
5.3.	모듈 별 Prefix.....	22
6.	<b>JOB PROCESSING .....</b>	<b>23</b>
6.1.	Crypto 알고리즘 별 Job Primitive 설정.....	23
6.1.1.	CRYPTO_HASH .....	23
6.1.2.	CRYPTO_MACGENERATE .....	24
6.1.2.1.	TDES CMAC.....	24
6.1.2.2.	AES CMAC.....	25
6.1.2.3.	AES GMAC .....	26
6.1.2.4.	HMAC.....	27
6.1.3.	CRYPTO_MACVERIFY .....	28

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

6.1.3.1.	TDES CMAC .....	28
6.1.3.2.	AES CMAC.....	29
6.1.3.3.	AES GMAC .....	30
6.1.3.4.	HMAC.....	31
6.1.4.	CRYPTO_ENCRYPT .....	32
6.1.4.1.	TDES.....	32
6.1.4.2.	AES .....	33
6.1.4.3.	CHACHA20.....	34
6.1.4.4.	RSAES-PKCS1_v1_5.....	35
6.1.4.5.	RSAES-OAEP.....	36
6.1.5.	CRYPTO_DECRYPT .....	37
6.1.5.1.	TDES.....	37
6.1.5.2.	AES .....	38
6.1.5.3.	CHACHA20.....	39
6.1.5.4.	RSAES-PKCS1_v1_5.....	40
6.1.5.5.	RSAES-OAEP.....	41
6.1.6.	CRYPTO_AEADENCRYPT .....	42
6.1.6.1.	AES GCM .....	42
6.1.6.2.	CHACHA20-POLY1305.....	44
6.1.7.	CRYPTO_AEADDECRYPT.....	45
6.1.7.1.	AES GCM .....	45

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

6.1.7.2.	CHACHA20-POLY1305 .....	47
6.1.8.	CRYPTO_SIGNATUREGENERATE .....	48
6.1.8.1.	RSASSA-PKCS1_v1_5.....	48
6.1.8.2.	RSASSA-PSS .....	49
6.1.8.3.	ECDSA .....	50
6.1.8.4.	EDDSA ED448.....	52
6.1.9.	CRYPTO_SIGNATUREVERIFY .....	53
6.1.9.1.	RSASSA-PKCS1_v1_5.....	53
6.1.9.2.	RSASSA-PSS .....	54
6.1.9.3.	ECDSA .....	55
6.1.9.4.	EDDSA ED448.....	57
6.1.10.	CRYPTO_RANDOMGENERATE .....	58
6.1.10.1.	DRBG .....	58
6.1.10.2.	PRNG .....	58
6.1.10.3.	TRNG .....	59
6.1.11.	CRYPTO_RANDOMSEED .....	60
6.1.11.1.	DRBG .....	60
6.1.12.	CRYPTO_KEYEXCHANGEALCPUBVAL.....	61
6.1.12.1.	Diffie-Hellman .....	61
6.1.12.2.	ECDH .....	62
6.1.13.	CRYPTO_KEYEXCHANGEALCSECRET.....	64

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

6.1.13.1.	Diffie-Hellman .....	64
6.1.13.2.	ECDH .....	65
6.1.14.	CRYPTO_KEYDERIVE.....	67
6.1.14.1.	PBKDF2 .....	67
6.1.15.	CRYPTO_KEYSETVALID .....	69
6.2.	신규 사용자 Crypto Primitive 추가 방법 .....	70
6.3.	Asynchronous Job을 위한 Crypto MainFunction 설정 .....	71
7.	KEY MANAGEMENT .....	72
7.1.	Crypto Key Element 설정 .....	72
7.1.1.	HSM Crypto Key Element 설정.....	72
7.1.2.	HSM Crypto Key Element 특성.....	74
7.2.	CryptoKeyElementValue 설정 방법.....	75
7.2.1.	CryptoKeyElementInitValue로 입력하는 방법.....	75
7.2.2.	Crypto_KeyElementSet API로 입력하는 방법.....	75
7.3.	신규 CryptoKey 추가 방법 .....	75
8.	CRYPTO CONFIGURATION .....	78
8.1.	CryptoGeneral.....	79
8.2.	CryptoDriverObjects .....	80
8.3.	CryptoPrimitives .....	81
8.4.	CryptoKeys .....	82
8.5.	CryptoKeyTypes .....	83
8.6.	CryptoKeyElements .....	84



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

9.

CRYPTO DRIVER 포팅 가이드.....

86

9.1.

mobilgene Classic 플랫폼 .....

86

9.2.

AUTOSAR 플랫폼 .....

86

9.3.

메모리 섹션 .....

87

9.3.1.

Link Script 섹션 .....

87

9.3.2.

플랫폼 메모리 맵 파일.....

88

부록.

기술지원 가이드 .....

90

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

그림

그림 1. AUTOSAR Layered View with HaeModule Crypto Driver .....

13

그림 2. HaeModule Crypto BSWMD 생성 .....

14

그림 3. HSM Generated Configuration Files .....

14

그림 4. HaeModule Crypto Driver 소스 구조 .....

14

그림 5. HaeModule Crypto Driver 지원 모듈 .....

21

그림 6. 모듈 별 Object와 Primitives .....

21

그림 7. 신규 CryptoPrimitive 추가 .....

70

그림 8. HaeModule CryptoKeyElement Container .....

73

그림 9. HSM Crypto Key Element 설정 .....

73

그림 10. AUTOSAR Crypto Container 구조 .....

78

그림 11. AUTOSAR CryptoGeneral Container .....

79

그림 12. AUTOSAR CryptoDriverObject Container .....

80

그림 13. AUTOSAR CryptoPrimitive Container .....

81

그림 14. AUTOSAR CryptoKey Container .....

82

그림 15. AUTOSAR CryptoKeyType Container .....

83

그림 16. AUTOSAR CryptoKeyElement Container .....

85

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

표

표 1. Crypto Driver Interface 지원

18

표 2. 모듈 별 Object와 Primitives

21

표 3. 모듈 별 Prefix

22

표 4. Object 별 Crypto MainFunction

71

표 5. HSM Key 테이블

72

표 6. HaeModule CryptoGeneral Container 내용

79

표 7. HaeModule CryptoDriverObject Container 내용

80

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 1. 문서 개요

### 1.1. 문서의 목적

본 문서는 자사의 HSM과 CryptoLib 모듈을 위한 AUTOSAR Crypto Driver 사용을 위한 가이드 제공을 목적으로 한다.

### 1.2. 문서 범위

본 문서는 자사의 HSM과 CryptoLib 모듈을 위한 AUTOSAR Crypto Driver 사용을 위한 Driver의 개요, 구조, 기능, API를 설명한다.

### 1.3. 대상


본 문서는 자사의 HSM 모듈 또는 CryptoLib이 탑재된 보안기능을 수행하는 ECU 개발을 담당하는 개발자를 위해 작성되었다. 본 문서의 사용자는 Autosar 표준 사양에 대한 매뉴얼을 사전에 숙지하고 있어야 한다.

### 1.4. 용어 정의

앞으로 자사의 HSM과 CryptoLib 모듈 모두를 말할 경우 HaeModule 이라는 용어로 사용하고, 각각의 모듈을 말할 경우 HSM 모듈 또는 CryptoLib 모듈 이라는 용어를 사용하도록 하겠다.

### 1.5. 참조 문서

번호	문서	버전
1	AUTOSAR_SWS_CryptoServiceManager.pdf	4.4.0, R19-11
2	AUTOSAR_SWS_CryptoInterface.pdf	4.4.0
3	AUTOSAR_SWS_CryptoDriver.pdf	4.4.0
4	[HSM_2.0spec]HSM_Framework_UserManual_Secure_Application_Guide_KOR_v2.0.pdf	2.00
5	[HSM_2.0spec]HSM_Framework_UserManual_Crypto_Service_KOR_v2.0.pdf	1.02
6	User's Manual of Hyundai AutoEver Cryptography Library.pdf	1.7.2

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 2. HaeModule Crypto Driver 개요

### 2.1. HaeModule Crypto Driver 아키텍처

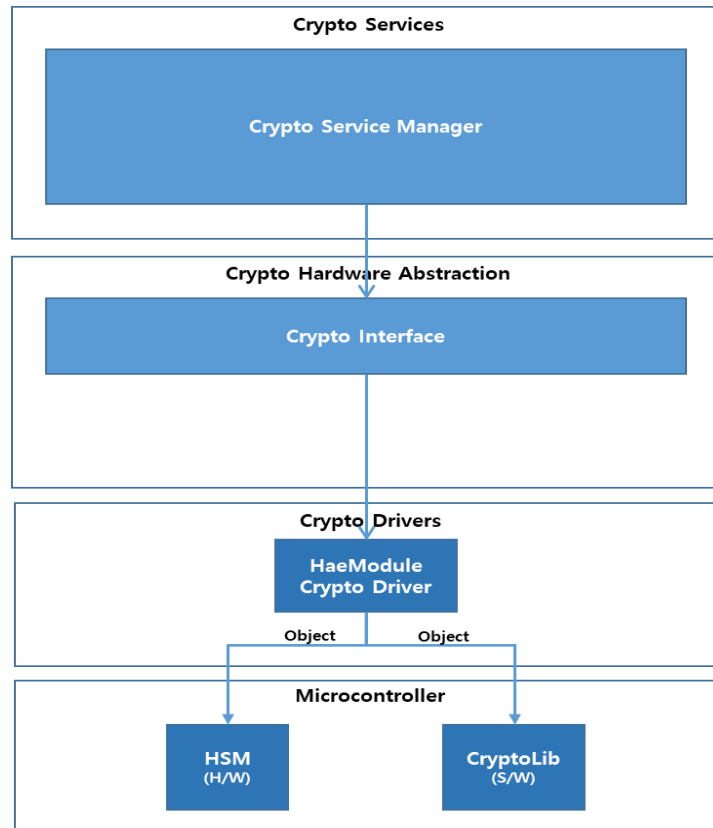


그림 1. AUTOSAR Layered View with HaeModule Crypto Driver

AUTOSAR 에서 Crypto Driver는 동기 및 비 동기의 암호화 서비스와 이를 위한 Key 관리 서비스를 제공한다. 위 그림은 HaeModule Crypto Driver 관점에서의 전체적인 Crypto Stack 아키텍처를 보여주고 있다. HaeModule Crypto Driver는 Crypto Drivers Layer에서 HSM과 CryptoLib 모듈을 위한 서비스를 제공한다.

HSM은 세부적으로는 HSM Driver와 Microcontroller 내의 별도 Core에서 동작하는 HSM으로 구성되어 있다. CryptoLib은 소프트웨어로 구현된 암호화 알고리즘 서비스를 제공하는 라이브러리 형태로 제공된다. 자세한 구조 및 동작은 각 모듈에서 제공하는 사용자 매뉴얼을 참고 바란다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 2.2. HaeModule Crypto Driver 구조

HaeModule Crypto BSWMD(Basic Software Module Description)의 생성을 과정을 보여 주고 있다.

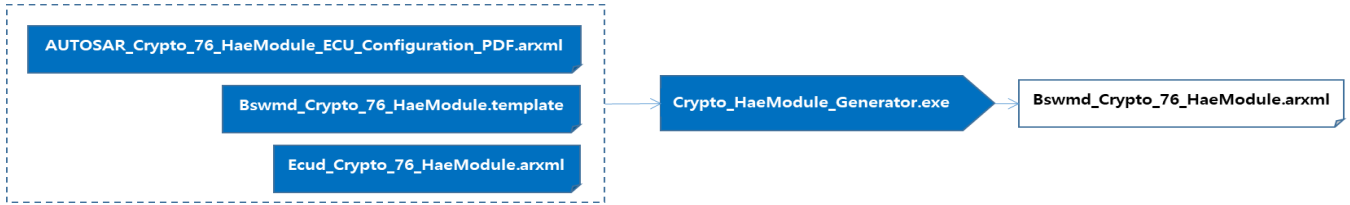


그림 2. HaeModule Crypto BSWMD 생성

Generator를 사용하여 HaeModule Crypto Driver의 Configuration File 생성 과정을 보여주고 있다.

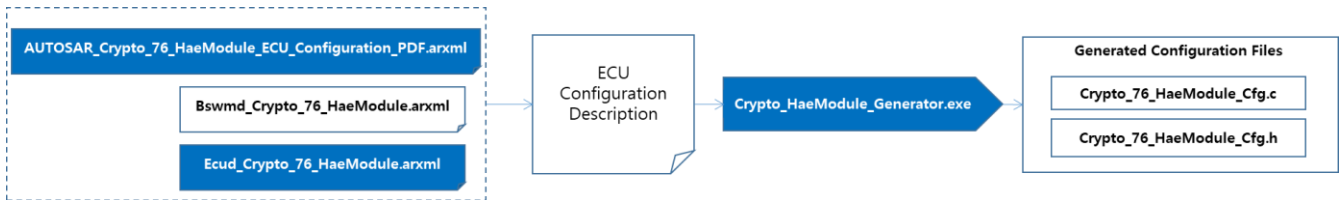


그림 3. HSM Generated Configuration Files

아래는 전체적인 HaeModule Crypto Driver의 소스 구조를 보여주고 있다. **HaeModule Crypto Driver는 HSM Driver 또는 CryptoLib 위에서 동작함으로 반드시 HSM Driver Library 또는 CryptoLib Library가 필요하다.**

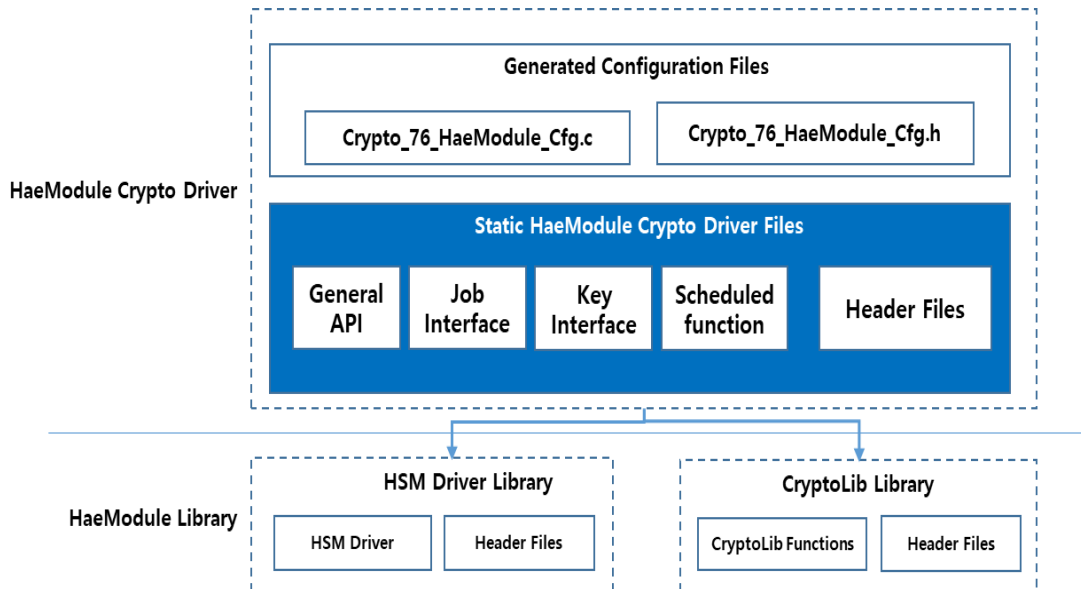


그림 4. HaeModule Crypto Driver 소스 구조

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 2.3. 제공 파일 및 툴

다음과 같은 파일들과 툴을 제공한다.

‘Crypto\_76\_HaeModule\_R44’ 패키지 명으로 제공되며 제공 아이템 별 디렉토리 위치는 다음과 같다.

NO	아이템	위치
1	Static HaeModule Crypto Driver 소스 파일	delivery/src
2	Static HaeModule Crypto Driver 헤더 파일	delivery/inc
3	AUTOSAR_Crypto_76_HaeModule_ECU_Configuration_PDF.arxml	generator
4	Ecud_Crypto_76_HaeModule.arxml	generator
5	Bswmd_Crypto_76_HaeModule.template	generator
6	Crypto Generator 툴 - Crypto_HaeModule_Generator.exe	generator
7	Crypto Generator 툴 - Crypto_HaeModule_Generator.bat	generator
8	본 매뉴얼	doc

### 2.3.1. Static HaeModule Crypto Driver 소스 파일 및 헤더 파일

AUTOSAR 스펙에 맞는 Crypto Driver 기능을 제공한다. Crypto Generator 툴에 의해 생성되는 소스 파일에 의해 Configuration 된다. HSM Driver나 CryptoLib API를 호출하여 서비스를 수행한다.

### 2.3.2. AUTOSAR\_Crypto\_76\_HaeModule\_ECU\_Configuration\_PDF.arxml

Ecud\_Crypto\_76\_HaeModule.arxml 검증을 위한 스키마를 제공한다.

### 2.3.3. Ecud\_Crypto\_76\_HaeModule.arxml

AUTOSAR XML 형식을 따른다. HaeModule Crypto Driver의 ECU Description을 제공한다. Configuration 파일 및 BSWMD 파일 생성을 위해 Crypto Generator 툴의 입력 파라미터에 설정되어야 한다.

### 2.3.4. Bswmd\_Crypto\_76\_HaeModule.template

Bswmd\_Crypto\_76\_HaeModule.arxml 파일을 위한 템플릿 파일이다.

### 2.3.5. Crypto Generator 툴

자세한 내용은 “4. Crypto Generator 툴”을 참조.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 3. General 사양

기본적으로 AUTOSAR Classic Release 4.4.0 사양을 따른다. 단지 일부 사양에 대해서는 상위 버전을 적용한다.

#### 3.1. 동작 조건

1. 자사에서 제공하는 HSM 보안 모듈과 CryptoLib 보안 모듈 지원을 위해 본 AUTOSAR Crypto Driver가 개발되었다.
2. 따라서 Crypto Driver 기능은 자사에서 제공하는 HSM 보안 모듈 또는 CryptoLib 보안 모듈에 의해 제한될 수 있다.
  - A. CryptoLib 버전은 v1.7.2부터 Crypto Driver를 제공한다.
3. Crypto Driver는 AUTOSAR 표준을 따르지만 자사의 HSM 보안 모듈이나 CryptoLib 보안 모듈의 기능이 나 동작 방법에 따라 지원 기능이나 동작에 약간의 차이가 있을 수 있다. 따라서 사용 전에 본 매뉴얼에 관련 기능 설명을 반드시 확인하기 바란다.
4. Crypto Driver가 동작하기 전에 자사 HSM 모듈 또는 CryptoLib 모듈이 정상적으로 동작하고 있어야 한다.
5. HSM 모듈의 초기화 함수(HSM\_DriverInitialize)가 실행되어 반드시 초기화가 완료된 후 Crypto Driver 초기화 함수(Crypto\_76\_HaeModule\_Init)가 그 후에 실행되어 초기화 되어야 한다.
6. CryptoLib의 초기화 함수가 실행되어 반드시 초기화가 완료된 후 Crypto Driver 초기화 함수가 그 후에 실행되어 초기화 되어야 한다.
7. Crypto Driver는 HSM 보안 모듈과 CryptoLib 보안 모듈의 설정에 관여하지 않는다. 단지 HSM Driver API나 CryptoLib API를 이용하여 AUTOSAR 표준 Crypto Driver 기능을 제공한다. 따라서 사용자는 Crypto Driver가 정상 동작하기 위해서 HSM 보안 모듈과 CryptoLib 보안 모듈을 각 모듈의 사용자 매뉴얼을 참조해서 Crypto Driver와 별도로 설치해야 한다.
8. Crypto Driver에서 사용하는 사용자 입력에 대한 정의는 Crypto\_76\_HaeModule\_UserTypes.h를 참고한다.

#### 3.2. 주의 사항

1. SYNC mode HSM API 를 사용시 함수 리턴 응답 시간이 task 의 주기보다 더 긴 경우 HOST core 사용자 환경에서 Watchdog Timeout 이 발생할 수 있다.
2. HSM API 또는 MCU 사양에 따라서 100ms 이상의 작업 시간을 가질 수 있다.
3. 사용자 환경마다 리턴 시간 각기 다르기 때문에 직접 측정하여 task 디자인을 고려해야 함. (오래 걸리는 HSM API 호출을 이상 없이 동작할 수 있도록 task 를 디자인하는 주체는 사용자임)



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 3.3. Interface 지원

다음 표에 Crypto Driver의 Interface 지원 여부를 표시하였다.

No.	Service name	Description	Support
1	Crypto_Init	Initializes the Crypto Driver.	○
2	Crypto_GetVersionInfo	Returns the version information of this module.	○
3	Crypto_ProcessJob	Performs the crypto primitive, that is configured in the job parameter	○
4	Crypto_CancelJob	This interface removes the provided job from the queue and cancels the processing of the job if possible	○
5	Crypto_KeyElementSet	Sets the given key element bytes to the key identified by cryptoKeyld.	○
6	Crypto_KeySetValid	Sets the key state of the key identified by cryptoKeyld to valid.	○
7	Crypto_KeyElementGet	This interface shall be used to get a key element of the key identified by the cryptoKeyld and store the key element in the memory location pointed by the result pointer. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation).	○
8	Crypto_KeyElementCopy	Copies a key element to another key element in the same crypto driver. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation)	○
9	Crypto_KeyElementCopyPartial	Copies a key element to another key element in the same crypto driver. The keyElementSourceOffset and keyElementCopyLength allows to copy just a part of the source key element into the destination. The offset of the target key is also specified with this function. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation).	○
10	Crypto_KeyCopy	Copies a key with all its elements to another key in the same crypto driver. Note: If the actual key element is directly mapped to flash memory, there could be a bigger delay when calling this function (synchronous operation)	○

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

11	Crypto_KeyElementIdsGet	Used to retrieve information which key elements are available in a given key.	○
12	Crypto_RandomSeed	This function generates the internal seed state using the provided entropy source. Furthermore, this function can be used to update the seed state with new entropy	○
13	Crypto_KeyGenerate	Generates new key material store it in the key identified by cryptoKeyld.	×
14	Crypto_KeyDerive	Derives a new key by using the key elements in the given key identified by the cryptoKeyld. The given key contains the key elements for the password, salt. The derived key is stored in the key element with the id 1 of the key identified by targetCryptoKeyld. The number of iterations is given in the key element CRYPTO_KE_KEYDERIVATION_ITERATIONS.	○
15	Crypto_KeyExchangeCalcPubVal	Calculates the public value for the key exchange and stores the public key in the memory location pointed by the public value pointer.	○
16	Crypto_KeyExchangeCalcSecret	Calculates the shared secret key for the key exchange with the key material of the key identified by the cryptoKeyld and the partner public key. The shared secret key is stored as a key element in the same key.	○
17	Crypto_CertificateParse	Parses the certificate data stored in the key element CRYPTO_KE_CERT_DATA and fills the key elements CRYPTO_KE_CERT_SIGNEDDATA, CRYPTO_KE_CERT_PARSEDPUBLICKEY and CRYPTO_KE_CERT_SIGNATURE.	×
18	Crypto_CertificateVerify	Verifies the certificate stored in the key referenced by cryptoValidateKeyld with the certificate stored in the key referenced by cryptoKeyld.	×
19	Crypto_MainFunction	If asynchronous job processing is configured and there are job queues, the function is called cyclically to process queued jobs.	○

표 1. Crypto Driver Interface 지원

### 3.4. AUTOSAR Classic Release R19-11 적용 사항

1. Key Management Interface 중에 아래와 같은 Certificate Interface가 삭제됨에 따라 HaeModule Crypto Driver에서도 더 이상 지원하지 않는다.

일반(Anyuser)/종원 본 문서는 HyundaiAutoever의 정보자산이므로 무단으로 전제 및 복제할 수 없으며, 이를 위반할 시에는 명사  
사규 및 관련 법규에 의해 제재를 받을 수 있습니다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

■ Crypto\_CertificateParse

■ Crypto\_CertificateVerify

2. Crypto\_JobType에서 “CryptoKeyId”와 “targetCryptoKeyId” 변수가 명확하게 정의됨에 따라 이를 적용하였다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 4. Crypto Generator 툴

Generator 툴은 Windows용 콘솔 어플리케이션이다. AUTOSAR ARXML 파일을 분석하여 Crypto Configuration 소스 파일과 BSWMD 파일을 생성한다.

### 4.1 사용 방법 및 입력 파라미터

입력 없이 Generator 툴을 실행하면 사용 방법 및 입력 파라미터 도움말을 얻을 수 있다.

### 4.2 소스 및 헤더 파일 생성

“Crypto\_76\_HaeModule\_Cfg.c”와 “Crypto\_76\_HaeModule\_Cfg.h” 파일을 생성한다. **CryIf ECUD 파일을 입력하면 CryIf와 연결된 Object와 Key만을 코드로 생성하여 사이즈를 줄일 수 있다.** 입력하지 않으면 Crypto ECUD에 설정되어 있는 모든 것을 코드로 생성한다. 헤더 파일과 소스 파일 생성 디렉토리 위치를 같은 곳에 위치하고 싶다면 “--ODirH”와 “--OdirC”를 동일하게 설정한다. 다음은 사용 예이다.

예)) `Crypto_HaeModule_Generator.exe --CODE --IEcudCryIf Ecud_CryIf.arxml --IEcudCrypto Ecud_Crypto_76_HaeModule.arxml -IBswmdCrypto Bswmd_Crypto_76_HaeModule.arxml --ODirH C:\W Generated\WBSw_Output\Winc --OdirC C:\WGenerated\WBSw_Output\Wsrc`

### 4.3 BSWMD 파일 생성

“Bswmd\_Crypto\_76\_HaeModule.arxml” 파일을 생성한다. 다음은 사용 예이다.

예)) `Crypto_HaeModule_Generator.exe --BSWMD --IEcudCrypto Ecud_Crypto_76_HaeModule.arxml --IBswmdCrypto Bswmd_Crypto_76_HaeModule.arxml --ODir C:\W Generated\WBSw_Output\Wbswmd`

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 5. Crypto Driver Objects

### 5.1. 지원 모듈

HaeModule Crypto Driver는 자사의 HSM 모듈과 CryptoLib 모듈을 지원하기 위해 두 개의 Object를 가진다.

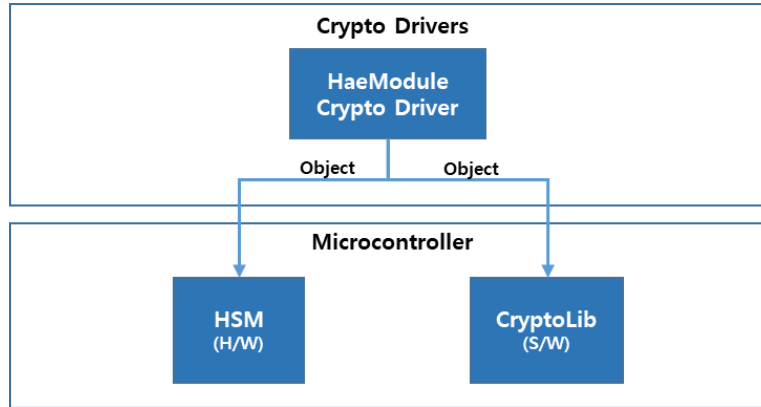


그림 5. HaeModule Crypto Driver 지원 모듈

### 5.2. 모듈 별 Object와 Primitives

다음은 각 지원 모듈의 Object와 Primitives는 다음과 같다.

No.	Module	Object	Primitives
1	HSM 모듈	HaeHsm	HaeHsmPrimitives
2	CryptoLib 모듈	HaeCryptoLib	HaeCryptoLibPrimitives

표 2. 모듈 별 Object와 Primitives

해당 이름의 object 또는 Primitives에서 각각 설정한다.

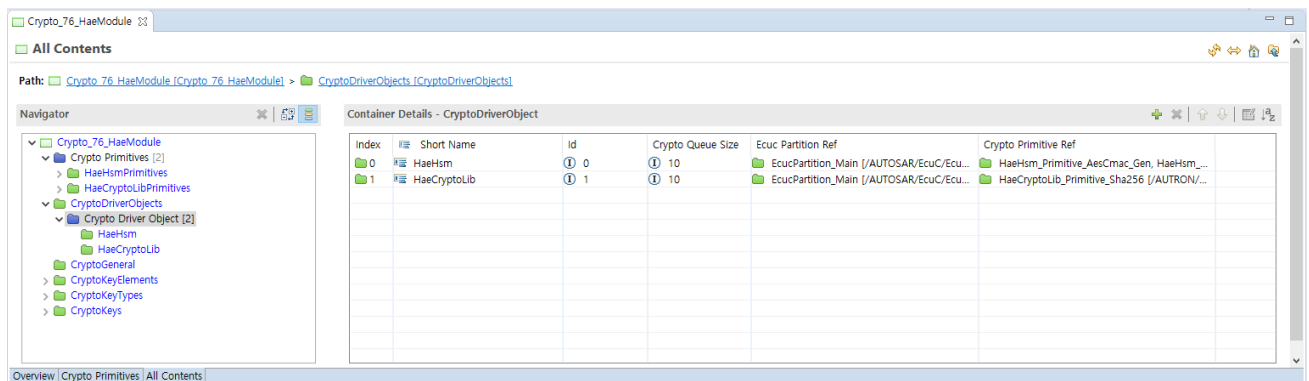


그림 6. 모듈 별 Object와 Primitives

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 5.3. 모듈 별 Prefix

모듈 별 설정 또는 Primitive를 구분하기 위해 다음과 같은 Prefix를 사용하기를 권장한다. 제공되는 소스 및 ARXML은 이 규칙을 따른다.

No.	Module	Prefix
1	HSM 모듈	HaeHsm
2	CryptoLib 모듈	HaeCryptoLib

표 3. 모듈 별 Prefix

HSM과 CryptoLib 모듈의 공통된 사항인 경우 “HaeModule”이라는 Prefix를 사용한다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6. Job Processing

### 6.1. Crypto 알고리즘 별 Job Primitive 설정

본 장에서는 Crypto 알고리즘을 사용하기 위한 Job 설정에 대해 설명한다. Job 설정을 통해 HAE HSM에서 제공하는 Crypto 알고리즘을 사용할 수 있다.

‘Crypto\_PrimitiveInfoType’ 테이블은 기본적으로 제공되는 ‘Ecud\_Crypto\_76\_HaeModule.arxml’ 파일에 설정된 값을 나타낸다. 사용자는 자신의 환경에 맞게 CSM(Crypto Service Manager)과 맞추어 값을 변경하도록 한다.

HaeHsmPrimitives와 HaeCryptoLibPrimitives가 제공하는 알고리즘이 상이할 때는 “< >” 로 구분하였으니 확인하도록 한다.

#### 6.1.1. CRYPTO\_HASH

Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_HASH	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_HASH	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_HASH	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_HASH	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_HASH	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_HASH	CRYPTO_ALGOFAM_SHA2_512_224	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_HASH	CRYPTO_ALGOFAM_SHA2_512_256	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_HASH	CRYPTO_ALGOFAM_SHAKE256	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	입력 데이터
uint32	inputLength	inputPtr 바이트 사이즈
uint8*	outputPtr	Hash 결과 데이터
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.2. CRYPTO\_MACGENERATE

### 6.1.2.1. TDES CMAC

Crypto\_PrimitiveInfoType < [HaeCryptoLibPrimitives](#) >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CMACE

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	CMAC 연산할 입력 데이터
uint32	inputLength	CMAC 연산할 입력 데이터의 바이트 사이즈
uint8*	outputPtr	CMAC 결과 데이터
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	TDES CMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	TDES CMAC 키 값 (8-byte)



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.2.2. AES CMAC

#### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CMACE

#### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	CMAC 연산할 입력 데이터
uint32	inputLength	CMAC 연산할 입력 데이터의 바이트 사이즈
uint8*	outputPtr	CMAC 결과 데이터
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

#### Crypto\_AlgorithmInfoType

Type	Parameter	Description
uint32	keyLength	The key length in bytes to be used with that algorithm 적용 AES Key Byte Length <ul style="list-style-type: none"> <li>AES-128 : 16</li> <li>AES-192 : 24 (<b>HaeHsmPrimitives 불가</b>)</li> <li>AES-256 : 32 (<b>HaeHsmPrimitives 불가</b>)</li> </ul>

- 설정 값에 따라 적용 알고리즘이 결정된다.

#### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyld	AES CMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	AES CMAC 키 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.2.3. AES GMAC

Crypto\_PrimitiveInfoType < **HaeCryptoLibPrimitives** >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_GMAC

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	GMAC 연산할 입력 메시지 데이터 (AAD는 사용하지 않음)
uint32	inputLength	GMAC 연산할 입력 메시지 데이터의 바이트 사이즈
uint8*	outputPtr	GMAC 결과 데이터
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

- CRYPTO\_OPERATIONMODE\_SINGLECALL 사용 시, g\_cryptoLib\_gmacLastUpdateCall = 1 설정이 필요
- 이 외 모드에서 데이터 블록 입력 시, g\_cryptoLib\_gmacLastUpdateCall = 0로 설정 후 inputPtr은 16 바이트의 배수만 입력 가능하다.  
마지막 데이터 블록 입력 시, g\_cryptoLib\_gmacLastUpdateCall = 1로 설정 후에는 16 바이트 외의 길이도 가능하다.  
추가 관련 사항은 CryptoLib 매뉴얼을 참고.

Crypto\_AlgorithmInfoType

Type	Parameter	Description
uint32	keyLength	The key length in bytes to be used with that algorithm 적용 AES Key Byte Length <ul style="list-style-type: none"> <li>AES-128 : 16</li> <li>AES-192 : 24</li> <li>AES-256 : 32</li> </ul>

- 설정 값에 따라 적용 알고리즘이 결정된다.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES GMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	AES GMAC 키 값
CRYPTO_KE_MAC_IV	1005	AES GMAC IV 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.2.4. HMAC

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACGENERATE	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	HMAC 연산할 입력 데이터
uint32	inputLength	HMAC 연산할 입력 데이터의 바이트 사이즈
uint8*	outputPtr	HMAC 결과 데이터
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	HMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	HMAC 키 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.3. CRYPTO\_MACVERIFY

#### 6.1.3.1. TDES CMAC

Crypto\_PrimitiveInfoType < [HaeCryptoLibPrimitives](#) >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CMACE

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	Holds a pointer to the data for which the MAC shall be verified.
uint32	inputLength	Contains the number of data bytes for which the MAC shall be verified.
const uint8*	secondaryInputPtr	Holds a pointer to the MAC to be verified.
uint32	secondaryInputLength	Contains the MAC length in BITS to be verified.
Crypto_VerifyResultType*	verifyPtr	Holds a pointer to the memory location, which will hold the result of the MAC verification.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	TDES CMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	TDES CMAC 키 값 (8-byte)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.3.2. AES CMAC

#### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CMACE

#### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	Holds a pointer to the data for which the MAC shall be verified.
uint32	inputLength	Contains the number of data bytes for which the MAC shall be verified.
const uint8*	secondaryInputPtr	Holds a pointer to the MAC to be verified.
uint32	secondaryInputLength	Contains the MAC length in BITS to be verified.
Crypto_VerifyResultType*	verifyPtr	Holds a pointer to the memory location, which will hold the result of the MAC verification.

#### Crypto\_AlgorithmInfoType

Type	Parameter	Description
uint32	keyLength	The key length in bytes to be used with that algorithm 적용 AES Key Byte Length <ul style="list-style-type: none"> <li>AES-128 : 16</li> <li>AES-192 : 24 (HaeHsmPrimitives 불가)</li> <li>AES-256 : 32 (HaeHsmPrimitives 불가)</li> </ul>

- 설정 값에 따라 적용 알고리즘이 결정된다.

#### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES CMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	AES CMAC 키 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.3.3. AES GMAC

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_GMAC

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	Holds a pointer to the data for which the MAC shall be verified. (not used AAD)
uint32	inputLength	Contains the number of data bytes for which the MAC shall be verified.
const uint8*	secondaryInputPtr	Holds a pointer to the MAC to be verified.
uint32	secondaryInputLength	Contains the MAC length in BITS to be verified.
Crypto_VerifyResultType*	verifyPtr	Holds a pointer to the memory location, which will hold the result of the MAC verification.

- CRYPTO\_OPERATIONMODE\_SINGLECALL 사용 시, g\_cryptoLib\_gmacLastUpdateCall = 1 설정이 필요
- 이 외 모드에서 데이터 블록 입력 시, g\_cryptoLib\_gmacLastUpdateCall = 0로 설정 후 inputPtr은 16 바이트의 배수만 입력 가능하다.  
마지막 데이터 블록 입력 시, g\_cryptoLib\_gmacLastUpdateCall = 1로 설정 후에는 16 바이트 외의 길이도 가능하다.  
추가 관련 사항은 CryptoLib 매뉴얼을 참고.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES GMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	AES GMAC 키 값
CRYPTO_KE_MAC_IV	1005	AES GMAC 키 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.3.4. HMAC

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC
CRYPTO_MACVERIFY	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_HMAC

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	Holds a pointer to the data for which the MAC shall be verified.
uint32	inputLength	Contains the number of data bytes for which the MAC shall be verified.
const uint8*	secondaryInputPtr	Holds a pointer to the MAC to be verified.
uint32	secondaryInputLength	Contains the MAC length in BITS to be verified.
Crypto_VerifyResultType*	verifyPtr	Holds a pointer to the memory location, which will hold the result of the MAC verification.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	HMAC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_MAC_KEY	1	HMAC 키 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.4. CRYPTO\_ENCRYPT

### 6.1.4.1. TDES

Crypto\_PrimitiveInfoType < [HaeCryptoLibPrimitives](#) >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_ECB
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CBC
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CTR

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	평문 데이터 바이트 사이즈
uint8*	outputPtr	암호화된 데이터 저장 포인터
uint32*	outputLengthPtr	암호화된 데이터 길이 포인터

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	TDES 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	TDES 키 값 (8-byte)
CRYPTO_KE_CIPHER_IV	5	TDES IV 값 (8-byte, ECB의 경우, 필요하지 않음)



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.4.2. AES

##### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_ECB
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CBC
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CTR
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_OFB

##### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	평문 데이터 바이트 사이즈
uint8*	outputPtr	암호화된 데이터 저장 포인터
uint32*	outputLengthPtr	암호화된 데이터 길이 포인터

##### Crypto\_AlgorithmInfoType

Type	Parameter	Description
uint32	keyLength	<p>The key length in bytes to be used with that algorithm</p> <p>적용 AES Key Byte Length</p> <ul style="list-style-type: none"> <li>AES-128 : 16</li> <li>AES-192 : 24 (<b>HaeHsmPrimitives 불가</b>)</li> <li>AES-256 : 32</li> </ul>

- 설정 값에 따라 적용 알고리즘이 결정된다.

##### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	AES GMAC 키 값
CRYPTO_KE_CIPHER_IV	5	AES GMAC IV 값 (ECB의 경우, 필요하지 않음)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.4.3. CHACHA20

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_CHACHA	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_20ROUNDS

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	평문 데이터 바이트 사이즈
uint8*	outputPtr	암호화된 데이터 저장 포인터
uint32*	outputLengthPtr	암호화된 데이터 길이 포인터

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	CHACHA20 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	CHACHA20 키 값 (32-byte)
CRYPTO_KE_CIPHER_IV	5	CHACHA20 InitialCounter    nonce (16-byte)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.4.4. RSAES-PKCS1\_v1\_5

##### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5

##### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	평문 데이터 데이터 바이트 사이즈 (최대 길이 256바이트)
uint8*	outputPtr	암호화된 데이터 저장 포인터
uint32*	outputLengthPtr	암호화된 데이터 길이 포인터

##### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	RSA 공개키 값 (260-byte = N    E (4-byte))

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.4.5. RSAES-OAEP

- CSM R4.4 CsmJobKeyDeriveAlgorithmSecondaryFamily 미지원 (R22-11 지원)

- CRYPTO\_ALGOFAM\_CUSTOM 설정 후, AlgorithmSecondaryFamilyCustom에 값 입력

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSAES_OAEP

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_RSAES_OAEP

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	HAEMODULE_RSAES_OAEP_INPUT_t의 포인터
uint32	inputLength	inputPtr 바이트 사이즈
uint8*	outputPtr	암호화된 데이터 저장 포인터
uint32*	outputLengthPtr	암호화된 데이터 길이 포인터

HAEMODULE\_RSAES\_OAEP\_INPUT\_t

Name	HSM_RSAES_OAEP_INPUT_t		
Type	Structure		
Element	uint8*	textPtr	평문 데이터
	uint32	textLength	평문 데이터 길이
	uint8*	labelPtr	Label 데이터
	uint32	labelLength	Label 데이터의 길이
Description	알고리즘의 입력 데이터 정보		

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	RSA 공개키 값 (260-byte = N    E (4-byte))
CRYPTO_KE_CIPHER_SEED	1016	RSAES-OAEP seed 값 (hash output length)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.5. CRYPTO\_DECRYPT

### 6.1.5.1. TDES

Crypto\_PrimitiveInfoType < [HaeCryptoLibPrimitives](#) >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_ECB
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CBC
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_TDES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CTR

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	암호문 데이터 포인터
uint32	inputLength	암호문 데이터 바이트 사이즈
uint8*	outputPtr	복호화된 데이터 저장 포인터
uint32*	outputLengthPtr	복호화된 데이터 길이 포인터

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	TDES 키 값 (8-byte)
CRYPTO_KE_CIPHER_IV	5	TDES IV 값 (8-byte, ECB의 경우, 필요하지 않음)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.5.2. AES

#### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_ECB
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CBC
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CTR
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_OFB

#### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	암호문 데이터 포인터
uint32	inputLength	암호문 데이터 바이트 사이즈
uint8*	outputPtr	복호화된 데이터 저장 포인터
uint32*	outputLengthPtr	복호화된 데이터 길이 포인터

#### Crypto\_AlgorithmInfoType

Type	Parameter	Description
uint32	keyLength	The key length in bytes to be used with that algorithm 적용 AES Key Byte Length <ul style="list-style-type: none"> <li>AES-128 : 16</li> <li>AES-192 : 24 (HaeHsmPrimitives 불가)</li> <li>AES-256 : 32</li> </ul>

- 설정 값에 따라 적용 알고리즘이 결정된다.

#### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	AES 키 값
CRYPTO_KE_CIPHER_IV	5	AES IV 값 (ECB의 경우, 필요하지 않음)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.5.3. CHACHA20

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_CHACHA	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_20ROUNDS

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	암호문 데이터 포인터
uint32	inputLength	암호문 데이터 바이트 사이즈
uint8*	outputPtr	복호화된 데이터 저장 포인터
uint32*	outputLengthPtr	복호화된 데이터 길이 포인터

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	CHACHA20 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	CHACHA20 키 값 (32-byte)
CRYPTO_KE_CIPHER_IV	5	CHACHA20 InitialCounter    nonce (16-byte)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.5.4. RSAES-PKCS1\_v1\_5

##### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_ENCRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5

##### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	암호 메시지 포인터
uint32	inputLength	암호 메시지 데이터 바이트 사이즈 (최대 길이 256바이트)
uint8*	outputPtr	Contains the pointer to the memory location where the decrypted data shall be stored.
uint32*	outputLengthPtr	Holds a pointer to the memory location in which the output length information is stored in bytes. On calling this function, this parameter shall contain the size of the buffer provided by outputPtr. When the request has finished, the actual length of the returned value shall be stored.

##### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyld	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	RSA 개인키 값 (512-byte = N    D)



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.5.5. RSAES-OAEP

- CSM R4.4 CsmDecryptAlgorithmSecondaryFamily 미지원 (R22-11 지원)

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSAES_OAEP

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_RSAES_OAEP
CRYPTO_DECRYPT	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_RSAES_OAEP

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	HAEMODULE_RSAES_OAEP_INPUT_t의 포인터
uint32	inputLength	inputPtr 바이트 사이즈
uint8*	outputPtr	복호화된 데이터 저장 포인터
uint32*	outputLengthPtr	복호화된 데이터 길이 포인터

HAEMODULE\_RSAES\_OAEP\_INPUT\_t

Name	HSM_RSAES_OAEP_INPUT_t		
Type	Structure		
Element	uint8*	textPtr	256Byte의 암호문 데이터
	uint32	textLength	암호문 데이터 길이
	uint8*	labelPtr	Label 데이터
	uint32	labelLength	Label 데이터의 길이
Description	알고리즘의 입력 데이터 정보		

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	RSA 개인키 값 (512-byte = N    D)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.6. CRYPTO\_AEADENCRYPT

### 6.1.6.1. AES GCM

Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_AEADENCRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_GCM

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	평문 데이터 바이트 사이즈
const uint8*	secondaryInputPtr	Aad 데이터 포인터
uint32	secondaryInputLength	Aad 데이터의 바이트 사이즈
uint8*	outputPtr	암호화된 데이터 포인터
uint32*	outputLengthPtr	암호화된 데이터 길이 포인터
uint8*	secondaryOutputPtr	암호문의 검증 Tag 데이터 포인터
uint32*	secondaryOutputLengthPtr	암호문의 검증 Tag 데이터 바이트 사이즈

- **HaeCryptoLibPrimitives 주의 가이드**
  - CRYPTO\_OPERATIONMODE\_SINGLECALL 사용 시, g\_cryptoLib\_gcmLastUpdateCall = 1 설정이 필요
  - 이 외 모드에서 데이터 블록 입력 시, g\_cryptoLib\_gcmLastUpdateCall = 0로 설정 후 inputPtr은 16 바이트의 배수만 입력 가능하다.  
마지막 데이터 블록 입력 시, g\_cryptoLib\_gcmLastUpdateCall = 1로 설정 후에는 16 바이트 외의 길이도 가능하다.  
추가 관련 사항은 CryptoLib 매뉴얼을 참고.
- **HaeHsmPrimitives 주의 가이드**
  - HSM 보안 모듈의 기능 제한으로 연산할 입력 데이터와 Aad의 길이는 최대 128 바이트만 지원한다.
  - CRYPTO\_OPERATIONMODE\_UPDATE 모드에서는 HSM Driver API 정상 동작을 위해 입력 데이터, Aad, Tag, 출력 데이터 저장 포인터 등 모든 파라미터들은 설정되어 있어야 한다.
  - 따라서 CRYPTO\_OPERATIONMODE\_SINGLECALL 동작 모드를 사용한다.

Crypto\_AlgorithmInfoType

Type	Parameter	Description
uint32	keyLength	The key length in bytes to be used with that algorithm 적용 AES Key Byte Length <ul style="list-style-type: none"> <li>• AES-128 : 16</li> <li>• AES-192 : 24 (<b>HaeHsmPrimitives 불가</b>)</li> <li>• AES-256 : 32</li> </ul>

일반(Anyuser)/종원 본 문서는 HyundaiAutoever의 정보자산이므로 무단으로 전제 및 복제할 수 없으며, 이를 위반할 시에는 4사  
사규 및 관련 법규에 의해 제재를 받을 수 있습니다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

- 설정 값에 따라 적용 알고리즘이 결정된다.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES GCM 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	AES GCM 키 값
CRYPTO_KE_CIPHER_IV	5	AES GCM IV 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.6.2. CHACHA20-POLY1305

Crypto\_PrimitiveInfoType < **HaeCryptoLibPrimitives** >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_AEADENCRYPT	CRYPTO_ALGOFAM_CHACHA	CRYPTO_ALGOFAM_POLY1305	CRYPTO_ALGOMODE_20ROUNDS

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	평문 데이터 바이트 사이즈
const uint8*	secondaryInputPtr	Aad 데이터 포인터
uint32	secondaryInputLength	Aad 데이터의 바이트 사이즈
uint8*	outputPtr	암호화된 데이터 포인터
uint32*	outputLengthPtr	암호화된 데이터 길이 포인터
uint8*	secondaryOutputPtr	암호문의 검증 Tag 데이터 포인터
uint32*	secondaryOutputLengthPtr	암호문의 검증 Tag 데이터 바이트 사이즈

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	CHACHA20-POLY1305 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	CHACHA20-POLY1305 키 값 (32-byte)
CRYPTO_KE_CIPHER_IV	5	CHACHA20-POLY1305 nonce 값 (12-byte)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.7. CRYPTO\_AEADDECRYPT

### 6.1.7.1. AES GCM

Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_AEADDECRYPT	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_GCM

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	복호화할 데이터 포인터
uint32	inputLength	복호화할 데이터 바이트 사이즈
const uint8*	secondaryInputPtr	Aad 데이터 포인터
uint32	secondaryInputLength	Aad 데이터의 바이트 사이즈
const uint8*	tertiaryInputPtr	암호문의 검증 Tag 데이터 포인터
uint32	tertiaryInputLength	암호문의 검증 Tag 데이터 바이트 사이즈
uint8*	outputPtr	복호화된 데이터 저장 포인터
uint32*	outputLengthPtr	복호화된 데이터 길이 포인터
Crypto_VerifyResultType*	verifyPtr	검증 결과

- **HaeCryptoLibPrimitives 주의 가이드**
  - CRYPTO\_OPERATIONMODE\_SINGLECALL 사용 시, g\_cryptoLib\_gcmLastUpdateCall = 1 설정이 필요
  - 이 외 모드에서 데이터 블록 입력 시, g\_cryptoLib\_gcmLastUpdateCall = 0로 설정 후 inputPtr은 16 바이트의 배수만 입력 가능하다.  
마지막 데이터 블록 입력 시, g\_cryptoLib\_gcmLastUpdateCall = 1로 설정 후에는 16 바이트 외의 길이도 가능하다.  
추가 관련 사항은 CryptoLib 매뉴얼을 참고.
- **HaeHsmPrimitives 주의 가이드**
  - HSM 보안 모듈의 기능 제한으로 연산할 입력 데이터와 Aad의 길이는 최대 128 바이트만 지원한다.
  - CRYPTO\_OPERATIONMODE\_UPDATE 모드에서는 HSM Driver API 정상 동작을 위해 입력 데이터, Aad, Tag, 출력 데이터 저장 포인터 등 모든 파라미터들은 설정되어 있어야 한다.
  - 따라서 CRYPTO\_OPERATIONMODE\_SINGLECALL 동작 모드를 사용한다.

Crypto\_AlgorithmInfoType

Type	Parameter	Description
uint32	keyLength	The key length in bytes to be used with that algorithm 적용 AES Key Byte Length <ul style="list-style-type: none"> <li>• AES-128 : 16</li> <li>• AES-192 : 24 (<b>HaeHsmPrimitives 불가</b>)</li> </ul>

일반(Anyuser)/종원 본 문서는 HyundaiAutoever의 정보자산이므로 무단으로 전제 및 복제할 수 없으며, 이를 위반할 시에는 당사  
사규 및 관련 법규에 의해 제재를 받을 수 있습니다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

- AES-256 : 32
- 설정 값에 따라 적용 알고리즘이 결정된다.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	AES GCM 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	AES GCM 키 값
CRYPTO_KE_CIPHER_IV	5	AES GCM IV 값

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.7.2. CHACHA20-POLY1305

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_AEADDECRYPT	CRYPTO_ALGOFAM_CHACHA	CRYPTO_ALGOFAM_POLY1305	CRYPTO_ALGOMODE_20ROUNDS

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	복호화할 데이터 포인터
uint32	inputLength	복호화할 데이터 바이트 사이즈
const uint8*	secondaryInputPtr	Aad 데이터 포인터
uint32	secondaryInputLength	Aad 데이터의 바이트 사이즈
const uint8*	tertiaryInputPtr	암호문의 검증 Tag 데이터 포인터
uint32	tertiaryInputLength	암호문의 검증 Tag 데이터 바이트 사이즈
uint8*	outputPtr	복호화된 데이터 저장 포인터
uint32*	outputLengthPtr	복호화된 데이터 길이 포인터
Crypto_VerifyResultType*	verifyPtr	검증 결과

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	CHACHA20-POLY1305 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_CIPHER_KEY	1	CHACHA20-POLY1305 키 값 (32-byte)
CRYPTO_KE_CIPHER_IV	5	CHACHA20-POLY1305 nonce 값 (12-byte)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.8. CRYPTO\_SIGNATUREGENERATE

### 6.1.8.1. RSASSA-PKCS1\_v1\_5

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	inputPtr 바이트 사이즈
uint8*	outputPtr	256 Byte의 생성된 서명 값
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	RSA 개인키 값 (512-byte = N    D)



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.8.2. RSASSA-PSS

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PSS

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_RSASSA_PSS
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_RSASSA_PSS
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PSS
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_RSASSA_PSS

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터 (HaeHsmPrimitive의 경우, HSM_RSA_PKCS1_PSS_SIGNGENERATE_INTPUT_t 로 입력)
uint32	inputLength	inputPtr 바이트 사이즈
uint8*	outputPtr	256 Byte의 생성된 서명 값
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	RSA 개인키 값 (512-byte = N    D)
CRYPTO_KE_SIGNATURE_SALT	1013	RSASSA-PSS salt 값 (HaeCryptoLibPrimitive만 해당)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.8.3. ECDSA

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_NOT_SET

- SHA256은 Curve Type이 P256R1만을, SHA512은 Curve Type이 P521R1만을 제공한다.

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	inputPtr 바이트 사이즈
uint8*	outputPtr	생성되는 서명 값
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

- 생성되는 서명 값의 크기는 Curve Type에 의존한다.  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P160R1 : 2 \* 20bytes  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P224R1 : 2 \* 28bytes  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P256R1 : 2 \* 32bytes  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P521R1 : 2 \* 66bytes

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	ECDSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	ECC 개인키 값
CRYPTO_KE_SIGNATURE_SALT	1013	ECDSA salt 값 (HaeHsmPrimitives의 경우, 선택적)
CRYPTO_KE_SIGNATURE_CURVETYPE	29	벤더 ECC Curve Type ID (4-byte, Curve Type별 'Curve Key Length' >= Hash Output Length'인 알고리즘만을 제공한다.)

- 정의된 Curve Type ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.

1. HAEMODULE\_CRYPTO\_CURVETYPE\_P160R1 = 0x01

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

- 2. HAEMODULE\_CRYPTO\_CURVETYPE\_P224R1 = 0x02
- 3. HAEMODULE\_CRYPTO\_CURVETYPE\_P256R1 = 0x03
- 4. HAEMODULE\_CRYPTO\_CURVETYPE\_P521R1 = 0x05

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.8.4. EDDSA ED448

##### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREGENERATE	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHAKE256	CRYPTO_ALGOMODE_NOT_SET

##### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	HAEMODULE_EDDSA_ED448_INTPUT_t 포인터
uint32	inputLength	inputPtr 바이트 사이즈
uint8*	outputPtr	생성되는 114byte 서명 값
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

- Update operation mode는 1회 호출만 가능하다.

##### HAEMODULE\_EDDSA\_ED448\_INTPUT\_t

Name	HAEMODULE_EDDSA_ED448_INTPUT_t		
Type	Structure		
Element	uint8*	message	서명할 메시지
	uint32	messageLength	서명할 메시지의 길이
	uint8*	context	서명에 필요한 Context (256 바이트 미만)
	uint32	contextLength	Context의 길이
Description	ECC EDDSA ED448 알고리즘의 입력 데이터 정보		

- HaeHsmPrimitives 주의 가이드
- HSM 보안 모듈의 기능 제한으로 연산할 입력 데이터 message와 context는 최대 64바이트만 지원한다.

##### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	ECC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	ED448 개인키 값

- 현재 EDDSA로 ED448만 지원하므로 별도 Curve Type은 설정하지 않는다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.9. CRYPTO\_SIGNATUREVERIFY

### 6.1.9.1. RSASSA-PKCS1\_v1\_5

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREVERIF	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	inputPtr 바이트 사이즈
const uint8*	secondaryInputPtr	256byte의 검증할 서명 값
uint32	secondaryInputLength	secondaryInputPtr 바이트 사이즈
Crypto_VerifyResultType*	verifyPtr	검증 결과 값

- 서명 값의 최상위 바이트가 0x00일 경우, +1byte까지 입력을 허용한다.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	RSA 공개키 값 (260-byte = N    E (4-byte))

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.9.2. RSASSA-PSS

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PSS

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_RSASSA_PSS
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_RSASSA_PSS
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_RSASSA_PSS
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_RSA	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_RSASSA_PSS

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터 (HaeHsmPrimitive의 경우, HSM_RSA_PKCS1_PSS_SIGNVERIFY_INTPUT_t 로 입력)
uint32	inputLength	inputPtr 바이트 사이즈
const uint8*	secondaryInputPtr	256byte의 검증할 서명 값
uint32	secondaryInputLength	secondaryInputPtr 바이트 사이즈
Crypto_VerifyResultType*	verifyPtr	검증 결과 값

- 서명 값의 최상위 바이트가 0x00일 경우, +1byte까지 입력을 허용한다.

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	RSA 공개키 값 (260-byte = N    E (4-byte))
CRYPTO_KE_SIGNATURE_SALT	1013	RSASSA-PSS salt 값의 바이트 길이 (4-byte 형식으로 표기) (HaeCryptoLibPrimitives만 해당됨)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.9.3. ECDSA

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_NOT_SET

- SHA256은 Curve Type이 P256R1만을, SHA512은 Curve Type이 P521R1만을 제공한다.

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA1	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_224	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_384	CRYPTO_ALGOMODE_NOT_SET
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHA2_512	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	평문 데이터 포인터
uint32	inputLength	inputPtr 바이트 사이즈
const uint8*	secondaryInputPtr	검증할 서명 값
uint32	secondaryInputLength	secondaryInputPtr 바이트 사이즈
Crypto_VerifyResultType*	verifyPtr	검증 결과 값

- 서명 값의 r, s 각각 상위 바이트가 0x00일 경우, +1byte 입력을 허용하여, +2byte까지 입력을 허용한다.
- 검증할 서명 값의 크기는 Curve Type에 의존한다.  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P160R1 : 2 \* 20bytes  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P224R1 : 2 \* 28bytes  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P256R1 : 2 \* 32bytes  
 HAEMODULE\_CRYPTO\_CURVETYPE\_P521R1 : 2 \* 66bytes

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	ECC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	ECC 공개키 값
CRYPTO_KE_SIGNATURE_SALT	1013	ECDSA salt 값 (HaeHsmPrimitives의 경우, 선택적)
CRYPTO_KE_SIGNATURE_CURVETYPE	29	벤더 ECC Curve Type ID (4-byte, Curve Type별 'Curve Key Length' >= Hash Output Length'인 알고리즘만을 제공한다.)

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

- 정의된 Curve Type ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.
  - HAEMODULE\_CRYPTOCURVETYPE\_P160R1 = 0x01
  - HAEMODULE\_CRYPTOCURVETYPE\_P224R1 = 0x02
  - HAEMODULE\_CRYPTOCURVETYPE\_P256R1 = 0x03
  - HAEMODULE\_CRYPTOCURVETYPE\_P521R1 = 0x05



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

#### 6.1.9.4. EDDSA ED448

##### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_SIGNATUREVERIFY	CRYPTO_ALGOFAM_ECCNIST	CRYPTO_ALGOFAM_SHAKE256	CRYPTO_ALGOMODE_NOT_SET

##### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	HAEMODULE_EDDSA_ED448_INTPUT_t 포인터
uint32	inputLength	inputPtr 바이트 사이즈
const uint8*	secondaryInputPtr	검증할 114byte 서명 값
uint32	secondaryInputLength	outputPtr 바이트 사이즈
Crypto_VerifyResultType*	verifyPtr	검증 결과

- 서명 값의 r, s 각각 상위 바이트가 0x00일 경우, +1byte 입력을 허용하여, +2byte까지 입력을 허용한다.
- HaeHsmPrimitives 주의 가이드**
  - ECC\_EDDSA\_ED448\_SIGN\_t는 HSM 모듈 사용자 매뉴얼을 참고 바랍니다.

##### HAEMODULE\_EDDSA\_ED448\_INTPUT\_t

Name	HAEMODULE_EDDSA_ED448_INTPUT_t		
Type	Structure		
Element	uint8*	message	서명할 메시지
	uint32	messageLength	서명할 메시지의 길이
	uint8*	context	서명에 필요한 Context (256 바이트 미만)
	uint32	contextLength	Context의 길이
Description	ECC EDDSA ED448 알고리즘의 입력 데이터 정보		

- HaeHsmPrimitives 주의 가이드**
  - HSM 보안 모듈의 기능 제한으로 연산할 입력 데이터 message와 context는 최대 64바이트만 지원한다.

##### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	ECC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_SIGNATURE_KEY	1	ED448 공개키 값

- 현재 EDDSA로 ED448만 지원하므로 별도 Curve Type은 설정하지 않는다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.10. CRYPTO\_RANDOMGENERATE

### 6.1.10.1. DRBG

#### 주의:

DRBG는 Seed가 필수적으로 입력되어야 한다.

따라서 RandomGenerate 사용 전, RandomSeed 서비스를 반드시 사용하여야 하고, RandomGenerate 사용 중, 오류를 반환할 시에도 RandomSeed 서비스를 사용하여야 한다.

- CSM R4.4 CsmJobKeyDeriveAlgorithmFamily 미지원
  - CRYPTO\_ALGOFAM\_CUSTOM 설정 후, AlgorithmFamilyCustom에 값 입력
- CSM R4.4 CsmJobKeyDeriveAlgorithmSecondaryFamily 미지원
  - CRYPTO\_ALGOFAM\_CUSTOM 설정 후, AlgorithmSecondaryFamilyCustom에 값 입력

Crypto\_PrimitiveInfoType < [HaeCryptoLibPrimitives](#) >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_RANDOMGENERATE	CRYPTO_ALGOFAM_DRBG	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
uint8*	outputPtr	랜덤 넘버 데이터 포인터
uint32*	outputLengthPtr	랜덤 넘버 데이터 길이 포인터 0 또는 1024 초과 값을 입력할 경우, E_PARAM_VALUE 발생

### 6.1.10.2. PRNG

Crypto\_PrimitiveInfoType < [HaeHsmPrimitives](#) >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_RANDOMGENERATE	CRYPTO_ALGOFAM_RNG	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
uint8*	outputPtr	랜덤 넘버 데이터 포인터
uint32*	outputLengthPtr	랜덤 넘버 데이터 길이 포인터 0 또는 1024 초과 값을 입력할 경우, E_PARAM_VALUE 발생

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.10.3. TRNG

Crypto\_PrimitiveInfoType < HaeHsmPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_RANDOMGENERATE	CRYPTO_ALGOFAM_RNG	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_CUSTOM

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
uint8*	outputPtr	랜덤 넘버 데이터 포인터
uint32*	outputLengthPtr	랜덤 넘버 데이터 길이 포인터 0 또는 1024 초과 값을 입력할 경우, E_PARAM_VALUE 발생

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.11. CRYPTO\_RANDOMSEED

### 6.1.11.1. DRBG

- CSM R4.4 CsmJobKeyDeriveAlgorithmFamily 미지원
  - CRYPTO\_ALGOFAM\_CUSTOM 설정 후, AlgorithmFamilyCustom에 값 입력
- CSM R4.4 CsmJobKeyDeriveAlgorithmSecondaryFamily 미지원
  - CRYPTO\_ALGOFAM\_CUSTOM 설정 후, AlgorithmSecondaryFamilyCustom에 값 입력

Crypto\_PrimitiveInfoType < [HaeCryptoLibPrimitives](#) >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_RANDOMGENERATE	CRYPTO_ALGOFAM_DRBG	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
uint8*	inputPtr	DRBG의 입력 Seed 데이터 포인터 (최초 Seed는 48바이트 이상 입력해야 한다)
uint32	inputLength	inputPtr 바이트 사이즈

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	DRBG 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_RANDOM_ALGORITHM	4	벤더 DRBG의 알고리즘 ID (Key Interface 사용 시, 4-byte)

- 정의된 DRBG ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.

1. HAEMODULE\_CRYPTODRBG\_SHA256 = 0x01

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.12. CRYPTO\_KEYEXCHANGE\_CALCPUBVAL

### 6.1.12.1. Diffie-Hellman

Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_KEYEXCHANGE_CALCPUBVAL	CRYPTO_ALGOFAM_DH	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
uint8*	outputPtr	생성된 공개키 값의 저장 포인터
uint32*	outputLengthPtr	outputPtr 바이트 사이즈 (256 바이트)

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	Key exchange를 위한 Key

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_KEYEXCHANGE_BASE	8	Modulus P 값과 Generator G 값을 연결, 512 바이트 사이즈
CRYPTO_KE_KEYEXCHANGE_PRIVKEY	9	개인 키, 256 바이트 사이즈
CRYPTO_KE_KEYEXCHANGE_ALGORITHM	12	벤더 Key Exchange의 알고리즘 ID (Key Interface 사용 시, 4-byte)

- 정의된 Key Exchange ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.

- HAEMODULE\_CRYPTODH = 0x01
- HAEMODULE\_CRYPTODH\_HSM = 0x81

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.12.2. ECDH

Crypto\_PrimitiveInfoType < HaeCryptoLibPrimitives >

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_KEYEXCHANGE_CALCPUBVAL	CRYPTO_ALGOFAM_DH	CRYPTO_ALGOFAM_ECNIST	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
uint8*	outputPtr	생성된 공개키 값의 저장 포인터
uint32*	outputLengthPtr	outputPtr 바이트 사이즈

- 생성될 공개키 값의 크기는 Curve Type에 의존한다.  
 HAEMODULE\_CRYPTOCURVETYPE\_P160R1 : 2 \* 20bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P224R1 : 2 \* 28bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P256R1 : 2 \* 32bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P521R1 : 2 \* 66bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_CURVE448 : 56bytes

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	Key exchange를 위한 Key

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_KEYEXCHANGE_PRIVKEY	9	ECC 개인 키
CRYPTO_KE_KEYEXCHANGE_CURVETYPE	29	벤더 ECC Curve Type ID (4-byte)
CRYPTO_KE_KEYEXCHANGE_ALGORITHM	12	벤더 Key Exchange의 알고리즘 ID (Key Interface 사용 시, 4-byte)

- 정의된 Curve Type ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.
  - HAEMODULE\_CRYPTOCURVETYPE\_P160R1 = 0x01
  - HAEMODULE\_CRYPTOCURVETYPE\_P224R1 = 0x02
  - HAEMODULE\_CRYPTOCURVETYPE\_P256R1 = 0x03
  - HAEMODULE\_CRYPTOCURVETYPE\_P521R1 = 0x05
  - HAEMODULE\_CRYPTOCURVETYPE\_CURVE448 = 0x06
- 정의된 Key Exchange ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.
  - HAEMODULE\_CRYPTOECDH = 0x02
  - HAEMODULE\_CRYPTOECDH\_HSM = 0x82

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.13. CRYPTO\_KEYEXCHANGECALCSECRET

### 6.1.13.1. Diffie-Hellman

Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_KEYEXCHANGECALCSECRET	CRYPTO_ALGOFAM_DH	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	비밀 값을 공유할 상대 멤버의 공개키 구조체
uint32	inputLength	inputPtr 바이트 사이즈

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	RSA 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_KEYEXCHANGE_BASE	8	Modulus P 값, 256 바이트 사이즈
CRYPTO_KE_KEYEXCHANGE_PRIVKEY	9	개인 키, 256 바이트 사이즈

- 결과 값은 다음 Element에 저장된다.

key element Name	key element ID	Comments
CYRPTO_KE_KEYEXCHANGE_SHAREDVALUE	1	Secret Number, 256 바이트 사이즈 상대방의 공개키와 나의 개인키를 연산하여 비밀값을 생성한다.
CRYPTO_KE_KEYEXCHANGE_ALGORITHM	12	벤더 Key Exchange의 알고리즘 ID (Key Interface 사용 시, 4-byte)

- 정의된 Key Exchange ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.

- HAEMODULE\_CRYPTODH = 0x01
- HAEMODULE\_CRYPTODH\_HSM = 0x81



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.13.2. ECDH

알고리즘에 사용되는 Curve 타입에 따라 여러 알고리즘을 지원한다. Crypto\_PrimitiveInfoType의 설정은 동일하며 cryptoKey의 Key Element 중 'CRYPTO\_KE\_KEYEXCHANGE\_CURVETYPE'에 설정된 값에 따라 알고리즘이 적용된다. 따라서 사용자는 본 알고리즘을 이용하기 전에 Crypto\_KeyElementSet 와 Crypto\_KeySetValid API를 이용하여 'CRYPTO\_KE\_KEYEXCHANGE\_CURVETYPE'을 포함한 필요한 Key Element를 설정하도록 한다.

#### 주의:

현재 사용하고 있는 HSM 보안 모듈의 지원 여부에 따라 일부 Curve 타입은 사용하지 못할 수도 있다. 따라서 현재 사용 중인 HSM의 매뉴얼을 확인하여 지원 여부를 확인하도록 한다.

#### Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_KEYEXCHANGE_CALC_SECRET	CRYPTO_ALGOFAM_DH	CRYPTO_ALGOMODE_ECCNIST	CRYPTO_ALGOMODE_NOT_SET

#### Crypto\_JobPrimitiveInputOutputType

Type	Parameter	Description
const uint8*	inputPtr	상대방의 공개키 (PXXXR1 curve의 경우, x좌표와 y좌표 연결)
uint32	inputLength	inputPtr 바이트 사이즈

- 공개키 값의 크기는 Curve Type에 의존한다.  
 HAEMODULE\_CRYPTOCURVETYPE\_P160R1 : 2 \* 20bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P224R1 : 2 \* 28bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P256R1 : 2 \* 32bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P521R1 : 2 \* 66bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_CURVE448 : 56bytes

#### Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	ECC 키 번호

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_KEYEXCHANGE_PRIVKEY	9	ECC 개인 키
CRYPTO_KE_KEYEXCHANGE_CURVETYPE	29	벤더 ECC Curve Type ID (4-byte)
CRYPTO_KE_KEYEXCHANGE_ALGORITHM	12	벤더 Key Exchange의 알고리즘 ID (Key Interface 사용 시, 4-byte)

- 정의된 Curve Type ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.

- HAEMODULE\_CRYPTOCURVETYPE\_P160R1 = 0x01

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

2. HAEMODULE\_CRYPTOCURVETYPE\_P224R1 = 0x02
  3. HAEMODULE\_CRYPTOCURVETYPE\_P256R1 = 0x03
  4. HAEMODULE\_CRYPTOCURVETYPE\_P521R1 = 0x05
  5. HAEMODULE\_CRYPTOCURVETYPE\_CURVE448 = 0x06
- 정의된 Key Exchange ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.
1. HAEMODULE\_CRYPTOECDH = 0x02
  2. HAEMODULE\_CRYPTOECDH\_HSM = 0x82
- 결과 값은 다음 Element에 저장된다.

key element Name	key element ID	Comments
CYRPTO_KE_KEYEXCHANGE_SHAREDVALUE	1	Secret Number. 상대방의 공개키와 나의 개인키를 연산하여 비밀값을 생성한다.

- 생성된 Secure Number 값의 크기는 Curve Type에 의존한다.  
 HAEMODULE\_CRYPTOCURVETYPE\_P160R1 : 20bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P224R1 : 28bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P256R1 : 32bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_P521R1 : 66bytes  
 HAEMODULE\_CRYPTOCURVETYPE\_CURVE448 : 56bytes

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.1.14. CRYPTO\_KEYDERIVE

### 6.1.14.1. PBKDF2

- CSM R4.4 CsmJobKeyDeriveAlgorithmSecondaryFamily 미지원 (R22~11 지원)
  - CRYPTO\_ALGOFAM\_CUSTOM 설정 후, AlgorithmSecondaryFamilyCustom에 값 입력

Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_KEYDERIVE	CRYPTO_ALGOFAM_PBKDF2	CRYPTO_ALGOFAM_SHA2_256	CRYPTO_ALGOMODE_HMAC

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	Holds the identifier of the key which is used for key derivation.
uint32	targetCryptoKeyId	Holds the identifier of the key which is used to store the derived key.

- cryptoKey는 서비스 실행 전에 다음과 같은 Key Element에 유효한 값이 설정되어 있어야 한다.
- 이때 cryptoKey와 targetCryptoKey는 반드시 서로 다른 Key로 설정해야 한다.

key element Name	key element ID	Comments
CRYPTO_KE_KEYDERIVATION_PASSWORD	1	PBKDF2 비밀번호 값
CRYPTO_KE_KEYDERIVATION_HSMKEYINDEX	1001	PBKDF2 HSM Key index 값 (Key Interface 사용 시, 4-byte)
CRYPTO_KE_KEYDERIVATION_SALT	13	PBKDF2 salt 값
CRYPTO_KE_KEYDERIVATION_ITERATIONS	14	PBKDF2 iteration 값
CRYPTO_KE_KEYDERIVATION_ALGORITHM	15	벤더 Key Derivation의 알고리즘 ID (Key Interface 사용 시, 4-byte)

- 정의된 Key Derivation ID는 Crypto\_76\_HaeModule\_UserTypes.h에서 확인할 수 있다.
  0. HAEMODULE\_CRYPTOPBKDF2\_HMAC\_SHA256 = 0x01
  1. HAEMODULE\_CRYPTOPBKDF2\_HMAC\_SHA256\_HSM = 0x81
- HaeHsmPrimitives 특화 기능 가이드
  - PBKDF2의 입력으로 필요한 password는 사용자의 입력으로 받거나, HSM Framework의 보안 영역에 미리 저장된 키 값으로 대체하여 사용할 수 있다. (상세 내용은 HSM\_Framework\_UserManual\_Crypto\_Service 매뉴얼 참고)
  - Password를 HSM 내부 Key로 대체하여 사용하고자 하는 경우에는 CRYPTO\_KE\_KEYDERIVATION\_HSMKEYINDEX의 KeyElementValue에 유효한 HSM Key Index가 설정되어 있어야 한다. (HSM Key Type/HSM Key Index는 설정하지 않아야 함.)
  - Runtime 중 HSM Key index를 변경하여 job을 수행하고자 한다면 Crypto\_KeyElementSet 와 Crypto\_KeySetValid API를 이용하여 CRYPTO\_KE\_KEYDERIVATION\_HSMKEYINDEX에 원하는 HSM Key Index로 변경할 수 있다. (사용 예. KeyElementSet(CRYPTO\_KE\_KEYDERIVATION\_HSMKEYINDEX, HsmKeyIndex, sizeof(HsmKeyIndex)))
  - 만일 CRYPTO\_KE\_KEYDERIVATION\_PASSWORD 와 CRYPTO\_KE\_KEYDERIVATION\_HSMKEYINDEX에 둘

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

다 유효한 값이 설정되어 있다면 설정된 HSM Key Index를 기준으로 Pbkdf2를 수행한다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.1.15. CRYPTO\_KEYSETVALID

Crypto\_PrimitiveInfoType

Service	Algorithm Family	Algorithm Secondary Family	Algorithm Mode
CRYPTO_KEYSETVALID	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOFAM_NOT_SET	CRYPTO_ALGOMODE_NOT_SET

Crypto\_JobType

Type	Parameter	Description
uint32	cryptoKeyId	대상 키

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 6.2. 신규 사용자 Crypto Primitive 추가 방법

“Ecud\_Crypto\_76\_HaeModule.arxml”에 신규로 Crypto Primitive를 추가할 경우를 설명한다.

다음과 같이 CryptoLib Object의 CryptoPrimitives에 신규로 “HaeCryptoLib\_Primitive\_Sha256”이라는 이름의 Primitive를 생성하였다고 가정한다.

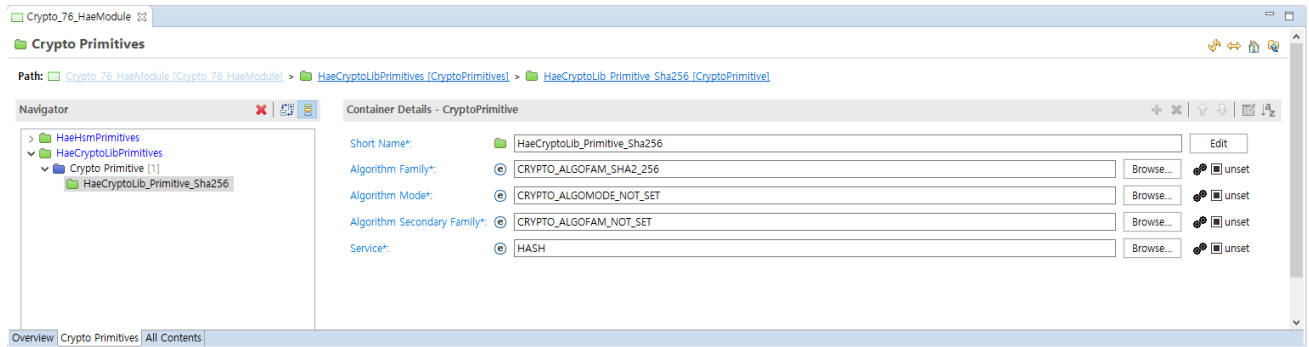


그림 7. 신규 CryptoPrimitive 추가

Crypto Generator 툴에 의해 생성되는 “Crypto\_76\_HaeModule\_Cfg.c” 파일에 다음과 같이 신규 Primitive가 추가된다.

```
const CryptoPrimitive HaeCryptoLib_CryptoPrimitives[] =
{
    {
        CRYPTO_HASH,
        CRYPTO_ALGOFAM_SHA2_256,
        CRYPTO_ALGOFAM_NOT_SET,
        CRYPTO_ALGOMODE_NOT_SET,
        (Crypto_PrimitiveProcess)HaeCryptoLib_Primitive_Sha256
    },
};
```

신규 Primitive와 연결된 실행 함수가 Generate 하면 Primitive 이름으로 생성된다. 따라서 사용자는 “Crypto\_PrimitiveProcess” 타입의 신규 Primitive 실행 함수를 구현해 주어야 한다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 6.3. Asynchronous Job을 위한 Crypto MainFunction 설정

Crypto MainFunction은 Asynchronous Job 처리를 위해서 일정 시간마다 수행될 수 있도록 설정되어야 한다. 그러나 Synchronous Job 처리 만을 필요로 할 경우에는 반드시 필요하지는 않다.

#### 8.4 Scheduled functions

##### 8.4.1.1 Crypto\_MainFunction

The `Crypto_MainFunction()` is necessary for asynchronous job processing. For synchronous job processing providing the main function is optional.

[SWS\_Crypto\_91012] [

<b>Service name:</b>	Crypto_MainFunction
<b>Syntax:</b>	void Crypto_MainFunction( void )
<b>Service ID[hex]:</b>	0x0c
<b>Description:</b>	If asynchronous job processing is configured and there are job queues, the function is called cyclically to process queued jobs.
<b>Available via:</b>	SchM_Crypto.h

HaeModule을 위한 Crypto MainFunction은 “Bswmd\_Crypto\_76\_HaeModule.arxml”에 정의되어 있다. 만약 Asynchronous Job과 Job Queue가 설정되어 있다면 이 함수를 일정 시간마다 수행될 수 있도록 설정해야 한다.

HaeModule은 각 Object 별로 Asynchronous Job 처리를 위해 MainFunction이 각각 생성된다. 따라서 정상적인 Job 처리를 위해서 각 MainFunction이 일정 시간마다 수행 될 수 있도록 설정한다.

No.	Object	MainFunction
1	HaeHsm	Crypto_HaeHsm_MainFunction
2	HaeCryptoLib	Crypto_HaeCryptoLib_MainFunction

표 4. Object 별 Crypto MainFunction

MainFunction 함수는 Generate 시 Object 별로 “Crypto\_76\_HaeModule\_Cfg.c” 파일에 생성된다. 따라서 Object가 사용되지 않아 삭제할 경우 해당 Object의 MainFunction은 생성되지 않는다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 7. Key Management

### 7.1. Crypto Key Element 설정

CryptoKeyElement 설정 Format은 CRYPTO\_KE\_FORMAT\_BIN\_OCTET 만을 지원한다.

#### 7.1.1. HSM Crypto Key Element 설정

HSM은 내부에 Crypto Key를 별도로 관리하고 있다. 사양에 따라 HSM에서 제공하는 Key 테이블은 다르다. 자세한 사항은 HSM 모듈과 같이 배포되는 사용자 매뉴얼 문서의 “Key Management” 내용을 참고하길 바란다.

HSM 내부 Key를 HaeModule Crypto Driver의 Key Element와 연결시키기 위해서는 특별한 설정이 필요하다. 이에 대한 설정 방법을 설명한다.

Type	AES Key			RSA Key			ECC p256r1 Key			ECC p521r1 Key			ECC ED448 Key [Sign/Verify]			ECC X448 Key [ECDH]		
	Key Index	영역	목적	Key Index	영역	목적	Key Index	영역	목적	Key Index	영역	목적	Key Index	영역	목적	Key Index	영역	목적
HKMC PSK Pre Shared Key	#1~#5	HKMC PSK	OTA(aSIMS)	#1	HKMC PSK	aSIMS	#1	HKMC PSK	aSIMS	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공
	#6~#10		OTA(PST)															
	#11~#15		FoD															
	#16~#50		TBD															
Tier PSK Pre Shared Key	#101	Tier PSK	TBD	#101	Tier PSK	TBD	#101	Tier PSK	TBD	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공
	#102			#102														
	#103			#103														
	#104			#104														
	#105			#105														
UDK (User Defined Key)	#201	UDK-AES128 (Updatable)	TBD	#201	UDK	TBD	#201	UDK	TBD	#201	UDK	TBD	#201	UDK	TBD	#201	UDK	TBD
	#202			#202			#202			#202								
	#203																	
	#204																	
	#205	UDK-AES256 (Updatable)			미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공	미제공		
	#206																	
	#207																	
	#208																	
HOST TEMP KEY	#301	RAM KEY AES 128	TBD	#301	RAM KEY	TBD	#301	RAM KEY	TBD	#301	RAM KEY	TBD	#301	RAM KEY	TBD	#301	RAM KEY	TBD
	#302	RAM KEY AES 256																

표 5. HSM Key 테이블

Crypto Key Element와 HSM Key의 연결을 위해 CryptoKeyElement에 “Hsm Key Type”과 Hsm Key Index” 설정을 추가하였다. Hsm Key Type과 Key Index는 “표 5. HSM Key 테이블”과 같이 HSM 모듈의 사용자 매뉴얼에 명시된 Key Type과 Index를 참조해서 설정한다.



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

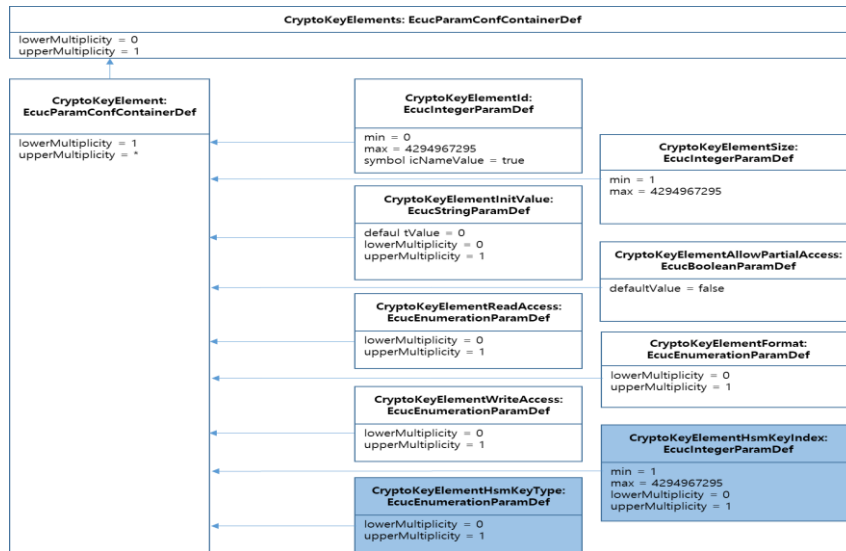


그림 8. HaeModule CryptoKeyElement Container

아래는 HSM의 AES Key의 UDK #201과 연결 예를 보여 주고 있다.

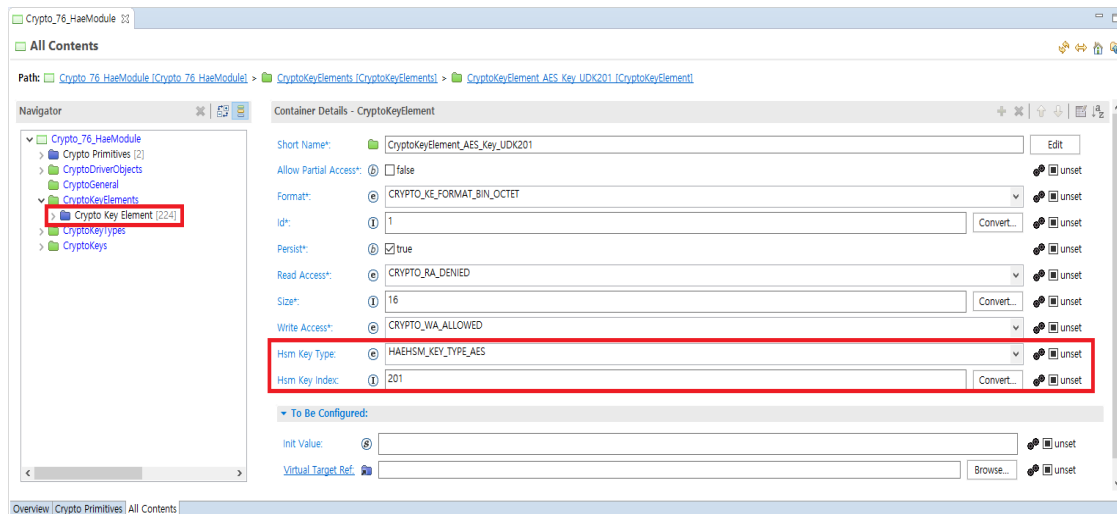


그림 9. HSM Crypto Key Element 설정

만약 Crypto Key Element가 HSM Key와 관련이 없다면 “Hsm Key Type”과 Hsm Key Index”를 설정하지 않도록 한다. 그럼에도 이를설정할 경우 HaeModule Crypto Driver는 설정된 Key Element를 HSM Key Management API 를 사용해서 설정하려고 하여 오동작이 발생한다..

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 7.1.2. HSM Crypto Key Element 특성

Crypto Key Element가 HSM Key와 연결된 경우 “Crypto\_KeySetValid” Interface 호출 시 실질적으로 HSM Driver를 통해서 HSM에 저장된다. 그 이외의 Crypto Key Element는 HOST상의 RAM 상에 저장된다. 예를 들어 AES Key의 경우 Crypto Key Element가 CRYPTO\_KE\_CIPHER\_IV(5)를 가지는 Init. Vector인 Element는 HOST의 RAM에 저장되며, “Crypto\_KeySetValid” Interface 호출 시에도 HSM에 저장되지는 않는다.

HSM에 저장되는 Key의 경우 Key Index으로만 참조할 수 있고 절대 읽을 수는 없다. 따라서 CryptoKeyElement의 CryptoKeyElementReadAccess는 CRYPTO\_RA\_DENIED 속성으로 반드시 설정해야 한다. 그리고, UDK(User Defined Key)와 HTK(Host Temp Key)의 경우에만 CryptoKeyElementWriteAccess의 권한이 활성화된다.

요약하면 HSM 관련 CryptoKeyElement는 다음과 같은 특성을 가진다.

1. HSM에 저장, 관리되는 Key는 HOST에서 절대로 읽을 수 없고 단지 Key Index로만 참조 할 수 있다.
2. 모든 CryptoKeyElement의 CryptoKeyElementReadAccess는 CRYPTO\_RA\_DENIED 속성으로 반드시 설정해야 한다.
3. 모든 CryptoKeyElement의 CryptoKeyElementAllowPartialAccess는 반드시 FALSE로 설정해야 한다.
4. PSK, Tier PSK 키의 CryptoKeyElementWriteAccess는 CRYPTO\_WA\_DENIED 속성으로 반드시 설정해야 한다.
5. UDK(User Defined Key), HTK(Host Temp Key) 키의 CryptoKeyElementWriteAccess는 CRYPTO\_WA\_ALLOWED 속성으로 반드시 설정해야 한다.
6. CryptoKeyElementInitValue는 AES Key의 Init. Vector와 UDK, HOST TEMP KEY 타입만 설정이 가능하다. Driver 초기 시 이 전에 설정된 Key 값이 없을 경우에만 CryptoKeyElementInitValue 값을 적용한다.
7. UDK와 HTK에 이미 설정된 값이 있다면 CryptoKeyElementInitValue는 새로 적용되지 않는다. 따라서 만약 CryptoKeyElementInitValue 값이 중간에 변경되었다고 해도 새로 적용되지 않는다. 이럴 경우 “Crypto\_KeyElementSet”과 “Crypto\_KeySetValid” Interface를 사용해서 직접 UDK, HTK를 설정해야 한다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 7.2. CryptoKeyElementValue 설정 방법

### 7.2.1. CryptoKeyElementInitValue로 입력하는 방법

CryptoKeyElementInitValue는 사용자가 올바른 값을 미리 설정해 두었을 경우 CryptoDriver 초기화시, CryptoKeyElementValue로 초기화된다. 1바이트 Hex값으로 C 언어 Standard의 배열 규칙 적용을 원칙으로 한다. 다음은 올바른 입력 값 예이다.

(예) 0x16, 0x15, 0x7e, 0x2b, 0xa6, 0xd2, 0xae, 0x28, 0x88, 0x15, 0xf7, 0xab

또는 다음과 같은 입력 형식도 가능하나 권장하지 않는다.

(예) 16 15 7e 2b a6 d2 ae 28 88 15 f7 ab

2바이트 이상의 워드(Word)의 경우 HOST의 Endian 형식을 따른다. 예를 들어 HOST가 Little Endian일 경우 다음과 같이 입력한다.

(예) 210309 : 0x85, 0x35, 0x03, 0x00

### 7.2.2. Crypto\_KeyElementSet API로 입력하는 방법

Crypto\_KeyElementSet을 통해 runtime 중에 CryptoKeyElementValue를 변경할 수 있다. Crypto\_KeyElementValid를 통해 Key를 유효한 상태로 설정하여야만 해당 Key를 활용할 수 있다.

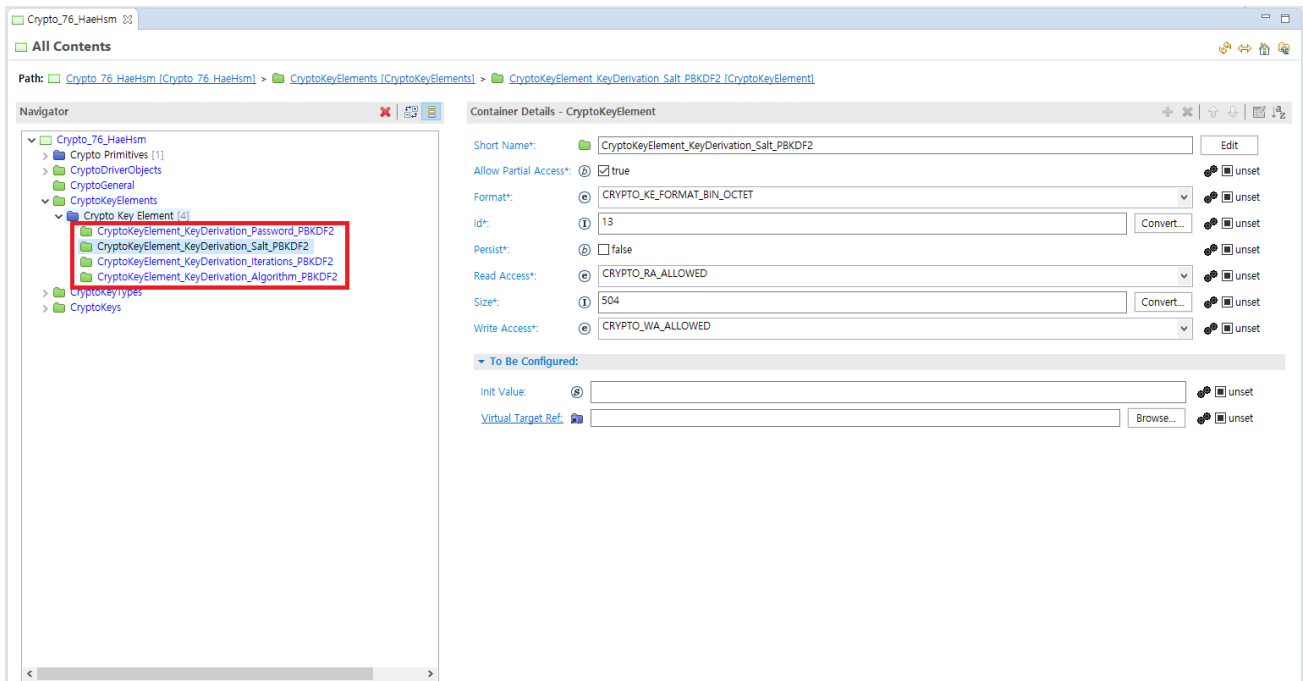
## 7.3. 신규 CryptoKey 추가 방법

“Ecud\_Crypto\_76\_HaeModule.arxml”에 신규로 CryptoKey를 추가할 경우를 설명한다. PBKDF2 알고리즘을 사용한 Key Derivation을 예를 들어 설명한다. AUTOSAR\_SWS\_CryptoServiceManager 매뉴얼의 ‘SWS\_Csm\_01022’을 보면 Key Derivation을 위한 Key Element는 다음과 같이 4개가 필요하다.

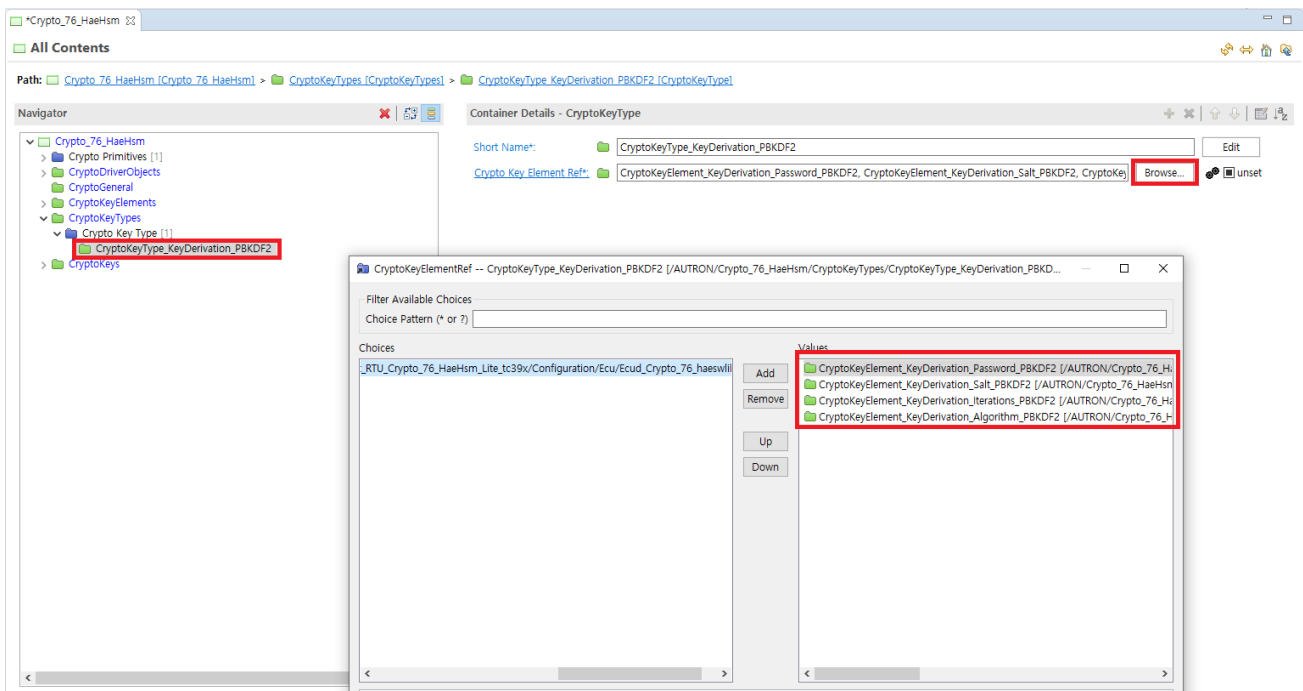
Key Derivation	Password	CRYPTO_KE_KEYDERIVATION_PASSWORD	1	x
	Salt	CRYPTO_KE_KEYDERIVATION_SALT	13	
	Iterations	CRYPTO_KE_KEYDERIVATION_ITERATIONS	14	
	Algorithm	CRYPTO_KE_KEYDERIVATION_ALGORITHM	15	

CryptoKeyElements에 다음과 같이 4개의 Key Element를 생성한다.


	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

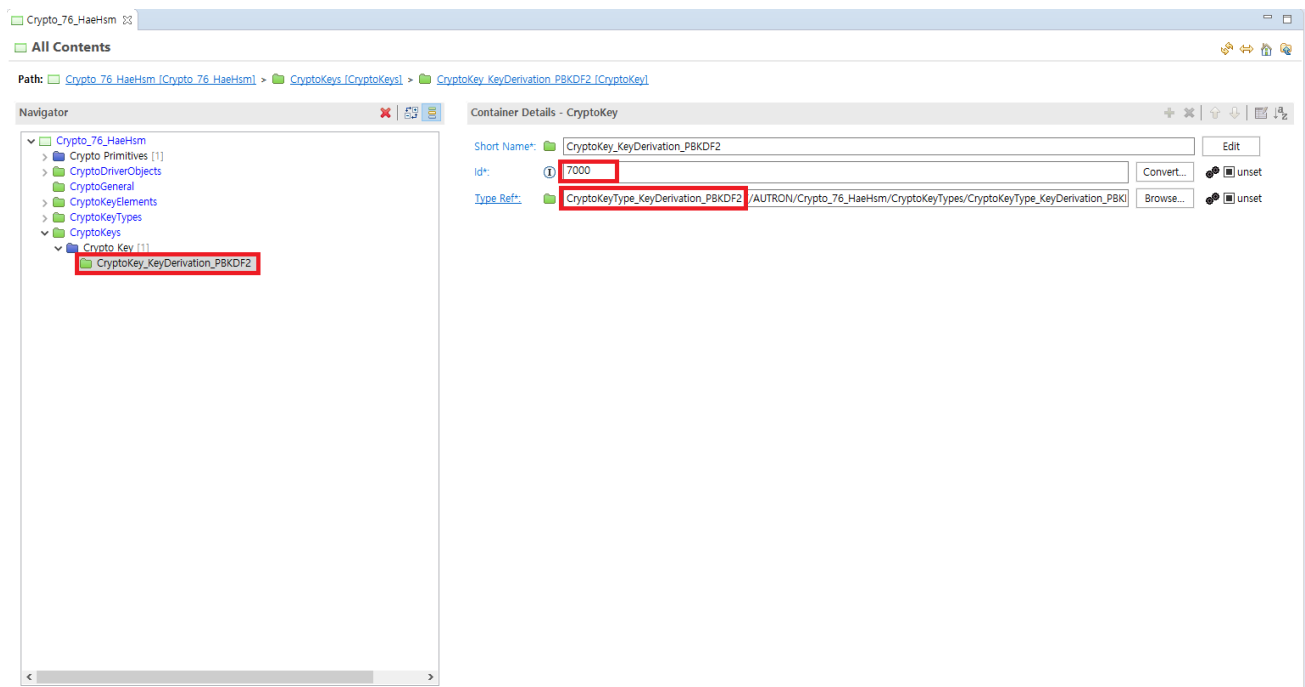


CryptoKeyTypes에 'CryptoKeyType\_KeyDerivation\_PBKDF2'을 생성한다. 그리고 'Crypto Key Element Ref'에 위에서 생성한 4개의 Key Element를 추가한다.




CryptoKeys에 'CryptoKey\_KeyDerivation\_PBKDF2'를 생성한다. 그리고 'Type Ref'에 위에서 생성한 'CryptoKeyType\_KeyDerivation\_PBKDF2'를 추가한다. 여기에서 CryptoKeyId는 7000을 설정하였다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b



사용자는 CryptoKeyId 값을 7000으로하여 여기에서 추가된 Key Derivation의 Key Element를 조작하거나 참조할 수 있다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 8. Crypto Configuration

Crypto 모듈의 Container와 파라미터에 대해서 설명한다. 자세한 파라미터의 의미는 AUTOSAR의 “Specification of Crypto Driver” 문서, 특히 “Configuration specification” 내용을 참조 바란다.

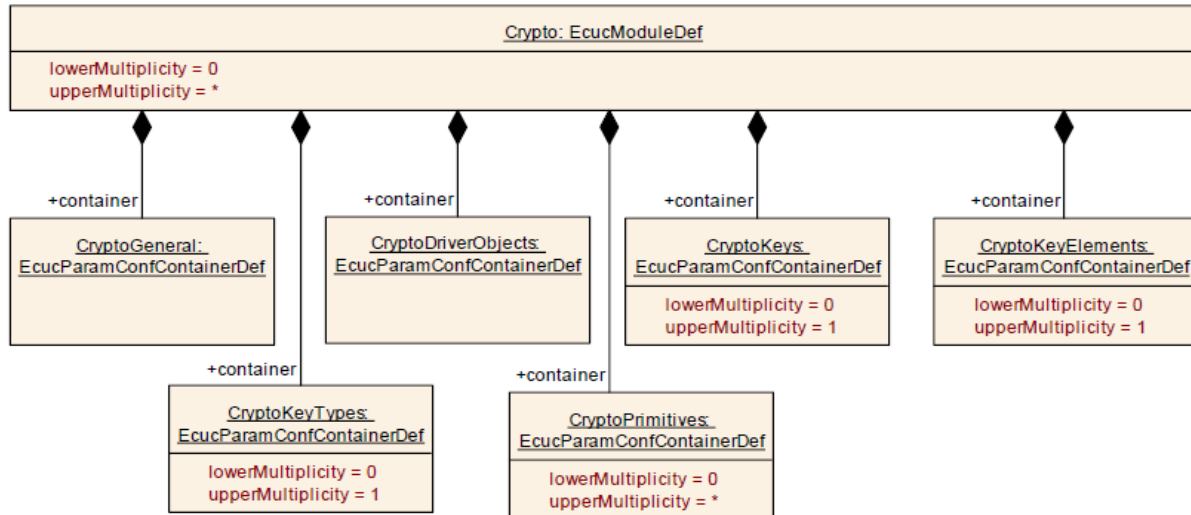



그림 10. AUTOSAR Crypto Container 구조

제공되는 HaeModule Crypto Driver의 ECU Description은 AUTOSAR Crypto Container 구조를 따른다.

그리고 ‘Ecud\_Crypto\_76\_HaeModule.arxml’ 파일에는 HSM 보안 모듈 사용을 위해 필요한 값들이 기본적으로 설정되어 있다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

8.1. CryptoGeneral

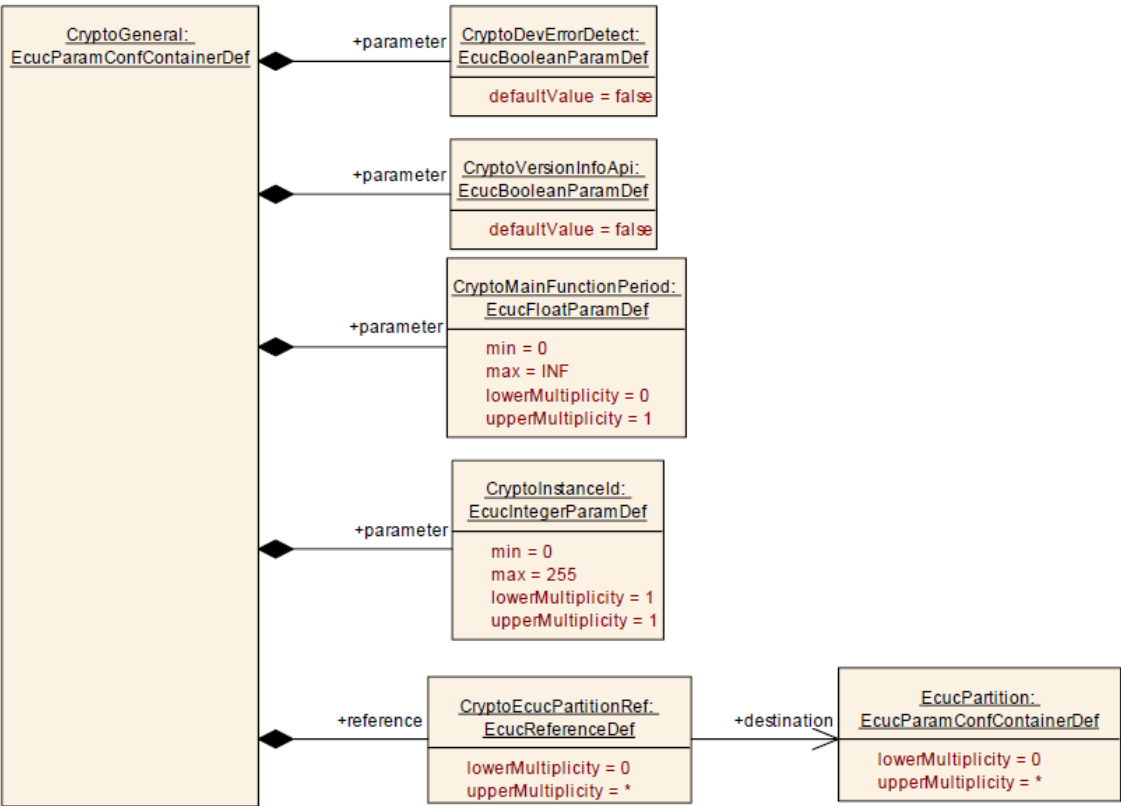


그림 11. AUTOSAR CryptoGeneral Container

HaeModule Crypto Driver의 CryptoGeneral Container 기본 설정 값은 아래 표와 같다. 사용자는 모든 설정 값을 사용 환경에 맞게 변경 가능하다.

Name	CryptoDevErrorDetect	CryptoVersionInfoApi	CryptoMainFunctionPeriod	CryptoInstancelId	CryptoEcucPartitionRef
CryptoGeneral	TRUE	TRUE	0.1	1	NULL

표 6. HaeModule CryptoGeneral Container 내용

## 8.2. CryptoDriverObjects

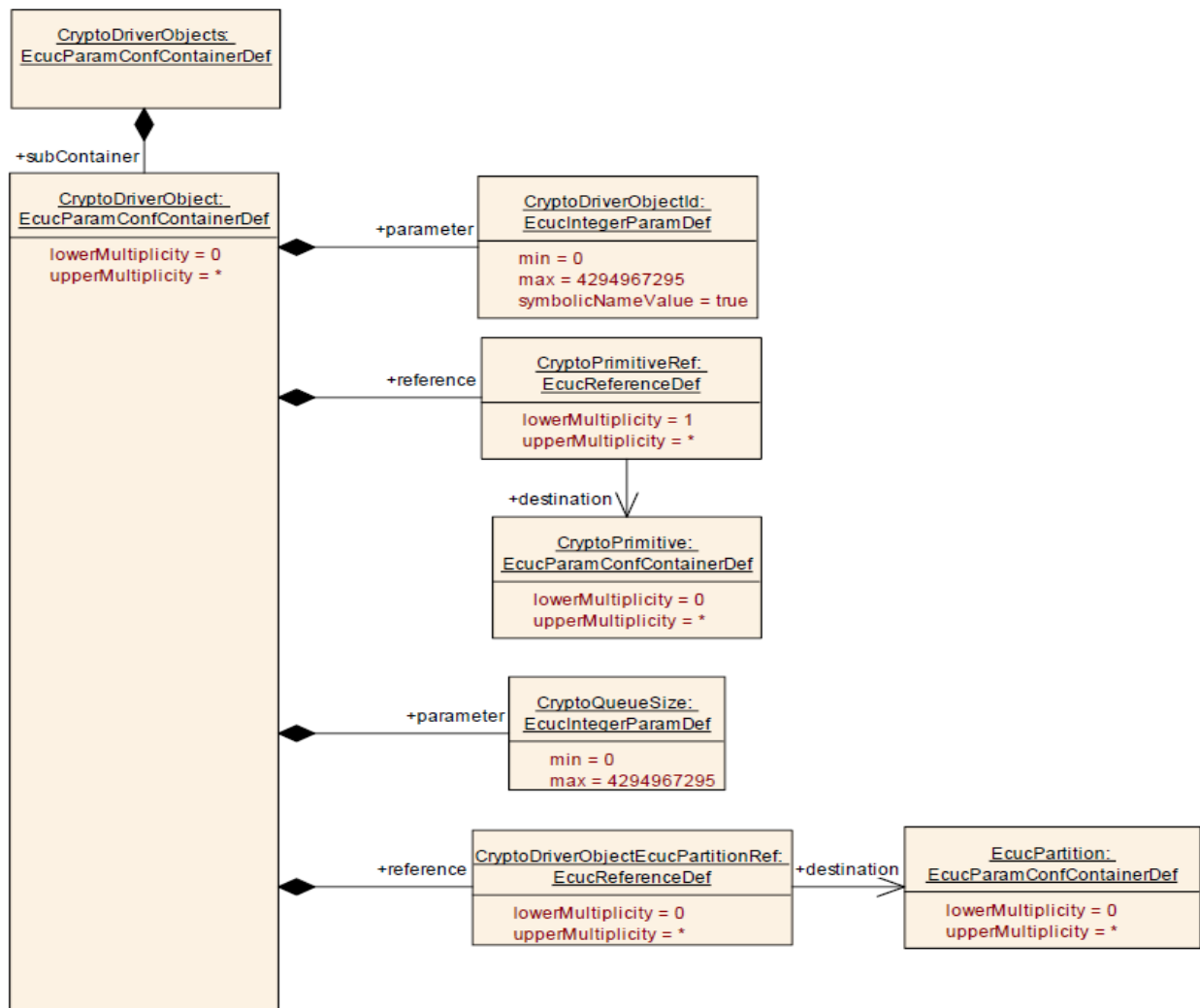


그림 12. AUTOSAR CryptoDriverObject Container

HaeModule Crypto Driver의 CryptoDriverObject Container 기본 설정 값은 아래 표와 같다. 사용자는 다음 항목에 대해서 사용 환경에 맞게 변경 가능하다.

- CryptoQueueSize
- CryptoDriverObjectEcucPartitionRef

Name	CryptoDriverObjectId	CryptoQueueSize	CryptoPrimitiveRef	CryptoDriverObjectEcucPartitionRef
HaeHsm	0	10	HaeHsm_CryptoPrimitives	NULL
HaeCryptoLib	1	10	HaeCryptoLib_CryptoPrimitives	NULL

표 7. HaeModule CryptoDriverObject Container 내용



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 8.3. CryptoPrimitives

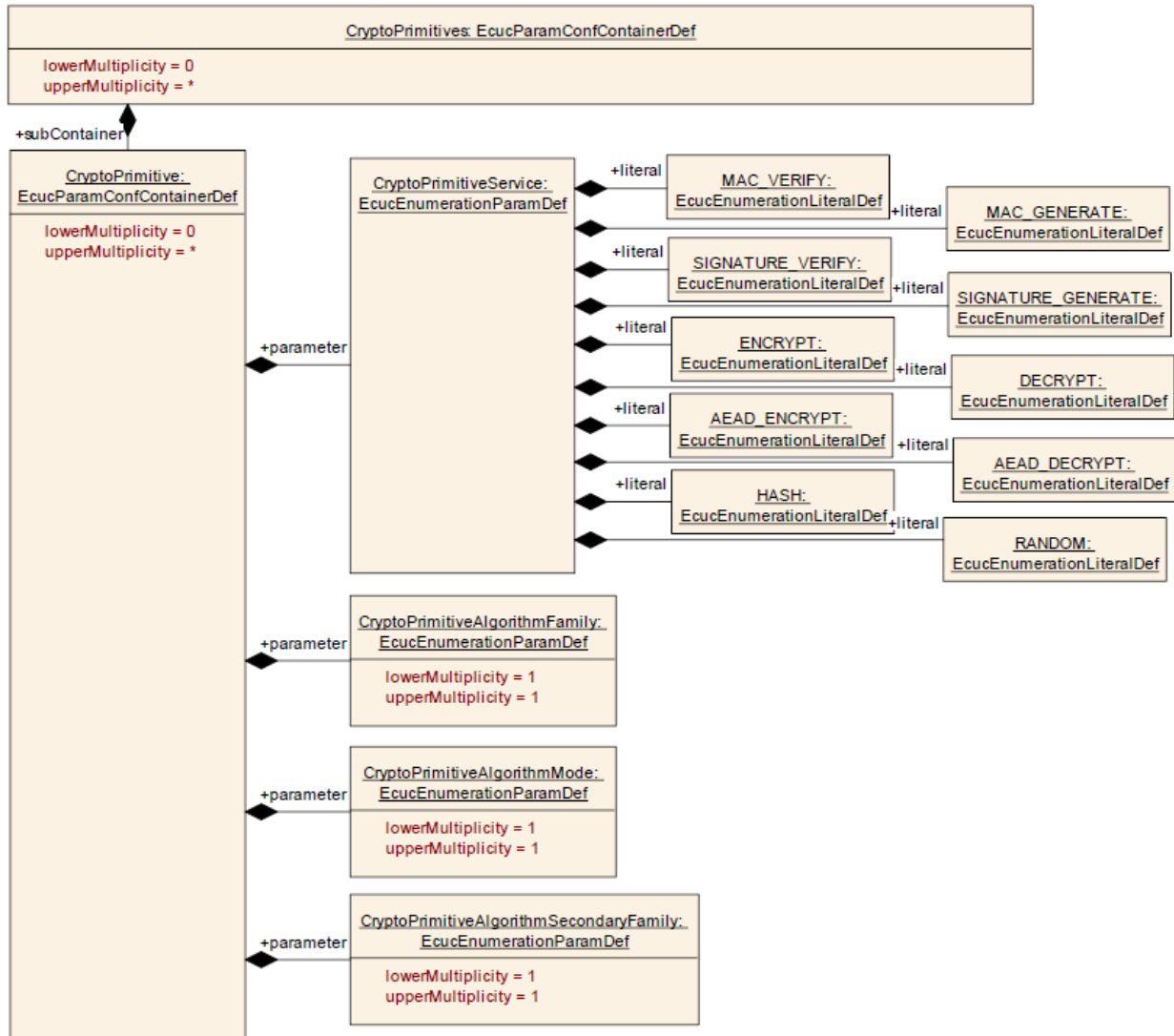


그림 13. AUTOSAR CryptoPrimitive Container

HaeModule Crypto Driver의 CryptoPrimitive Container는 HSM 모듈과 CryptoLib 모듈에서 제공하는 Crypto 알고리즘을 서비스한다. 자세한 서비스 별 파라미터 설정 방법은 본 매뉴얼의 “6.1. Crypto 알고리즘 별 Job Primitive 설정” 내용을 참고 바란다. 그리고 사용자가 필요에 의해 서비스를 추가할 경우 본 매뉴얼의 “6.2. 신규 사용자 Crypto Primitive 추가 방법” 내용 참고해서 추가가 가능하다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 8.4. CryptoKeys

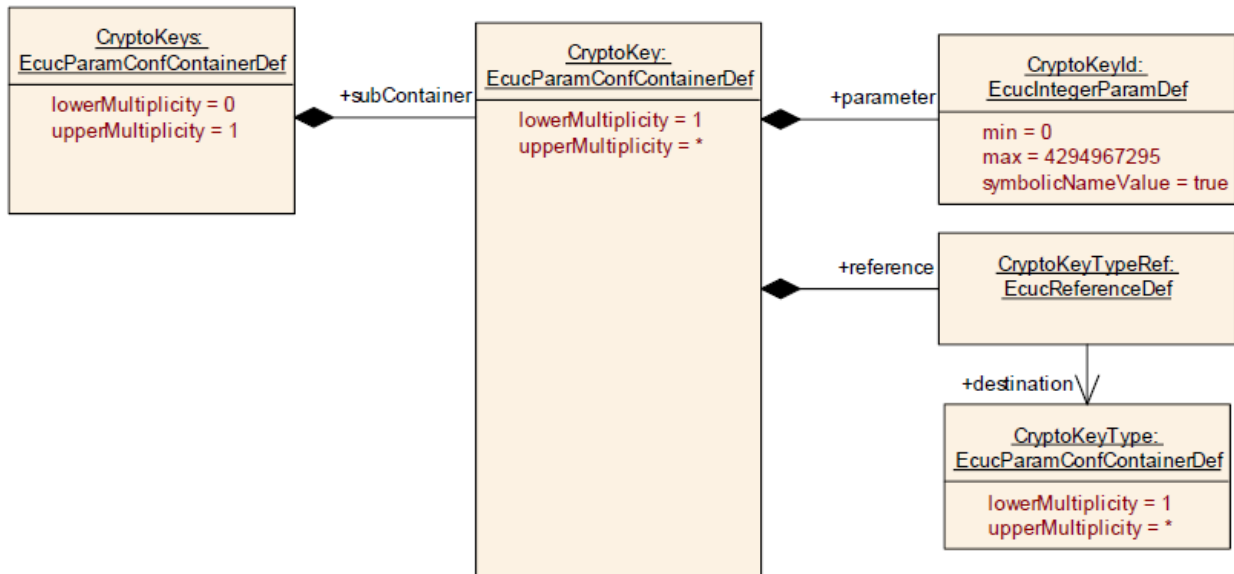


그림 14. AUTOSAR CryptoKey Container

HaeModule Crypto Driver의 CryptoKey Container는 HSM 내에 관리하는 Key와 연결할 수 있다. 이때, KeyId는 중복해서 설정하면 오동작한다. HSM Key외에 사용자 Key를 추가하고자 할 경우 본 매뉴얼의 “7.3. 신규 CryptoKey 추가 방법” 내용을 참고하길 바란다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 8.5. CryptoKeyTypes

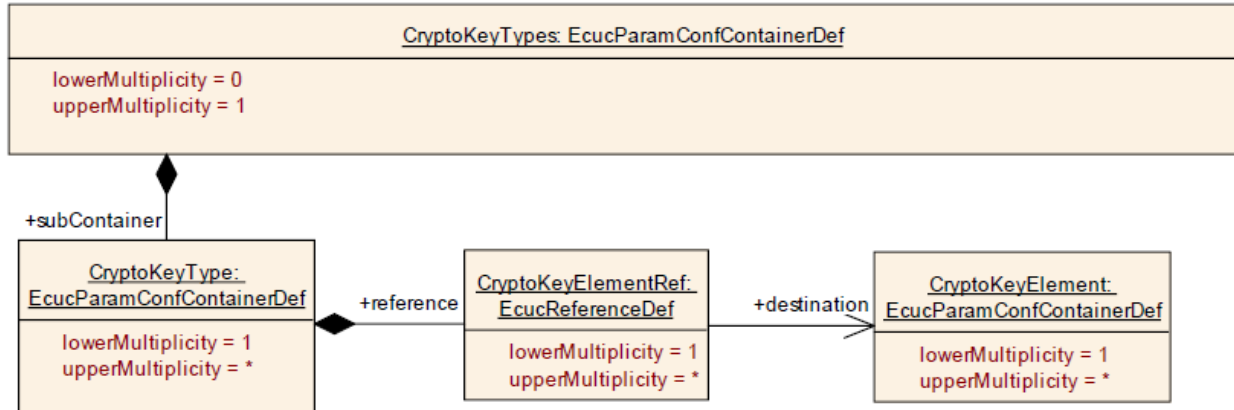
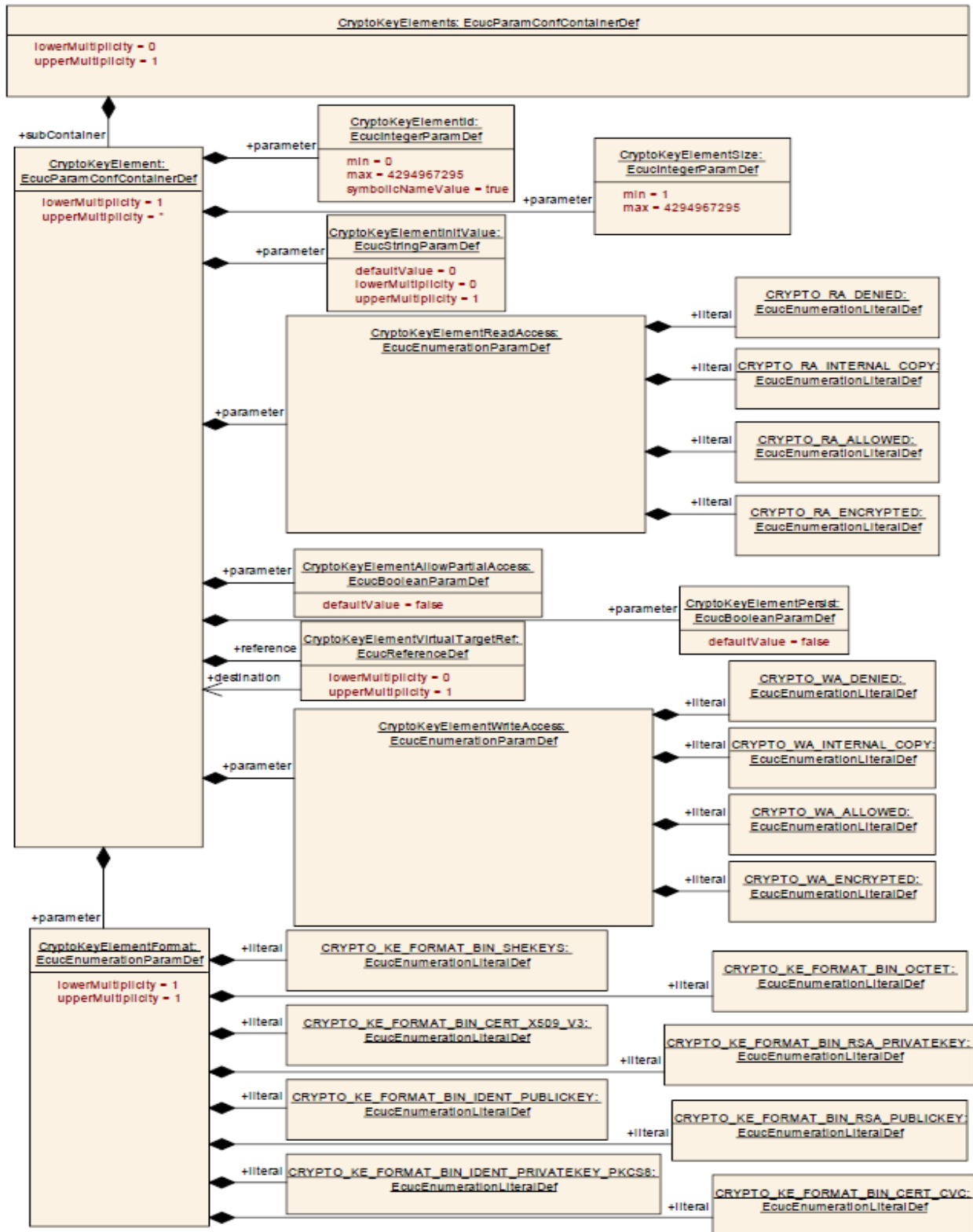


그림 15. AUTOSAR CryptoKeyType Container

HaeModule Crypto Driver의 CryptoKeyType Container는 사용을 위해 필요한 값들이 기본적으로 설정되어 있다. AES 키는 IV(Initialization Vector)가 필요하다. 이 IV Element는 HOST의 RAM상에 생성되고 관리된다. 예를 들어 CryptoKeyType\_AES\_PSK1는 두 개의 Element를 Reference하고 있다. 하나는 HSM에 저장되는 Element인 CryptoKeyElement\_AES\_Key\_PSK1와 HOST RAM상에 관리되는 Element인 CryptoKeyElement\_AES\_IV\_PSK1를 가진다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 8.6. CryptoKeyElements



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 그림 16. AUTOSAR CryptoKeyElement Container

HaeModule Crypto Driver의 CryptoKeyElement Container는 사용을 위해 필요한 값들이 기본적으로 설정되어 있다. HSM의 CryptoKeyElement에 대한 자세한 내용은 본 매뉴얼 “7.1.2. HSM Crypto Key Element 특성”을 참고 바란다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 9. Crypto Driver 포팅 가이드

### 9.1. mobilgene Classic 플랫폼

모빌진 클래식 환경에서의 포팅 가이드는 아래 사이트를 참고 바란다. 최신 가이드 내용이 업데이트 됨으로 자주 방문해서 확인하도록 한다.

- 플랫폼 버전: R4.x

[https://swpfaq.hyundai-autoever.com/display/MCF/%5BHAE\\_HSM%5D%5BR4X%5D%5BIM%5D+IM+of+Crypto\\_76\\_HaeHsm](https://swpfaq.hyundai-autoever.com/display/MCF/%5BHAE_HSM%5D%5BR4X%5D%5BIM%5D+IM+of+Crypto_76_HaeHsm)

- 플랫폼 버전: R4.4

[https://swpfaq.hyundai-autoever.com/display/MCF/%5BHAE\\_HSM%5D%5BR44%5D%5BIM%5D+IM+of+Crypto\\_76\\_HaeHsm](https://swpfaq.hyundai-autoever.com/display/MCF/%5BHAE_HSM%5D%5BR44%5D%5BIM%5D+IM+of+Crypto_76_HaeHsm)

### 9.2. AUTOSAR 플랫폼

각자의 플랫폼 매뉴얼과 AUTOSAR Crypto 관련 매뉴얼을 참고해서 설치하도록 한다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

### 9.3. 메모리 섹션

#### 9.3.1. Link Script 섹션

HaeModule Crypto Driver는 다음과 같은 메모리 섹션을 가진다. Link Script에 해당 섹션을 위치시킨다.

섹션은 다음 파일에 정의되어 있다.

Crypto\_76\_HaeModule\_R44/delivery/inc/ **Crypto\_76\_HaeModule\_MemMap.h**

구분	섹션명	메모리	설명
rodata	CRYPTO_76_HAEMODULE_ROM_CONST_UNSPECIFIED	Flash	Read Only 데이터
bss	CRYPTO_76_HAEMODULE_RAM_VAR_CLEARED_UNSPECIFIED	RAM	초기값이 없는 변수 데이터. 부팅 시 0x00 값으로 설정한다.
data	CRYPTO_76_HAEMODULE_RAM_VAR_INIT_UNSPECIFIED	RAM	초기값을 가지는 변수 데이터. 초기값이 Flash에 가지고 있다가 부 팅 시 RAM 섹션 주소에 복사된다.
text	CRYPTO_76_HAEMODULE_SEC_CODE	Flash	프로그램 코드

다음은 **Tasking 컴파일러**를 예를 들어 설명한다. 컴파일러 마다 Link Script는 다르며 섹션 명도 다소 다르게 생성될 수 있다. 정확한 내용은 컴파일러 매뉴얼을 참조한다. 그러나 대략적인 내용은 비슷하여 아래 설명을 참조하면 다른 컴파일 환경에서도 설정이 가능 할 것으로 생각된다.

#### rodata

기존 rodata 섹션과 같은 곳에 위치시키려면 Link Script에서 다음과 같은 섹션 명 중 하나가 존재하는지 확인한다.

- select ".rodata.\*\_ROM\_CONST\_\*";
- select ".rodata\*";

기존 섹션이나 새로운 섹션에 위치시키고자 한다면 원하는 섹션 위치에 다음과 같이 명기한다.

- select ".rodata.CRYPTO\_76\_HAEMODULE\_ROM\_CONST\_UNSPECIFIED";

#### bss

기존 bss 섹션과 같은 곳에 위치시키려면 Link Script에서 다음과 같은 섹션 명 중 하나가 존재하는지 확인한다.

- select ".bss.\*\_RAM\_VAR\_CLEARED\_\*";
- select ".bss\*";

기존 섹션이나 새로운 섹션에 위치시키고자 한다면 원하는 섹션 위치에 다음과 같이 명기한다.

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

- `select ".bss.CRYPTO_76_HAEMODULE_RAM_VAR_CLEARED_UNSPECIFIED";`

## data

기존 data 섹션과 같은 곳에 위치시키려면 Link Script에서 다음과 같은 섹션 명 중 하나가 존재하는지 확인한다.

- `select ".data.*_RAM_VAR_INIT_*";`
- `select ".data*";`

기존 섹션이나 새로운 섹션에 위치시키고자 한다면 원하는 섹션 위치에 다음과 같이 명기한다.

- `select ".data.CRYPTO_76_HAEMODULE_RAM_VAR_INIT_UNSPECIFIED";`

## text

기존 text 섹션과 같은 곳에 위치시키려면 Link Script에서 다음과 같은 섹션 명 중 하나가 존재하는지 확인한다.

- `select ".text.*_CODE";`
- `select ".text*";`

기존 섹션이나 새로운 섹션에 위치시키고자 한다면 원하는 섹션 위치에 다음과 같이 명기한다.

- `select ".text.CRYPTO_76_HAEMODULE_SEC_CODE";`

### 9.3.2. 플랫폼 메모리 맵 파일

모빌진 플랫폼의 경우 'MemMap.h' 파일이 존재하며 버전에 따라 'User\_MemMap.h' 파일이 별도 존재하는 경우가 있다.

#### User\_MemMap.h 파일이 존재하는 경우 (R4.4 버전)

파일에 다음과 같이 추가한다.

```
#if defined (SECTION_NOT_FOUND)
    #undef SECTION_NOT_FOUND
    #include "Crypto_76_HaeModule_MemMap.h"
#endif
```



	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## MemMap.h 파일만 존재하는 경우

파일 끝에 다음과 같이 추가한다.

```
#if defined (SECTION_NOT_FOUND)
    #undef SECTION_NOT_FOUND
    #include "Crypto_76_HaeModule_MemMap.h"
#endif
```

	문서번호	AUTOSAR HaeModule Crypto Driver User Manual	개정일자	'24. 04
	-		개정번호	1.03b

## 부록. 기술지원 가이드

기술지원은 Redmine을 통해서 받을 수 있다.

자세한 절차는 HSM 보안 모듈과 같이 제공되는 사용자 매뉴얼을 참고하기 바란다.