무선 응용계층 보안 프로토콜 규격

Wireless Application Layer Security Protocol Specification

v1.30

2003년 12월

안전한 IT 세상을 만드는

한국정보보호진흥원

목 차

무선 응용계층 보안 프로토콜 규격

Wireless Application Layer Security Protocol Specification

1. 개 요

본 규격은 전자서명인증체계에서 사용되는 무선 PKI 기반의 응용계층 소프트웨어간의 호환성 있는 보안프로토콜을 위해 전자서명 및 암호화 API를 정의하며 응용프로그램이 이들 함수를 이용하기 위한 필요한 요구사항과 절차를 명시한다.

2. 규격의 구성 및 범위

본 규격에서는 응용계층 보안프로토콜을 위해 전자서명 및 암호화 API를 정의하고 관련된 정보를 크게 두 부분으로 나누어 기술한다.

첫 번째로, 무선 단말기의 응용프로그램 및 컨텐츠 제공자의 서비스 사이트에서 전자서명 기능을 수행할 때 사용될 수 있는 signText 함수를 기술한다.

두 번째로, 무선 단말기의 응용프로그램 및 컨텐츠 제공자의 서비스 사이트에서 암호화 기능을 수행할 때 사용될 수 있는 encrypt 함수를 기술한다.

3. 관련 표준 및 규격

3.1 국외 표준 및 규격

[WMLSCrypto] WAP Forum Approved Version 20-June-2001, *WMLScript Crypto Library*, June 2001

[RFC1521] IETF RFC 1521, *MIME(Multipurpose Internet Mail Extensions), Part One : Mechanisms for Specifying and Describing the Format of Message Bodies*, September 1993

[RFC2279] IETF RFC 2279, *UTF-8, a transformation format of Unicode and ISO 10646*, January 1998

[RFC2630] IETF RFC 2630, *Cryptographic Message Syntax,* June 1999

[PKCS#1] RSA, PKCS#1 v2.0, *RSA Cryptography Standard*, October 1998

[PKCS#7] RSA Laboratories, PKCS#7 v1.5, *Cryptographic Message Syntax,* November 1993

[RFC2119] IETF, RFC2119, *Key words for use in RFCs to Indicate Requirement Levels,* March 1997

[DH-X9.42] ANSI, X9.42, *Public Key Cryptography for the Financial Services Industry : Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, March 2001

[SEC1] Standards for Efficient Cryptography, *SEC 1: Elliptic Curve Cryptography version 1.0, 2000*

3.2 국내 표준 및 규격

해당사항 없음

3.3 기타

해당사항 없음

4. 정의

4.1 전자서명법 용어 정의

본 가이드라인에서 사용된 다음의 용어들은 법률 제6585호 및 동법 시행령, 정보통신부고시 제2000-47호에 정의되어 있다.

가) 인증서
나) 공인인증서
다) 공인인증기관
라) 전자서명인증체계
마) 가입자
바) 가입자 설비(가입자 소프트웨어)

4.2 용어의 정의

해당사항 없음

4.3 용어의 효력

본 가이드라인에서 사용된 다음의 용어들은 공인인증기관 및 가입자 소프트웨어가 전자서명 알고리즘을 생성하거나 처리하는데 따라야 할 구현 정도를 의미하는 것으로 [RFC2119]를 준용하며 다음과 같은 의미를 지닌다.

가) 해야한다, 필수이다, 강제한다 (기호 : M)
반드시 준수해야 한다.
나) 권고한다 (기호 : R)
보안성 및 상호연동을 고려하여 준수할 것을 권장한다.
다) 할 수 있다, 쓸 수 있다 (기호 : O)
주어진 상황을 고려하여 필요한 경우에 한해 선택적으로 사용할 수 있다.
라) 권고하지 않는다 (기호 : NR)
보안성 및 상호연동을 고려하여 사용하지 말 것을 권장한다.
마) 금지한다, 허용하지 않는다 (기호 : X)
반드시 사용하지 않아야 한다.
바) 언급하지 않는다, 정의하지 않는다 (기호 : -)
준수 여부에 대해 기술하지 않는다.

5. 약어

본 가이드라인에서는 다음의 약어가 이용된다.

가) API : Application Programming Interface, 어플리케이션 프로그래밍 인터페이스
나) PKCS : Public-Key Cryptography Standards 공개키 암호 표준
다) WML : Wireless Markup Language, 무선 웹 표현 언어

6. 전자서명 함수

6.1. 개요

본 규격에서 정의하는 signText 함수는 무선 단말기와 컨텐츠 제공자간 통신을 통해 전달되는 데이터에 전자서명할 때 사용되는 것으로 [WMLSCrypto]·[RFC2630]·[PKCS#7]의 *SignedData*를 준용해야 한다. 무선 단말기와 컨텐츠 제공자는 RSA 및 ECDSA 전자서명 알고리즘 모두를 이용하여 전자서명을 수행할 수 있어야 한다.

```
signedString = Crypto.signText(
                                    StringToSign,
                                    options,
                                    keyIdType, keyId )
```

6.2 입력값

signText 함수의 입력은 다음 네 가지 파라미터로 구성되어야 한다.

가) s*tringToSign*

전자서명될 데이터로 일반 텍스트 문자열(String)로 표기되어야 한다.

나) o*ptions*

결과값인 *singedString*에 포함되는 선택적인 값으로 정수값(Integer)을 가져야 하며 다음에 기술되는 값들이 OR 연산을 통해 함께 사용될 수 있다.

정수값	설명
0x0001	INCLUDE_CONTENT · 결과값에 전자서명 대상 원문인 *StringToSign*을 포함한다.
0x0002	INCLUDE_KEY_HASH · 결과값에 전자서명생성키에 대응되는 전자서명검증키의 해쉬값을 포함한다.
0x0004	INCLUDE_CERTIFICATE · 결과 값에 전자서명 인증서 자체 또는 전자서명 인증서의 URL을 포함한다. · 만일 무선 단말기내에서 인증서를 획득할 수 없는 경우 결과값으로 “error:noCert"값을 반환해야 한다.

다) *keyIdType*

전자서명검증키의 키식별자 유형을 나타내는 정수값으로 값에 따라 네 번째 파라미터인 *keyId*의 내용이 결정된다.

정수값	설명
0	None ・*keyId*를 사용하지 않는 경우에 쓰인다.
1	USER_KEY_HASH ・가입자 전자서명검증키의 해쉬값이 사용될 경우 선택되는 값으로 실제 해쉬값은 네 번째 파라미터인 *keyId*를 통해 제공된다. ・해쉬알고리즘으로 반드시 SHA1을 사용해야 한다. ・무선 단말기 및 컨텐츠 제공자는 해쉬값에 대응되는 전자서명생성키를 찾아 전자서명을 수행해야 하며 적절한 전자서명생성키를 찾지 못하는 경우 “error:noCert" 값을 반환해야만 한다.
2	TRUSTED_KEY_HASH ・신뢰된 공인인증기관(들)의 전자서명검증키의 해쉬값이 사용될 경우 선택되는 값이며 실제 해쉬값은 네 번째 파라미터인 *KeyId*를 통해 제공된다. ・해쉬알고리즘으로 반드시 SHA1을 사용해야 한다. ・무선 단말기 및 컨텐츠 제공자는 해쉬값으로 표현된 공인인증기관 전자서명생성키로 인증된 가입자 전자서명생성키를 찾아 전자서명을 수행해야 하며 적절한 전자서명생성키를 찾지 못하는 경우 “error:noCert" 값을 반환해야만 한다.

라) *keyId*

세 번째 파라미터인 *keyIdType* 정수값에 따라 NULL값 또는 SHA-1값을 가져야 한다. *keyId*가 2인 경우 신뢰된 공인인증기관 전자서명검증키의 해쉬값이 여러 개 포함될 수 있다.

6.3 결과값

`signText` 함수의 결과값인 *SingedString*은 전자서명 수행결과 또는 예외상황에 따라 네 가지 유형의 결과값을 반환해야 한다.

가) 전자서명 수행 성공

전자서명이 성공적으로 수행된 경우 결과값은 [RFC1521]을 준용하여 Base64 인코딩으로 표현되는 *SignedContent* 문자열로 표시되어야 한다.

나) 전자서명 수행 실패

적절한 전자서명인증서나 전자서명생성키를 획득하지 못해 실패하는 경우 결과값은 “error:noCert"로 표현되는 문자열을 반환해야 한다.

다) 전자서명 수행 취소

가입자가 전자서명 수행을 도중에 취소하는 경우 결과값은 “error:userCancel"로 표현되는 문자열을 반환해야 한다.

라) 예외 상황

`signText` 함수에 입력되는 값들이 적절하게 표현된 값이 아니거나 내부오류 등으로 예기치 못한 오류상황 발생시 결과값으로 invalid값을 반환해야 한다.

7. 암호화 함수

7.1 개요

본 규격에서 정의하는 `encrypt` 함수는 무선 단말기와 컨텐츠 제공자간 통신을 통해 전달되는 데이터를 암호화하는데 사용되어야 하며 [RFC2630]・[PKCS#7]의 *EnvelopedData*와 호환성을 지닌다.

```
wapEnvelopedData = Crypto.encrypt(
                       flag,
                       dataToEncrypt,
                       recipientPublicKey,
                       KeyManagementAlgorithm,
                       contentEncryptionAlgorithm,   rid_type,   rid)
```

7.2 입력값

가) *flag*

암호화 대상 데이터를 나타내는 두 번째 파라미터인 *dataToEncrypt*를 어떻게 처리할 것인가를 나타내는 부가정보로 본 규격을 준용하는 무선 단말기 및 컨텐츠 제공자는 데이터에 한글이 포함될 경우 정수 1값을 사용해야 한다.

Flag	설 명
0	· 각 문자가 하나의 옥텟(octet)으로 변환가능한 경우에 한해 사용할 수 있다. · 만일 암호화 대상 데이터에 255 이상의 유니코드(Unicode)가 포함된 경우 “error:octetsExpected"를 반환해야 한다.
1	· 한글이 포함된 경우 적용되어야 하며 반드시 [RFC2279]를 준용하여 UTF-8으로 인코딩되어야 한다.
	· 상기 이외의 값은 본 규격에서 정의하지 않는다.

나) *dataToEncrypt*

암호화 대상 데이터를 문자열(String) 형태로 입력되어야 하며 첫 번째 파라미터인 flag와 올바르게 연관되어 사용되어야 한다.

다) *recipientPublicKey*

암호화에 사용되는 대칭키를 암호화하는데 사용되는 수신자의 공개키 유형을 표현하는 정수값으로 본 규격에서는 RSA · ECDSA · Diffie-Hellman 공개키의 사용을 정의한다.

정수값	설명
2	RSA
3	ECDSA
5	Diffie-Hellman

라) *KeyManagementAlgorithm = Integer*

*dataToEncrypt*의 암호화에 사용되는 대칭키를 안전하게 전달 · 관리하기 위하

여 사용되는 공개키 알고리즘을 표현하는 정수값으로 본 규격에서는 RSA · ECDSA · Diffie-Hellman 타입에 맞는 KeyManagementAlgorithm을 정의한다.

정수값	설명	공개키 타입
0	공개키 형태를 나타내는 recipientPublicKey에 따른 기본적인 알고리즘의 사용	
1	[DH-X9.42]에 따른 Ephemeral-Static Diffie-Hellman	Diffie-Hellman
2	[SEC1]에 따른 Elliptic curve ephemeral-static Diffie-Hellman	ECDSA
3	[PKCS#1]에 따른 RSA block type 2 encryption	RSA

마) *contentEncryptionAlgorithm*

*dataToEncrypt*를 암호화 하는데 사용되는 대칭키 알고리즘을 표현하는 정수값으로 본 규격을 준용하는 응용 소프트웨어는 3DES와 SEED를 지원해야 한다.

정수값	설명
1	Triple-DES CBC EDE [3DES]
2	RC5 CBC with 16 rounds & 128-bit block size
9	SEED-CBC with 16 rounds & 128-bit block size

바) *rid_type*

수신자 키식별자의 유형을 구분하기 위한 정수값으로 결과값인 *wapEnvelopedData*가 [RFC2630]의 *EncryptedData*로 변환되는 경우 본 입력값이 포함되어야 한다.

정수값	설명
1	수신자 공개키식별자(subjKeyId)
2	수신자 인증서의 발급자 인증서 일련번호(issuerSerNo)
255	본 규격에서는 정의하지 않음

사) *rid*

수신자의 키식별자 정보를 포함하는 정보로 *rid_type*에 따라 적절하게 처리되어야 하며 encrypt함수의 결과값인 *wapEnvelopedData*가 [RFC2630]의 *EncryptedData*로 변환되는 경우 본 입력값이 포함되어야 한다.

7.3 결과값

encrypt 함수의 결과값인 *wapEnvelopedData*는 암호화 수행결과 또는 예외상황에 따라 다섯 가지 유형의 결과값을 반환해야 한다.

가) 암호화 수행 성공

데이터 암호화가 성공적으로 수행된 경우 결과값은 [RFC1521]을 준용하여 Base64 인코딩으로 표현되는 *wapEnvelopedData* 문자열로 표시되어야 한다.

나) 부적절한 데이터 표현형식으로 인한 암호화 수행 실패

*flag*가 0이지만 *dataToEncrypt*가 단일 옥텟으로 표현할 수 없는 문자일 경우 결과값은 “error:octetsExpected"로 표현되는 문자열을 반환해야 한다.

다) 지원하지 못하는 공개키 유형으로 인한 암호화 수행 실패

*recipientPublicKey*가 가입자가 지원하지 못하는 공개키 유형일 경우 결과값은 “error:unsupportedPublicKey"로 표현되는 문자열을 반환해야 한다.

라) 지원하지 못하는 암호키관리 알고리즘으로 인한 암호화 수행 실패

*keyManagementAlgorithm*에 표시된 키관리 알고리즘을 가입자가 지원하지 못하는 경우 결과값은 “error:unsupportedKeyManagementAlgorithm"으로 표현되는 문자열을 반환해야 한다.

마) 예외 상황

encrypt 함수에 입력되는 값들이 적절하게 표현된 값이 아니거나 내부오류 등으로 예기치 못한 오류상황 발생시 결과값으로 invalid값을 반환해야 한다.

7.4 암호화 수행절차

무선단말기와 컨텐츠 제공자간 단대단 보안프로토콜을 사용하기 위해서 단말기는 먼저 컨텐츠제공자가 제공하는 encrypt 함수 관련 정보를 획득해야 한다. encrypt 함수 관련 정보는 공개키 유형 및 크기, 키관리 알고리즘, 대칭키 알고리즘 등으로 무선 단말기는 자신의 환경에 적절한 값들을 선택해야 한다. 다음으로 무선 단말기에서 encrypt 함수에 필요한 파라미터가 입력되어 생성된 데이터에 대한 암호화 결과값인 *wapEnvelopedData*를 컨텐츠 제공자에게 전달해야 한다.

본 규격을 준용하는 무선 단말기 및 컨텐츠 제공자는 [부록 1]에 기술된 보안 프로토콜 절차를 준용하여 단대단 보안이 이루어질 수 있도록 생성 또는 처리해야 한다.

부록 1. 단대단 데이터 암호화 절차

[표] 단대단 데이터 암호화 절차도

	무선단말기		컨텐츠 제공자
암호화 수행 사전단계			
1	컨텐츠 제공자의 서비스사이트접속	→	
2		←	ServerCertChain, keyManagementAlgorithm, contentEncryptionAlgorithm 등을 스크립트(WML등)에 포함시켜 Client에게 전송한다.
3	수신된 스크립트 페이지의 각 항목들을 검증한다. ・ServerCertChain ・keyManagementAlgorithm 지원여부 ・contentEncryptionAlgorithm 지원여부		
4	스크립트 페이지의 각 항목에 대한 확인이 끝나면 가입자는 스크립트 페이지에서 요구하는 값을 입력한다.		
5	입력된 값 및 나머지 변수들을 encrypt()함수에 입력한다.		
무선 단말기에서의 암호화 수행단계			
6	수신된 인증서에서 공개키를 추출하고 유형 및 길이 등의 지원여부를 확인한다.		
7	*flag*에의해 *dataToEncrypt*를 인코딩한다.		
8	*dataToEncrypt*를 암호화할 임의의 대칭키와 (필요하다면) 초기 벡터값을 만든다.		
9	대칭키로 (인코딩된) *dataToEncrypt*를 암호화한다		
10	암호화에 사용된 대칭키를 컨텐츠 제공자의 인증서에서 추출된 공개키로 암호화한다.		
11	암호화 결과값인 *wapEnvelope-dData*을 컨텐츠 제공자에게 전송한다.	→	

※ WapEnvelopedData를 이용하는 곳에서 CMS(Cryptography Message Syntax)로 변환하여 사용하는 것으로 정의

※ SEED의 keyID : SEED-CBC 16라운드 128비트, 9번으로 정의

부록 2. 규격 연혁

버전	제・개정일	제・개정내역
v1.00	2001년 7월	・"무선 응용계층 보안 Protocol 규격"으로 제정
v1.21	2001년 8월	・무선 응용계층 Protocol를 위한 Crypto Library 권고에서 무선 응용계층 보안 Protocol 규격으로 변경
v1.30	2003년 12월	・KeyID 생성방안 명시 (SHA-1) ・KeyManagementAlgorithm에 공개키 타입 추가 ・관련 국외 표준 및 규격 반영 ・단대단 데이터 암호화 절차도 수정(부록 1)

규격 작성 공헌자

본 규격의 개정을 위해 아래와 같이 많은 분들이 공헌을 하였습니다.

구분	성명	소속사
규격 제안	암호인증기술팀	한국정보보호진흥원
규격 초안 제출	암호인증기술팀	한국정보보호진흥원
규격 검토	임선간	한국정보보호진흥원
	박종욱	한국정보보호진흥원
	전인경	한국정보보호진흥원
	박상환	한국정보보호진흥원
	박배효	한국정보보호진흥원
	최태규	한국정보보호진흥원
	심희원	한국정보인증
	장재환	한국정보인증
	이성진	한국증권전산
	장석한	한국증권전산
	이성국	한국증권전산
	오중효	금융결제원
	이한욱	금융결제원
	양승모	한국전자인증
	이성철	한국무역정보통신
	이제호	한국무역정보통신
	국상진	한국무역정보통신
규격안 편집	박배효	한국정보보호진흥원