

Translated English of Chinese Standard: GM/T0020-2012

www.ChineseStandard.net → Buy True-PDF → Auto-delivery.

Sales@ChineseStandard.net

GM

PASSWORD INDUSTRY STANDARD OF THE PEOPLE'S REPUBLIC OF CHINA

ICS 35.040

L 80

File No.: 38318-2013

GM/T 0020-2012

Certificate application integrated service interface specification

证书应用综合服务接口规范

GM/T 0020-2012 -- How to BUY & immediately GET a full-copy of this standard?

1. www.ChineseStandard.net;
2. Search --> Add to Cart --> Checkout (3-steps);
3. No action is required - Full-copy of this standard will be automatically & immediately delivered to your EMAIL address in 0~60 minutes.
4. Support: Sales@ChineseStandard.net. Wayne, Sales manager

Issued on: November 22, 2012

Implemented on: November 22, 2012

Issued by: State Cryptography Administration

Table of Contents

Foreword.....	3
Introduction	4
1 Scope	5
2 Normative references	5
3 Terms and definitions	5
4 Abbreviation	6
5 Algorithm identifier and data structure	6
6 Overview of certificate application integrated service interface	7
7 Definition of function of certificate application integrated service interface	8
Annex A (normative) Error code definition of integrated service interface of certificate application	35
Annex B (informative) Typical deployment model of integrated service interface of certificate application	38
Annex C (informative) Integrated example of integrated service interface of certificate application	40
Bibliography	43

Foreword

This Standard was drafted in accordance with the rules given in GB/T 1.1-2009.

Attention is drawn to the possibility that some of the elements of this Standard may be the subject of patent rights. The issuing authority shall not be held responsible for identifying any or all such patent rights.

This Standard was proposed by and shall be under the jurisdiction of State Cryptography Administration.

Annex A of this Standard is normative. Annex B and Annex C are informative.

The drafting organizations of this Standard: Beijing Digital Certification Co., Ltd., Shanghai Geer Software Co., Ltd., Beijing Haitai radius Technology Co., Ltd., Shanghai Digital Certificate Certification Center Co., Ltd., Wuxi Jiangnan Information Security Engineering Technology Center, Chengdu Wei Shi Tong Information Industry Co., Ltd., Changchun Ji Tai Yuan Information Technology Co., Ltd., Xing Tang Communication Technology Co., Ltd., Shandong De'an Information Technology Co., Ltd., National Information Security Engineering Technology Research Center, National Cryptography Authority Commercial Password Detection Center.

The drafters of this Standard: Liu Ping, Li Shusheng, Tan Wuzheng, Liu Zengshou, Liu Cheng, Xu Qiang, Li Yuanzheng, Zhao Lili, Wang Nina, Kong Fanyu, Yuan Feng, Li Zhiwei.

Any content related to cryptographic algorithm in this Standard shall be in accordance with the relevant national laws and regulations.

Certificate application integrated service interface specification

1 Scope

This Standard specifies a unified service interface for certificate application.

This Standard is applicable to the development of cryptographic application service products under public key cryptographic application technology system, to the research and testing of cryptographic application support platform. It can also be used to guide the direct use of cryptographic device and the integration and development of application system of cryptographic service.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

GM/T 0006, *Cryptographic Application Identifier Criterion Specification*

GM/T 0009, *SM2 Cryptography Algorithm Application Specification*

GM/T 0010, *SM2 Cryptography Message Syntax Specification*

GM/T 0015, *Digital Certificate Format Based on SM2 Algorithm*

GM/T 0019, *Universal Cryptography Service Interface Specification*

PKCS #7, *Cryptographic Message Syntax*

RFC3275, *(Extensible Markup Language) XML-Signature Syntax and Processing*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1 digital certificate

a digital document of certification authority digital signature containing public

Data type A: when the public key algorithm is RSA, the structure of the data shall follow PKCS #1; when the public key algorithm is SM2, the structure of the data shall follow GM/T 0009.

Data type B: when the public key algorithm is RSA, the structure of the data shall follow PKCS #7; when the public key algorithm is SM2, the structure of the data shall follow GM/T 0010.

6 Overview of certificate application integrated service interface

6.1 Overview

The certificate application integrated service interface is located between the application system and the typical cryptographic service interface. It directly provides the certificate information analysis, the confidentiality, integrity, non-repudiation and other advanced cryptographic services based on digital certificate identity and information to the application layer. The interface can be directly used for system calls, turning the application's cryptographic service request to the common cryptographic service interface, which calls corresponding cryptographic device, through the cryptographic service interface, to realize specific cryptographic operation and key operation. The common cryptographic service interface shall follow GM/T 0019.

The certificate application integrated service interface specified in this Specification includes two types: client service interface and server service interface. The server service interface uses descriptions of COM component form and Java form. The digital certificate format involved in this document shall follow GM/T 0015.

6.2 Client service interface

The client service interface defined in this Specification uses client control method. The client control is applicable to client program calls. The interface forms include DLL dynamic library, ActiveX control, Applet plugin, etc. The interface shall support the mainstream operating systems used by Windows XP, Windows 2000, Windows 2003, Vista, Windows 7.

The main functions of the client control interface shall include configuration management, certificate resolution, signature and authentication, encryption and decryption, digital envelop, XML data signature and authentication.

When defining the client service interface, this Specification takes ActiveX control as an example for description, of which BSTR represents the function return value or parameter type is OLECHAR string type. Different development

- m) obtain certificate extension information: SOf_GetCertInfoByOid
- n) obtain device information: SOf_GetDeviceInfo
- o) validate certificate validity: SOf_ValidateCert
- p) digital signature: SOf_SignData
- q) validate signature: SOf_VerifySignedData
- r) file signature: SOf_SignFile
- s) validate file signature: SOf_VerifySignedFile
- t) encrypt data: SOf_EncryptData
- u) decrypt data: SOf_DecryptData
- v) file encryption: SOf_EncryptFile
- w) file decryption: SOf_DecryptFile
- x) message signature: SOf_SignMessage
- y) validate message signature: SOf_VerifySignedMessage
- z) parse message signature: SOf_GetInfoFromSignedMessage
- aa) XML digital signature: SOf_SignDataXML
- bb) validate XML digital signature: SOf_VerifySignedDataXML
- cc) parse XML signature data: SOf_GetXMLSignatureInfo
- dd) generate random number: SOf_GenRandom
- ee) obtain latest error code: SOf_GetLastError()

Take ActiveX control form as an example to define the interface function.

7.1.2 Obtain interface version number: SOf_GetVersion

Prototype:	BSTR SOf_GetVersion()	
Description:	Obtaining the version number of the control	
Parameter:	Null	
Return value:	Not void	Successful
	Void	Failed

7.1.3 Set signature algorithm: SOf_SetSignMethod

Prototype:	long SOf_SetSignMethod (long SignMethod)
------------	--

The definitions of COM component interface functions are as follows:

- a) Set certificate trust list: SOF_SetCertTrustList
- b) Inquire alternative name of certificate trust list:
SOF_GetCertTrustListAltNames
- c) Inquire certificate trust list: SOF_GetCertTrustList
- d) Delete certificate trust list: SOF_DelCertTrustList
- e) Initialize application policy: SOF_InitCertAppPolicy
- f) Set signature algorithm: SOF_SetSignMethod
- g) Obtain current signature algorithm: SOF_GetSignMethod
- h) Set encryption algorithm: SOF_SetEncryptMethod
- i) Obtain encryption algorithm: SOF_GetEncryptMethod
- j) Obtain server certificate: SOF_GetServerCertificate
- k) Generate random number: SOF_GenRandom
- l) Obtain certificate information: SOF_GetCertInfo
- m) Obtain certificate extension information: SOF_GetCertInforByOid
- n) Validate certificate validity: SOF_ValidateCert
- o) Digital signature: SOF_SignData
- p) Validate signature: SOF_VerifySignedData
- q) File signature: SOF_SignFile
- r) Validate file signature: SOF_VerifySignedFile
- s) Encrypt data: SOF_EncryptData
- t) Decrypt data: SOF_DecryptData
- u) File encryption: SOF_EncryptFile
- v) File decryption: SOF_DecryptFile
- w) Message signature: SOF_signMessage
- x) Validate message signature: SOF_VerifySignedMessage

- y) Message signature without original text: SOF_SignMessageDetach
- z) Validate message signature without original text:
SOF_VerifySignedMessageDetach
- aa) Parse message signature: SOF_GetInfoFromSignedMessage
- bb) XML digital signature: SOF_SignDataXML
- cc) Validate XML digital signature: SOF_VerifySignedDataXML
- dd) Parse XML signature data: SOF_GetXMLSignatureInfo
- ee) Create timestamp request: SOF_CreateTimeStampRequest
- ff) Create timestamp response: SOF_CreateTimeStampResponse
- gg) Validate timestamp: SOF_VerifyTimeStamp
- hh) Parse timestamp: SOF_GetTimeStampInfo
- ii) Obtain latest error code: SOF_GetLastError

7.2.2 Set certificate trust list: SOF_SetCertTrustList

Prototype:	long SOF_SetCertTrustList(BSTR CTLAltName, BSTR CTLContent, short CTLContentLen)	
Description:	Set certificate trust list	
Parameter:	CTLAltName[in]	Alternative name of certificate trust list
	CTLContent[in]	base64 encoded format certificate trust list content
	CTLContentLen[in]	Certificate trust list length
Return value:	0	Successful
	Other	Failed, see error code

7.2.3 Inquire alternative name of certificate trust list: SOF_GetCertTrustListAltNames

Prototype:	BSTR SOF_GetCertTrustListAltNames()	
Description:	Inquire alternative name of certificate trust list.	
Parameter:	Null	
Return value:	Not void	A string combination of trust list alternative name, such as "CA001@CA002@CA003"
	Void	Failed

7.2.4 Inquire certificate trust list: SOF_GetCertTrustList

Prototype:	BSTR SOF_GetCertTrustList(BSTR CTLAltName)	
Description:	Inquire certificate trust list according to alternative name.	
Parameter:	CTLAltName[in]	Alternative name of certificate trust list
Return value:	Not void	base64 encoded certificate trust list

- e) Obtain living example of designated application: SOF_getInstance(java.lang.String PolicyName)
- f) Set signature algorithm: SOF_setSignMethod
- g) Obtain current signature algorithm: SOF_getSignMethod
- h) Set encryption algorithm: SOF_setEncryptMethod
- i) Obtain encryption algorithm: SOF_getEncryptMethod
- j) Obtain server certificate: SOF_getServerCertificate
- k) Obtain server certificate of designated key usage: SOF_getServerCertificateByUsage
- l) Generate random number: SOF_genRandom
- m) Obtain certificate information: SOF_getCertInfo
- n) Obtain certificate extension information: SOF_getCertInfoByOid
- o) Validate certificate validity: SOF_validateCert
- p) Digital signature: SOF_signData
- q) Validate signature: SOF_verifySignedData
- r) File signature: SOF_signFile
- s) Validate file signature: SOF_verifySignedFile
- t) Encrypt data: SOF_encryptData
- u) Decrypt data: SOF_decryptData
- v) File encryption: SOF_encryptFile
- w) File decryption: SOF_decryptFile
- x) Message signature: SOF_signMessage
- y) Validate message signature: SOF_verifySignedMessage
- z) Parse message, signature: SOF_getInfoFromSignedMessage
- aa) Message signature without original text: SOF_signMessageDetach
- bb) Validate message signature without original text: SOF_verifySignedMessageDetach

Return value:	Not void	base64 encoded format certificate trust list
		returned
	Void	Failed

7.3.5 Delete certificate trust list: SOF_delCertTrustList

Prototype:	public boolean SOF_delCertTrustList (java. lang. String ctlAltName)	
Description:	Delete certificate trust list according to alternative name.	
Parameter:	ctlAltName	Alternative name of certificate trust list
Return value:	true	Successful
	false	Failed

7.3.6 Obtain living example of designated application: SOF_getInstance(java. lang. String PolicyName)

Prototype:	public static java. lang. Object SOF_getInstance(java. lang. String PolicyName)	
Description:	Initialize interface. Obtain living example through application's alternative name. The application's alternative name is related to the configured certificate, key, trust certificate chain, algorithm type, CRL as well as certificate validation policy. If user has multi-application request, it shall obtain more than one example objects at the same time to meet different use request. Different living example shall have different effect when calling method (for example, signature of different key, encryption of different algorithm, different certificate validation policy).	
Parameter:	PolicyName	Application policy name
Return value:	Not void	Corresponding living example of this application policy name returned
	Void	Failed

7.3.7 Set signature algorithm: SOF_setSignMethod

Prototype:	void SOF_setSignMethod(long signMethod)	
Description:	Set signature algorithm used by Java component signature operation.	
Parameter:	signMethod	Signature algorithm identifier (see GM/T 0006)
Return value:	Null	

7.3.8 Obtain current signature algorithm: SOF_getSignMethod

Prototype:	java. lang. long SOF_getSignMethod()	
Description:	Obtain signature algorithm used by component signature operation.	
Parameter:	Null	
Return value:	Not 0	Current signature algorithm identifier (see GM/T 0006)
	0	Not setting algorithm

7.3.9 Set encryption algorithm: SOF_setEncryptMethod

Prototype:	void SOF_setEncryptMethod(long encryptMethod)	
------------	---	--

	inFile	Plaintext file path to be encrypted
	outFile	Storage path of ciphertext file
Return value:	true	Successful
	false	Failed

7.3.24 File decryption: SOF_decryptFile

Prototype:	boolean SOF_decryptFile(java. lang. String ContainerName, java. lang. String inFile, java. lang. String outFile)	
Description:	Decrypt ciphertext file.	
Parameter:	ContainerName	Corresponding certificate unique identifier of decryption key
	inFile	Ciphertext file path to be decrypted, ciphertext is data type B.
	outFile	Storage path of plaintext file
Return value:	true	Successful
	false	Failed

7.3.25 Message signature: SOF_signMessage

Prototype:	java. lang. String SOF_SignMessage(byte[] inData)	
Description:	Digitally sign string data, signature format is data type B with original text.	
Parameter:	inData	Data original text to be signed
Return value:	Not void	base64 encoded signature value
	Void	Failed

7.3.26 Validate message signature: SOF_verifySignedMessage

Prototype:	boolean SOF_verifySignedMessage(java. lang. String SignedMessage)	
Description:	Validate digital signature, signature format is data type B with original text.	
Parameter:	SignedMessage	Base64 encoded message signature packet
Return value:	true	Successful
	false	Failed

7.3.27 Parse message, signature: SOF_getInfoFromSignedMessage

Prototype:	byte[] SOF_getInfoFromSignedMessage (java. lang. String SignedMessage, short type)	
Description:	Parse information of signature packet of data type B. Original text, signature value, signature certificate and other information shall be obtained.	
Parameter:	SignedMessage	Signature packet
	Type	type definition, 1: original text; 2: signer certificate; 3: signature value
Return value:	Not void	Corresponding value of type returned
	Void	Failed

7.3.28 Message signature without original text: SOF_signMessageDetach

Bibliography

- [1] GB/T 19713-2005, *Information technology - Security techniques - Public key infrastructure - Online certificate status protocol*
- [2] *RSA Security: Public-Key Cryptography Standards (PKCS)*
- [3] *PKCS #11, Cryptographic Token Interface Standard*
- [4] IETF RFC2459, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*
- [5] IETF RFC2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol*
- [6] IETF RFC1777, *Lightweight Directory Access Protocol*
- [7] IETF RFC2587, *Internet X.509 Public Key Infrastructure LDAPv2 Schema*
- [8] IETF RFC3647, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*
- [9] ISO/IEC 8825-1:1998, *Information technology -- ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

_____ **END** _____

This is an excerpt of the PDF (Some pages are marked off intentionally)

Full-copy PDF can be purchased from 1 of 2 websites:

1. <https://www.ChineseStandard.us>

- SEARCH the standard ID, such as GB 4943.1-2022.
- Select your country (currency), for example: USA (USD); Germany (Euro).
- Full-copy of PDF (text-editable, true-PDF) can be downloaded in 9 seconds.
- Tax invoice can be downloaded in 9 seconds.
- Receiving emails in 9 seconds (with download links).

2. <https://www.ChineseStandard.net>

- SEARCH the standard ID, such as GB 4943.1-2022.
- Add to cart. Only accept USD (other currencies - <https://www.ChineseStandard.us>).
- Full-copy of PDF (text-editable, true-PDF) can be downloaded in 9 seconds.
- Receiving emails in 9 seconds (with PDFs attached, invoice and download links).

Translated by: Field Test Asia Pte. Ltd. (Incorporated & taxed in Singapore. Tax ID: 201302277C)

About Us (Goodwill, Policies, Fair Trading...): <https://www.chinesestandard.net/AboutUs.aspx>

Contact: Wayne Zheng, Sales@ChineseStandard.net

Linkin: <https://www.linkedin.com/in/waynezhengwenrui/>

----- The End -----