ICS 35.030
CCS L 80

GM

中华人民共和国密码行业标准

GM/T 0131—2023

电子签章应用接口规范

Electronic seal signature application interface specification

2023-12-04 发布　　2024-06-01 实施

国家密码管理局　　发布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：北京国脉信安科技有限公司、北京数字认证股份有限公司、上海市数字证书认证中心有限公司、数字广东网络建设有限公司、北京点聚信息技术有限公司、方正国际软件（北京）有限公司、三未信安科技股份有限公司、北京信安世纪科技有限公司、航天网安技术（深圳）有限公司。

本文件主要起草人：袁峰、郑志梅、张立圆、封维端、李中声、张永强、刘岩、赵子轩、赵剑竹、李腾跃、李祖金、汪杰、刘梦吟、陆猛、杨玉坤、李婧、高志权、王腾飞、郝松。

电子签章应用接口规范

1 范围

本文件规定了电子签章系统对外提供的服务接口，为电子签章系统与应用系统之间提供统一的数据交互格式和使用接口，包括服务调用和组件调用两种电子印章使用接口形式。

本文件适用于电子签章系统的研制、使用和检测，以及基于电子签章系统的应用开发。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 20518 信息安全技术 公钥基础设施 数字证书格式

GB/T 20520 信息安全技术 公钥基础设施 时间戳规范

GB/T 32010.1 文献管理 可移植文档格式 第1部分：PDF1.7

GB/T 32905 信息安全技术 SM3密码杂凑算法

GB/T 33190 电子文件存储与交换格式 版式文档

GB/T 33560 信息安全技术 密码应用标识规范

GB/T 35275—2017 信息安全技术 SM2密码算法加密签名消息语法规范

GB/T 35291 信息安全技术 智能密码钥匙应用接口规范

GB/T 36322 信息安全技术 密码设备应用接口规范

GB/T 38540—2020 信息安全技术 安全电子签章密码技术规范

GM/T 0031—2014 安全电子签章密码技术规范

GM/T 0094—2020 公钥密码应用技术体系框架规范

GM/T 0099 开放式版式文档密码应用技术规范

GM/T 0112 PDF格式文档的密码应用技术要求

GM/Z 4001 密码术语

RFC 4648 Base16、Base32、Base64数据编码格式(Base16, Base32, and Base64 Data Encodings)

3 术语和定义

GM/Z 4001界定的以及下列术语和定义适用于本文件。

3.1

电子印章 electronic seal

一种由电子印章制作者数字签名的安全数据。

注：包括电子印章所有者信息和图形化内容的数据，用于安全签署数据电文。

[来源：GB/T 38540—2020，3.1]

3.2

电子签章 electronic seal signature

使用电子印章签署数据电文的过程。

注：电子签章可实现与纸质文件签章操作相似的可视效果，可保障数据来源的真实性、数据完整性以及签名人行为的不可否认性。

[来源：GB/T 38540—2020，3.2]

3.3

签章服务　electronic seal signature service

提供支持 OFD 和 PDF 等格式的数字电文的电子印章签署服务。

3.4

电子签章系统　electronic seal signature system

对数据电文提供电子签章、数字签名、签名验证等服务的软件系统。

3.5

分散式签章　decentralized electronic seal signature

在电子印章所有者掌控的密码模块中实现电子印章私钥的存储及其计算。

注：私钥计算包括协同签名方法。

3.6

集中式签章　centralized electronic seal signature

电子印章私钥统一存储在指定的密码设备中，该密码设备由电子签章系统掌控。电子印章使用者通过电子签章系统实现对电子印章的使用。

3.7

用户印章　user seal

用户在电子签章系统里被授权的电子印章。

3.8

用户证书　user certificate

用户在电子签章系统里被授权的数字证书。

3.9

B/S 模式　browser/server mode

客户端统一为 WEB 浏览器，系统功能实现的核心部分集中到服务器上的一种应用系统模式。

4　缩略语

下列缩略语适用于本文件。

DER：可辨别编码规则(Distinguished Encoding Rules)

HTTP：超文本传输协议(Hyper Text Transfer Protocol)

JSON：JavaScript 对象标记(JavaScript Object Notation)

OFD：开放式版式文档(Open Fixedlayout Document)

OID：对象标识符(Object ID)

PDF：可移植文档格式(Portable Document Format)

PKI：公钥基础设施(Public Key Infrastructure)

SDK：软件开发工具包(Software Development Kit)

5　电子签章应用接口概述

5.1　通用要求

本文件在 GM/T 0094—2020 中第 4 章的公钥密码应用技术体系框架中的位置见图 1。

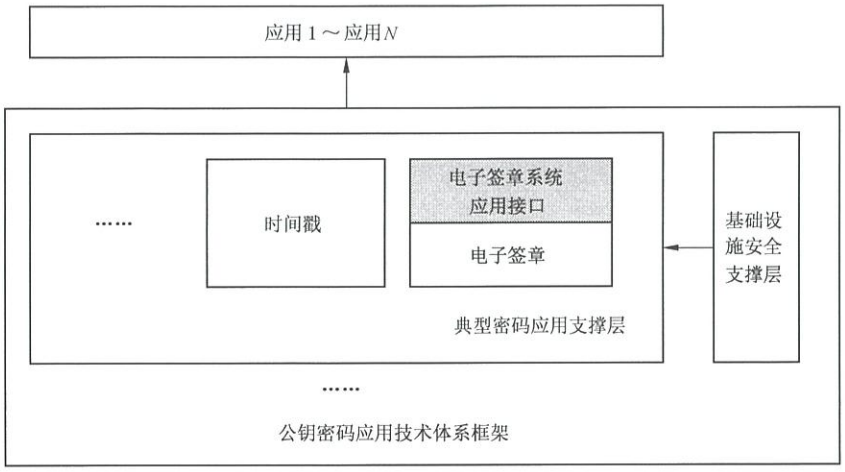

图1　电子签章应用接口在公钥密码应用技术体系框架中的位置

电子签章应用接口包括服务调用接口和组件调用接口两种接口形式。服务调用可采用HTTP或HTTPS服务。组件调用可采用部署在用户端的DLL/SO等SDK组件。这两种接口形式分别包含基础应用接口和综合应用接口，支持分散式签章和集中式签章两种模式。电子签章应用接口列表如表1所示。

表1　电子签章应用接口列表

分类	序号	接口名	功能
基础接口	1	SEF_GetProviderInfo	获取签章服务信息
	2	SEF_GetUserSealInfoList	获取用户印章信息列表
	3	SEF_QuerySealsByNameOrID	根据印章名称/ID查询印章
	4	SEF_GetUserCertInfoList	获取用户证书信息列表
	5	SEF_GetCertByCertSN	根据证书序列号查询证书
	6	SEF_GetAlgOID	获取签名和杂凑算法标识
	7	SEF_PackSignedValue	封装签章结构
	8	SEF_AddTimestampToSignedValue	添加时间戳到签章结构体
	9	SEF_PackSignedMessage	封装消息签名结构
	10	SEF_VerifySignedData	验证签章/消息签名数据
	11	SEF_GetSignedValueInfo	解析签章数据
	12	SEF_GetSealInfo	解析印章数据
	13	SEF_GetErrMessage	获取错误信息
综合接口	1	SEF_SealSignByPosition	文件定位签章
	2	SEF_SealSignOnKeywords	文件关键字签章
	3	SEF_SealSignOnCrosspages	文件加盖骑缝章
	4	SEF_CertSign	文件数字签名
	5	SEF_AddAnnot	文件添加文字或图片注释

表 1　电子签章应用接口列表（续）

分类	序号	接口名	功能
综合接口	6	SEF_VerifySealSignedFile	文件签章验证
	7	SEF_VerifyCertSignedFile	文件签名验证
	8	SEF_GetSignCount	获取文件签章/签名个数
	9	SEF_VerifySignByIndex	文件指定签章/签名验证
	10	SEF_FileToImage	文件转图像
	11	SEF_VerifySignedFile	文件签章/签名验证，验证所有签章和签名

本文件所涉及的 OFD 文件应符合 GB/T 33190 和 GM/T 0099 的规定，PDF 文件应符合 GB/T 32010.1 和 GM/T 0112 的规定。

电子签章的签名算法应与电子印章的签名算法一致，如果签名算法是 SM2，则电子印章结构、电子签章结构应符合 GB/T 38540 或 GM/T 0031 的规定，签名值应符合 GB/T 35275 的规定，电子印章和电子签章数据结构的数字证书应符合 GB/T 20518 的规定。当杂凑算法使用 SM3 时，杂凑值应符合 GB/T 32905 的规定。

所有涉及电子印章私钥操作的接口应实施私钥验证安全机制。智能密码钥匙私钥操作应符合 GB/T 35291 的规定，密码设备私钥操作应符合 GB/T 36322 的规定。服务端获取用户印章信息列表和服务端获取用户证书信息列表的接口应进行用户身份鉴别。用户身份鉴别流程可参考文献[1][2][3]。

自定义 JSON 报文数据结构应符合附录 A 的规定，自定义 C 语言数据结构应符合附录 B 的规定，典型服务调用接口示例见附录 C，典型组件调用接口示例见附录 D，错误代码定义应符合附录 E 的规定。

智能密码钥匙操作应符合 GB/T 35291 的规定，密码设备操作应符合 GB/T 36322 的规定。

5.2　服务调用接口

服务调用接口为请求/响应服务的形式，可用于 B/S 模式的应用系统。客户端通过网络数据报文方式发送请求数据，并接收服务端的响应数据报文。请求数据和响应数据采用 JSON 报文数据格式封装，JSON 数据中的二进制数据应进行 Base64 编码，编码方式应符合 RFC 4648 中第 4 章的规定。String 类型的数据除非特别说明，默认使用 UTF-8 编码。

5.3　组件调用接口

组件调用接口由电子签章系统通过 SDK 组件的方式为应用系统提供本地化的接口调用。SDK 组件直接部署在应用系统中，采用本地化接口调用方式提供服务。组件接口中 int 类型大小为 32 比特。

6　服务调用接口

6.1　基础接口

6.1.1　获取签章服务信息

功能描述：获取签章服务信息，包括签章服务名称、厂家名称和版本号等。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetProviderInfo

调用方式：HTTP　GET(JSON)

请求参数：无

响应参数：获取签章服务信息响应参数的JSON格式见表2。

表2 获取签章服务信息响应参数

参数名称	类型	是否必选	描述
resultCode	Int	是	返回码，0表示成功，非0表示错误码
resultMessage	String	是	与resultCode对应的文字说明(UTF-8编码)
providerName	String	是	UTF-8编码的签章服务名称
providerCompany	String	是	UTF-8编码的签章服务提供商的公司名称
providerVersion	String	是	签章服务版本
extendParam	String	否	UTF-8签章服务扩展信息

6.1.2 获取用户印章信息列表

功能描述：获取用户印章信息列表，返回印章名称、印章标识、印章状态和印章版本号信息列表。对于分散式用章，为枚举智能密码钥匙里的印章信息列表，对于集中式用章，为获取指定的用户印章信息列表。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetUserSealInfoList

调用方式：HTTP　POST(JSON)

请求参数：获取用户印章信息列表JSON格式请求参数见表3。

表3 获取用户印章信息列表请求参数

参数名称	类型	是否必选	描述
requestTime	Int	是	时间值，为当前时间的毫秒数，用于超时校验
userID	String	否	用户ID，集中式用章应为非空，分散式用章可为空
siteID	String	否	外部应用ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
status	Int	是	印章状态要求， 0：所有；1：有效；2：无效；3：冻结
authType	Int	是	认证类型，0：无认证；1：token认证；2：签名值认证
authValue	String	否	认证数据，如果是签名值认证，则签名算法默认为SM2，签名原文为上面所有请求参数顺序组合

响应参数：获取用户印章信息列表JSON格式响应参数见表4。

表 4　获取用户印章信息列表响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
userSealInfoList	[UserSealInfo Object]	是	用户印章信息列表，对应的用户印章信息 JSON 报文结构定义应符合表 A.1 的规定

6.1.3　根据印章名称/ID 查询印章

功能描述：根据印章名称或者印章 ID 查询印章数据。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_QuerySealsByNameOrID

调用方式：HTTP　POST(JSON)

请求参数：根据印章名称/印章 ID 查询印章的 JSON 格式请求参数见表 5。

表 5　根据印章名称/印章 ID 查询印章的请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
sealName	String	否	印章名称
sealID	String	否	印章 ID
status	int	是	印章状态，0：所有；1：有效；2：无效；3：冻结

注：sealName 和 sealID 至少有一项是有效的，如果两项同时有效，则两个条件同时满足。

响应参数：根据印章名称/印章 ID 查询印章的 JSON 格式响应参数见表 6。

表 6　根据印章名称/印章 ID 查询印章响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
sealDataInfoList	[SealDataInfo Object]	是	印章数据信息列表，对应印章数据信息 JSON 报文结构定义应符合表 A.2 的规定

6.1.4　获取用户证书信息列表

功能描述：获取用户证书信息列表。分散式用章枚举智能密码钥匙里的数字证书，集中式用章只能获取指定的用户证书。本接口只获取对应的签名证书，数字证书格式符合 GB/T 20518。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetUserCertInfoList

调用方式：HTTP　POST(JSON)

请求参数：获取用户证书信息列表 JSON 格式请求参数见表 7。

表 7　获取用户证书信息列表请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
userID	String	否	用户 ID，分散式签章为 NULL，集中式签章不能为 NULL
siteID	String	否	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
status	int	是	证书状态，0：所有；1：有效；2：无效；3：冻结
authTypc	int	是	认证类型，0：无认证；1：token 认证；2：签名值认证
authType	int	是	认证类型，0：无认证；1：token 认证；2：签名值认证

响应参数：获取用户证书信息列表的 JSON 格式响应参数见表 8。

表 8　获取用户证书信息列表响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
userCertInfoList	[UserCertInfo Object]	是	用户数字证书列表，对应的用户证书信息 JSON 报文结构定义应符合表 A.3 的规定

6.1.5　根据证书 ID 获取数字证书

功能描述：根据证书 ID，获取用于数字签名的数字证书。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetCertByCertSN

调用方式：HTTP　POST(JSON)

请求参数：获取数字证书的 JSON 格式请求参数见表 9。

表 9　根据证书 ID 获取数字证书请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
certSN	String	是	证书序列号
status	int	是	证书状态，0：所有；1：有效；2：无效；3：冻结

响应参数：根据证书 ID 获取数字证书数据的 JSON 格式响应参数见表 10。

表 10　根据证书 ID 获取数字证书响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
certData	String	是	Base64 编码证书数据，如果数字证书签名算法使用 SM2，应符合 GB/T 20518 的规定

6.1.6　获取签名和杂凑算法标识

功能描述：根据传入的印章 ID 或者证书 ID，查找对应证书和印章对应的签名和杂凑算法，返回签名算法和杂凑算法标识 OID，应符合 GB/T 33560 的规定。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetAlgOID

调用方式：HTTP　POST(JSON)

请求参数：获取签名/杂凑算法标识的 JSON 格式请求参数见表 11。

表 11　获取签名/杂凑算法标识的请求参数

参数名称	类型	是否必选	描述
type	int	是	标识类型， 0：印章 ID；1：证书 ID
dataID	String	是	ID 数据

响应参数：获取签名/杂凑算法标识的 JSON 格式响应参数见表 12。

表 12　获取签名/杂凑算法标识的响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
digestAlgOID	String	是	杂凑算法标识的 OID，应符合 GB/T 33560 的规定
signatureAlgOID	String	是	签名算法标识的 OID，应符合 GB/T 33560 的规定

6.1.7　封装签章结构

功能描述：按照 GB/T 38540—2020 中 7.2 的电子签章生成流程，封装不含时间戳的签章结构体，签章数据结构和版本号根据印章数据结构和版本号决定。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_PackSignedValue

调用方式：HTTP　POST(JSON)

请求参数：封装签章结构的 JSON 格式请求参数见表 13。

表 13　封装签章结构的请求参数

参数名称	类型	是否必选	描述
requestTime	String	是	时间值，为当前时间的毫秒数，用于超时校验
userID	String	否	用户 ID，分散式用章时为 NULL，如果是集中式用章，不能为 NULL
siteID	String	否	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
sealID	String	是	印章标识
propertyInfo	String	否	原文属性，一般为签章保护范围对应的原文地址信息
dataHash	String	是	签章保护原文数据的杂凑值数据的 Base64 编码
authType	int	是	认证类型，0：无认证；1：token 认证；2：签名值认证
authValue	String	否	认证值数据；如果是签名值认证，签名算法默认为 SM2，签名原文为上面所有请求参数顺序组合

响应参数：封装签章结构的 JSON 格式响应参数见表 14。

表 14　封装签章结构的响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
sealSignedValue	String	是	Base64 编码的签章数据结构，如果签名算法使用 SM2，应符合 GB/T 38540 的规定

6.1.8　添加时间戳到签章结构体

功能描述：把时间戳封装到签章结构体中，签章结构体应符合 GB/T 38540 的规定，时间戳数据应符合 GB/T 20520 的规定。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_AddTimestampToSealSignedValue

调用方式：HTTP　POST(JSON)

请求参数：添加时间戳到签章结构体的 JSON 格式请求参数见表 15。

表 15　添加时间戳到签章结构体请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
sealSignedValue	String	是	Base64 编码的签章结构(不带时间戳)
timestamp	String	是	Base64 编码的时间戳，应符合 GB/T 20520 的规定

响应参数：添加时间戳到签章结构体的 JSON 格式响应参数见表 16。

表 16　添加时间戳到签章结构体响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明（UTF-8 编码）
outSealSignedValue	String	是	Base64 编码的带时间戳的签章结构，如果签名算法为 SM2，应符合 GB/T 38540 的规定

6.1.9　封装消息签名结构

功能描述：调用户印章私钥进行数字签名，封装不含时间戳的消息签名数据，如果签名算法是 SM2，消息签名数据应符合 GB/T 35275—2017 中第 8 章规定的签名数据类型。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_PackSignedMessage

调用方式：HTTP　POST(JSON)

请求参数：封装消息签名结构的 JSON 格式请求参数见表 17。

表 17　封装消息签名结构的请求参数

参数名称	类型	是否必选	描述
requestTime	String	是	时间值，为当前时间的毫秒数，用于超时校验
userID	String	否	用户 ID，分散式用章时可为空，集中式用章时不能为空
siteID	String	否	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
certSN	String	是	证书序列号
dataHash	String	是	Base64 编码的签章保护原文数据的杂凑值（不带 ID）
digestMethod	String	是	原文杂凑算法 OID，应符合 GB/T 33560 的规定
authType	int	是	认证类型，0：无认证；1：token 认证；2：签名值认证
authValue	String	否	认证值数据；如果是签名值认证，签名算法默认为 SM2，签名原文为上面所有请求参数顺序组合

响应参数：封装消息签名结构的 JSON 格式响应参数见表 18。

表 18　封装消息签名结构的响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明（UTF-8 编码）
signedMessage	String	是	Base64 编码的消息签名数据，如果使用 SM2 算法，应符合 GB/T 35275 规定的签名数据类型

6.1.10 验证签章/消息签名数据

功能描述：验证电子签章/消息签名数据。签章数据应按照 GB/T 38540—2020 中 7.3 的电子签章验证流程进行验证；消息签名数据应按照 GB/T 35275 的规定对消息签名格式进行验证。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_VerifySignedData

调用方式：HTTP　POST(JSON)

请求参数：验证签章/签名数据的 JSON 格式请求参数见表 19。

表 19　验证签章/签名数据的请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
dataHash	String	是	Base64 编码的杂凑数据
signedData	String	是	Base64 编码的签章/签名数据
signStructType	int	是	签章或者签名格式类型，0：未知；1：GB/T 38540；2：GB/T 35275；3：GM/T 0031；4：pkcs7-RSA

响应参数：验证签章/签名数据的 JSON 格式响应参数见表 20。

表 20　验证签章/签名数据的响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)

6.1.11 解析签章数据

功能描述：从电子签章数据中获取相关信息，包括签章时间、签名算法、签章保护原文 hash 等，至少应支持 GB/T 38540 和 GM/T 0031 规定的电子签章数据。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetSignedValueInfo

调用方式：HTTP　POST(JSON)

请求参数：解析签章数据的 JSON 格式请求参数见表 21。

表 21　解析签章数据的请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
signedValue	String	是	Base64 编码的签章数据
version	int	是	签章数据版本

响应参数：解析签章数据的 JSON 格式响应参数见表 22。

表 22　解析签章数据的响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
signedValueInfo	SignedValueInfo Object	是	签章数据结构体，对应的签章信息 JSON 报文结构应符合表 A.4 的规定

6.1.12　解析印章数据

功能描述：获取用于签章的印章信息，包括印章 ID、印章名称、印模图像、印章所有者证书和制章人证书等信息，至少应支持 GB/T 38540 和 GM/T 0031 规定的电子印章数据。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetSealInfo

调用方式：HTTP POST(JSON)

请求参数：解析印章数据的 JSON 格式请求参数见表 23。

表 23　解析印章数据请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
sealData	String	是	Base64 编码印章数据，如果印章签名算法使用 SM2，应符合 GB/T 38540 或者 GM/T 0031 的规定
version	int	是	印章数据版本

响应参数：解析印章数据的 JSON 格式响应参数见表 24。

表 24　解析印章数据响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
sealInfo	SealInfo Object	是	印章数据结构，对应印章信息 JSON 报文结构应符合表 A.5 的规定

6.2　综合接口

6.2.1　文件定位签章

功能描述：根据签章位置参数对文件进行签章，其中印章及印章的位置参数由调用方设定，页面坐标以左上角为原点，坐标值宜采用比率，即坐标位置与整个页面高度/宽度的百分比。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_SealSignByPosition

调用方式：HTTP　POST(JSON)

请求参数：指定位置签章的 JSON 格式请求参数见表 25。

表 25　指定位置签章请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值， 为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型， 0：原数据，1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定
lockFile	int	是	是否锁定文件； 0：不锁定；1：签章方式锁定；2：签章后再添加消息签名方式锁定
userID	String	否	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
siteID	String	否	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
outFileDataType	int	是	返回数据类型， 0：返回签章数据电文流；1：返回签章数据地址
positionSealSignInfoList	[PositionSeal SignInfo Object]	是	定位签章印章信息列表，对应定位签章印章信息 JSON 报文结构应符合表 A.6 的规定
authType	int	是	认证类型， 0：无认证；1：token 认证；2：签名值认证
authValue	String	否	认证值数据， 如果是签名值认证，签名算法默认为 SM2，签名原文为上面所有请求参数顺序组合

响应参数：指定位置签章的 JSON 格式响应参数见表 26。

表 26　指定位置签章响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明（UTF-8 编码）
outFileDataType	int	是	返回数据类型，0：数据电文流；1：数据地址
outFileData	String	是	Base64 编码的输出数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 outFileDataType 确定

6.2.2　文件关键字签章

功能描述：给定关键字，加盖印章到关键字所在位置。当有多个关键字时，可指定加盖第几个关键

字，默认加盖所有关键字，默认关健字的中心点与印模中心点重合。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_SealSignOnKeywords

调用方式：HTTP　POST(JSON)

请求参数：文件关键字签章的 JSON 格式请求参数见表 27。

表 27　文件关键字签章请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间，为当前时间的毫秒数。该时间用于超时校验，不要用做签章时间
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型，0：原数据；1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定
lockFile	int	是	是否锁定文件；0：不锁定；1：签章方式锁定；2：签章后再添加消息签名方式锁定
userID	String	否	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
siteID	String	否	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
outFileDataType	int	是	返回数据类型， 0：返回签章数据电文流；1：返回签章数据地址
keyWordSealSignInfoList	[KeyWordSeal SignInfo Object]	是	关键字签章的印章信息列表，对应关键字签章印章信息 JSON 报文结构应符合表 A.7 的规定
authType	int	是	认证类型， 0：无认证；1：token 认证；2：签名值认证
authValue	String	否	认证值数据；如果是签名值认证，签名算法默认为 SM2，签名原文为上面所有请求参数顺序组合

响应参数：文件关键字签章的 JSON 格式响应参数见表 28。

表 28　文件关键字签章响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
outFileDataType	int	是	返回数据类型， 0：数据电文流；1：数据地址
outFileData	String	是	Base64 编码的输出数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 outFileDataType 确定

6.2.3 文件加盖骑缝章

功能描述：对文件加盖骑缝章，可设置骑缝章纵坐标位置，以及一个整章切分的份数，同时可设置是否加盖主章，如果存在主章，需要指定主章所在页面的位置坐标。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_SealSignOnCrosspages

调用方式：HTTP　POST(JSON)

请求参数：文件加盖骑缝章的 JSON 格式请求参数见表 29。

表 29　文件加盖骑缝章请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	请求时间，为当前时间的毫秒数。该时间用于超时校验，不要用做签章时间
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型， 0：原数据；1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定
lockFile	int	是	是否锁定文件， 0：不锁定；1：签章方式锁定；2：签章后再添加消息签名方式锁定
userID	String	否	用户 ID， 分散式用章时可为 NULL，集中式用章不能为 NULL
siteID	String	否	外部应用 ID，可区分不同外部应用，适用于服务器签章场景
outFileDataType	int	是	返回数据类型， 0：数据电文流；1：数据地址
pageCrossSealSignInfoList	[PageCrossSealSignInfo Object]	是	骑缝章信息列表，对应骑缝章印章信息 JSON 报文结构应符合表 A.8 的规定
authType	int	是	认证类型， 0：无认证；1：token 认证；2：签名值认证
authValue	String	否	认证值数据； 如果是签名值认证，签名算法默认为 SM2，签名原文为上面所有请求参数顺序组合

响应参数：文件加盖骑缝章的 JSON 格式响应参数见表 30。

表 30　文件加盖骑缝章响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
outFileDataType	int	是	返回数据类型， 0：数据电文流；1：数据地址
outFileData	String	是	Base64 编码的输出数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 outFileDataType 确定

6.2.4　文件添加消息签名

功能描述：对文件添加消息签名，可以锁定签名，也可以不锁定签名。如果使用 SM2，消息签名数据应符合 GB/T 35275—2017 中第 8 章规定的签名数据类型。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_CertSign

调用方式：HTTP　POST(JSON)

请求参数：文件数字签名的 JSON 格式请求参数见表 31。

表 31　文件数字签名请求参数

参数名称	类型	是否必选	描述
requestTime	String	是	时间值，为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型，0：原数据；1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定
lockFile	int	是	是否锁定文件， 0：不锁定；1：添加锁定签名；2：添加两个签名，第一个不锁定文档，第二个锁定文档
userID	String	否	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
siteID	String	否	外部应用 ID，可区分不同外部应用，适用于服务器签章场景
certSN	String	是	证书序列号
outFileDataType	int	是	返回数据类型，0：数据电文流；1：数据地址
authType	int	是	认证类型， 0：无认证；1：token 认证；2：签名值认证
authValue	String	否	认证值数据；如果是签名值认证，签名算法默认为 SM2，签名原文为上面所有请求参数顺序组合

响应参数：文件数字签名的 JSON 格式响应参数见表 32。

表 32　文件数字签名响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
outFileDataType	int	是	返回数据类型， 0：数据电文流；1：数据地址
outFileData	String	是	Base64 编码的输出数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 outFileDataType 确定

6.2.5　文件添加文字或图片注释

功能描述：对文件添加注释，包括文字、图片、水印。支持同时添加多条文字、图片，以及文字和图片混合添加。页面坐标以左上角为原点，可指定文字、图片的页面位置，否则默认添加到所有页面。图片水印和文字水印可设置透明度。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_AddAnnot

调用方式：HTTP　POST(JSON)

请求参数：文件添加图片或文字的 JSON 格式请求参数见表 33。

表 33　文件添加文字或图片注释请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型，0：原数据；1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定
outFileDataType	int	是	返回数据类型，0：数据电文流；1：数据地址
annotdataList	[AnnotDataInfo Object]	是	注释数据信息列表，对应注释数据信息 JSON 报文结构应符合表 A.9 的规定

响应参数：文件添加图片或文字的 JSON 格式响应参数见表 34。

表 34　文件添加图片或文字响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
outFileDataType	int	是	返回数据类型， 0：数据电文流；1：数据地址
outFileData	String	是	Base64 编码的输出数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 outFileDataType 确定

6.2.6 文件签章验证

功能描述：验证已签署文件里所有签章，返回验证结果和签章结构体数据。
接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_VerifySealSignedFile
调用方式：HTTP　POST(JSON)
请求参数：文件签章验证的 JSON 格式请求参数见表 35。

表 35　文件签章验证的请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型， 0：原数据，1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定

响应参数：文件签章验证的 JSON 格式响应参数见表 36。

表 36　文件签章验证的响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
fileVerifyInfo	FileVerifyInfo Object	是	文件验证结果列表，对应文件验证结果 JSON 报文结构应符合表 A.10 的规定

6.2.7 文件签名验证

功能描述：验证已签署文件里所有消息签名，返回验证结果和所有消息签名结构体。
接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_VerifyCertSignedFile
调用方式：HTTP　POST(JSON)
请求参数：文件签名验证的 JSON 格式请求参数见表 37。

表 37　文件签名验证请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值， 为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型， 0：原数据；1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定

响应参数：文件签名验证的 JSON 格式响应参数见表 38。

表 38　文件签名验证响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
fileVerifyInfo	FileVerifyInfo Object	是	文件验证结果列表， 对应文件验证结果 JSON 报文结构应符合表 A.10 的规定

6.2.8　获取文件签章/消息签名个数

功能描述：获取文件中的签章和消息签名个数。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_GetSignCount

调用方式：HTTP　POST(JSON)

请求参数：获取文件签章/签名个数的 JSON 格式请求参数见表 39。

表 39　获取文件签章/签名个数请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型，0：原数据，1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定

响应参数：获取文件签章/签名个数的 JSON 格式响应参数见表 40。

表 40　获取文件签章/签名个数响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
signCount	int	是	文件内签名和签章总数

6.2.9　文件指定签章/消息签名验证

功能描述：验证文件中指定签章或者消息签名，返回验证结果和对应的签章/消息签名结构体。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_VerifySignByIndex

调用方式：HTTP　POST(JSON)

请求参数：指定签章/签名验证的 JSON 格式请求参数见表 41。

表 41　指定签章/签名验证的请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型，0：原数据，1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定
index	String	是	签章/签名索引

响应参数：指定签章/签名验证的 JSON 格式响应参数见表 42。

表 42　指定签章/签名验证的响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
verifySignInfo	VerifySignInfo Object	是	签章/消息签名验证结果，对应签章/消息签名验证结果 JSON 报文结构应符合表 A.11 的规定

6.2.10　文件转图像

功能描述：将版式文件转换成图像数据，每一页转成一张图像，至少支持 JPG 和 PNG 两种格式。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_FileToImage

调用方式：HTTP　POST(JSON)

请求参数：版式文件转图像的 JSNO 格式请求参数见表 43。

表 43　版式文件转图像请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值， 为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型，0：原数据；1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据地址，具体根据入参 fileDataType 确定
imageType	string	是	图像类型，支持“PNG”“JPG”“BMP”
dpi	int	是	像素，默认 120
pageStart	int	是	起始页索引，0 为首页
pageEnd	int	是	终止页面索引，−1 表示取章文档的最后一页

响应参数：版式文件转换图像的 JSON 格式响应参数见表 44。

表 44　版式文件转换图像响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
pageImageInfoList	[PageImageInfo Object]	是	页面图像信息列表，对应页面图像信息 JSON 报文结构应符合表 A.12 的规定

6.2.11　文件验证

功能描述：验证数据电文，包括所有签章和消息签名等。

接口地址：HTTP(S)://SERVER:PORT/SEF/V1/SEF_VerifySignedFile

调用方式：HTTP　POST(JSON)

请求参数：数据电文验证的 JSON 格式请求参数见表 45。

表 45　文件验证请求参数

参数名称	类型	是否必选	描述
requestTime	int	是	时间值，为当前时间的毫秒数，用于超时校验
filename	String	是	UTF-8 编码的文件名
fileDataType	String	是	数据电文类型，0：原数据；1：数据地址
fileData	String	是	Base64 编码待签章数据电文或者 UTF-8 编码的数据电文地址，具体根据入参 fileDataType 确定

响应参数：文件验证的 JSON 格式响应参数见表 46。

表 46　文件验证响应参数

参数名称	类型	是否必选	描述
resultCode	int	是	0 表示成功，非 0 表示错误码
resultMessage	String	是	与 resultCode 对应的文字说明(UTF-8 编码)
fileVerifyInfo	FileVerifyInfo OBJECT	是	文件验证结果，对应文件验证结果 JSON 报文结构应符合表 A.10 的规定

7　组件调用接口

7.1　基础接口

7.1.1　获取签章服务信息

函数原型　　int SEF_GetProviderInfo(char * pchProviderName, int * piProviderNameLen,

char * pchProviderCompany, int * piProviderCompanyLen,
char * pchProviderVersion, int * piProviderVersionLen,
char * pchExtendParam, int * piExtendParamLen);

功能描述　获取签章服务信息，包括签章服务名称、厂家名称、版本号等

参数

pchProviderName[in/out]	签章服务名称（UTF-8 编码），当为 NULL 时，通过 piProviderNameLen 给出长度
piProviderNameLen[in/out]	签章服务名称长度
pchProviderCompany[in/out]	签章服务公司名称（UTF-8 编码），当为 NULL 时，通过 piProviderCompanyLen 给出长度
piProviderCompanyLen[in/out]	签章服务公司名称长度
pchProviderVersion[in/out]	签章服务版本（UTF-8 编码），当为 NULL 时，通过 piProviderVersionLen 给出长度
piProviderVersionLen[in/out]	签章服务版本长度
pchExtendParam [in/out]	签章服务扩展信息（UTF-8 编码），当为 NULL 时，通过 piExtendParamLen 给出长度
piExtendParamLen [in/out]	签章服务扩展信息长度

返回值

FEC_SUCCESS：	成功
其他：	错误码

7.1.2 获取用户印章信息列表

函数原型　int SEF _ GetUserSealInfoList (const int iRequestTime, const char * pchUserID, const char * pchSiteID, const int iStatus, const int iAuthType, const unsigned char * puchAuthValue, const int iAuthValueLen, USERSEALINFO * pUserSealInfoList, int * piUserSealInfoListLen);

功能描述　获取用户印章信息列表，返回印章名称、印章标识、印章状态和印章版本号信息列表。对于智能密码钥匙，枚举智能密码钥匙里的印章信息列表，对于集中式用章，只能获取指定的用户印章信息列表

参数

iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
pchUserID[in]	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
pchSiteID[in]	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
iStatus[in]	对印章状态要求， 0：所有；1：有效；2：无效；3：冻结
iAuthType[in]	认证类型， 0：无认证；1：token 认证；2：签名值认证
puchAuthValue[in]	认证数据
iAuthValueLen[in]	认证数据长度
pUserSealInfoList [in/out]	用户印章信息列表，当为 NULL 时，通过 piUserSealInfoListLen 给出长度，用户印章信息数据结构定义应符合 B.2 的规定
piUserSealInfoListLen [in/out]	用户印章信息列表长度

返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.3 根据印章名称/ID查询印章

函数原型	int SEF_QuerySealsByNameOrID(const int iRequestTime, const char * pchSealID,const char * pchSealName,const int iStatus, SEALDATAINFO * pSealDataInfoList,int * piSealDataInfoListLen)；	
功能描述	根据印章名称或者印章ID查询印章数据，或者查询同时满足两个条件的印章数据。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchSealID[in]	印章ID
	pchSealName[in]	UTF-8编码的印章名称
	iStatus[in]	对印章状态要求， 0：所有；1：有效；2：无效；3：冻结
	pSealDataInfoList[in/out]	印章数据信息列表，当为NULL时，通过piSealDataInfoListLen给出长度，印章数据信息结构定义应符合B.3的规定。
	piSealDataInfoListLen [in/out]	用户印章数据信息列表长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码
备注	根据印章名称查询的印章可能包含多个，因此使用列表格式返回。 印章ID和印章名称两者应有一项是有效的，不能都为NULL，如果两项同时有效，则两个条件同时满足。	

7.1.4 获取用户证书信息列表

函数原型	int SEF_GetUserCertInfoList(const int iRequestTime, const char * pchUserID,const char * pchSiteID,const int iStatus,const int iAuthType, const unsigned char * puchAuthValue,const int iAuthValueLen, USERCERTINFO * pUserCertInfoList,int * piUserCertInfoListLen)；	
功能描述	获取用户证书信息列表，对于智能密码钥匙，枚举智能密码钥匙里的数字证书，对于集中式用章，只能获取指定的用户证书。本接口只获取对应的签名证书，数字证书格式符合GB/T 20518。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchUserID[in]	用户ID，分散式用章时可为NULL，如果是集中式用章，不能为NULL
	pchSiteID[in]	外部应用ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
	iStatus[in]	证书状态，0：所有；1：有效；2：无效；3：冻结
	iAuthType[in]	认证类型， 0：无认证；1：token认证；2：签名值认证
	puchAuthValue[in]	认证数据
	iAuthValueLen[in]	认证数据长度

	pUserCertInfoList[in/out]	用户证书信息列表，当为 NULL 时，通过 piSealDataInfoListLen 给出长度，用户证书信息数据结构定义应符合 B.4 的规定。
	piUserCertInfoListLen [in/out]	用户证书信息列表数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.5 根据证书 ID 获取数字证书

函数原型	int SEF_GetCertByCertSN(const int iRequestTime,const char * pchCertSN, const int iStatus, unsigned char * puchCert,int * piCertLen);	
功能描述	根据证书 ID，获取用于数字签名的数字证书。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchCertSN[in]	证书序列号
	iStatus[in]	证书状态，0：所有；1：有效；2：无效；3：冻结
	puchCert[in/out]	DER 编码证书数据，当为 NULL 时，通过 piCertLen 给出长度
	piCertLen[in/out]	证书数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.6 获取签名算法和杂凑算法标识

函数原型	int SEF_GetAlgOID(const int iType,const char * pchDataID, char * pchDigestAlgOID, int * piDigestAlgOIDLen, char * pchSignAlgOID,int * piSignAlgOIDLen);	
功能描述	根据传入的印章 ID 或者证书 ID，查找对应证书和印章对应的签名和杂凑算法，返回签名算法和杂凑算法标识 OID，应符合 GB/T 33560 的规定。	
参数	iType[in]	标识类型，0：印章 ID；1：证书 ID
	puchDataID[in]	ID 数据，标识类型为 0 时表示印章 ID；标识类型为 1 时表示证书 ID
	puchDigestAlgOID[in/out]	杂凑算法标识的 OID，应符合 GB/T 33560 的规定。当为 NULL 时，通过 piDigestAlgOIDLen 给出长度
	piDigestAlgOIDLen[in/out]	杂凑算法标识的 OID 的长度
	puchSignAlgOID[in/out]	签名算法标识的 OID，应符合 GB/T 33560 的规定。当为 NULL 时，通过 piSignAlgOIDLen 给出长度
	piSignAlgOIDLen[in/out]	签名算法标识的 OID 的长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.7 封装签章结构

函数原型	int SEF_PackSignedValue(const int iRequestTime,const char * pchUserID, const char * pchSiteID,const char * pchSealID, const char * pchPropertyInfo,const unsigned char * puchDataHash, const int iDataHashLen,const int iAuthType, const unsigned char * puchAuthValue,const int iAuthValueLen, unsigned char * puchSealSignedValue,int * piSealSignedValueLen);	
功能描述	按照 GB/T 38540—2020 中 7.2 的电子签章生成流程，封装不含时间戳的签章结构体，签章数据结构和版本号根据印章数据结构和版本号决定。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchUserID[in]	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
	pchSiteID[in]	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
	pchSealID[in]	印章标识
	pchPropertyInfo[in]	UTF-8 编码的原文属性，一般为签章保护范围对应的原文地址信息
	puchDataHash[in]	签章保护原文数据的杂凑数据
	iDataHashLen[in]	dataHash 的数据长度
	iAuthType[in]	认证类型，0：无认证；1：token 认证；2：签名值认证
	puchAuthValue[in]	认证数据
	iAuthValueLen[in]	认证数据长度
	puchSealSignedValue [in/out]	DER 编码的签章数据结构体，如果签名算法使用 SM2，应符合 GB/T 38540 的规定，当为 NULL 时，通过 piSealSignedValueLen 给出长度
	piSealSignedValueLen [in/out]	签章数据结构长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.8 添加时间戳到签章结构体

函数原型	int SEF_AddTimestampToSealSignedValue(const int iRequestTime, const unsigned char * puchSealSignedValue,const int iSealSignedValueLen, const unsigned char * puchTimestamp,const int * iTimestampLen, unsigned char * puchOutSealSignedValue,int * piOutSealSignedValueLen);	
功能描述	把时间戳封装到签章结构体中，时间戳数据应符合 GB/T 20520 的规定，签章结构体应符合 GB/T 38540 的规定。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	puchSealSignedValue[in]	DER 编码的签章数据结构(不带时间戳)
	iSealSignedValueLen[in]	输入签章结构长度

	puchTimestamp[in]	DER 编码的时间戳，应符合 GB/T 20520 的规定
	iTimestampLen[in]	时间戳长度
	puchOutSealSignedValue [in/out]	DER 编码的带时间戳的签章结构，如果签名算法使用 SM2，应符合 GB/T 38540 的规定，当为 NULL 时，通过 piOutSealSignedValueLen 给出长度
	piOutSealSignedValueLen [in/out]	输出签章结构长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.9 封装消息签名结构

函数原型	int SEF _ PackSignedMessage (const int iRequestTime, const char * pchUserID,const char * pchSiteID,const char * pchCertSN, const char * pchDigestAlgOID,const unsigned char * puchDataHash, const int iDataHashLen,const int iAuthType, const unsigned char * puchAuthValue,const int iAuthValueLen, unsigned char * puchSignedMessage,int * piSignedMessageLen);	
功能描述	调用用户印章的签名，封装不含时间戳的消息签名数据，如果签名算法是 SM2，消息签名数据应符合 GB/T 35275—2017 中第 8 章规定的签名数据类型。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchUserID[in]	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
	pchSiteID[in]	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
	pchCertSN[in]	证书序列号
	pchDigestAlgOID[in]	原文杂凑算法 OID，应符合 GB/T 33560 的规定
	puchDataHash[in]	签章保护原文数据的杂凑数据
	iDataHashLen[in]	dataHash 的数据长度
	iAuthType[in]	认证类型， 0：无认证；1：token 认证；2：签名值认证
	puchAuthValue[in]	认证数据
	iAuthValueLen[in]	认证数据长度
	puchSignedMessage [in/out]	DER 编码的消息签名数据，如果使用 SM2 算法，应符合 GB/T 35275 规定的签名数据类型。当为 NULL 时，通过 piSignedMessageLen 给出长度
	piSignedMessageLen [in/out]	消息签名数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.10 验证签章/消息签名数据

函数原型	int SEF_VerifySignedData(const int iRequestTime, const unsigned char * puchDataHash,const int iDataHashLen,const int iSignStructType, unsigned char * puchSignedData,int * piSignedDataLen);	
功能描述	验证电子签章/签名数据，对于签章数据，按照 GB/T 38540—2020 中 7.3 或者 GM/T 0031—2014 中 6.2.3 的电子签章验证流程进行验证；对于签名数据，按照 GB/T 35275—2017 中第 8 章签名数据类型格式进行验证。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	puchDataHash[in]	印章保护原文的 hash 值
	iDataHashLen[in]	Hash 值长度
	iSignStructType	签章或者签名格式类型，0：未知；1：GB/T 38540；2：GB/T 35275；3：GM/T 0031；4：pkcs7-RSA
	puchSignedData[in]	DER 编码的签章/签名数据结构体
	piSignedDataLen[in]	签章数据结构体长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.11 解析签章数据

函数原型	int SEF_GetSignedValueInfo(const int iRequestTime,const unsigned char * puchSignedValue,const int pSignedValueLen,const int iVersion, SIGNEDVALUEINFO * pSignedValueInfo,int * piSignedValueInfoLen);	
功能描述	解析电子签章数据信息，包括签章时间、签名算法、签章人数字证书、签章保护原文 hash 等信息，至少应支持 GB/T 38540 和 GM/T 0031 规定的电子签章数据。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	puchSignedValue[in]	DER 编码的签章数据结构体
	piSignedValueLen[in]	签章数据结构体长度
	iVersion[in]	签章数据版本
	pSignedValueInfo [in/out]	签章信息数据结构体，当为 NULL 时，通过 piSignedValueInfoLen 给出长度，签章信息数据结构定义应符合 B.5 的规定
	piSignedValueInfoLen [in/out]	签章数据结构体长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.12 解析印章数据

函数原型	int SEF_GetSealInfo(const int iRequestTime, const unsigned char * puchSealData,const int iSealDataLen,const int iVersion, SEALINFO * pSealInfo,int * piSealInfoLen);	
功能描述	解析电子印章数据信息，包括印章 ID、印章名称、印模图像、印章所有者证书和制章人证书等信息。至少应支持 GB/T 38540 和 GM/T 0031 格式的电子印章数据。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验

	puchSealData[in]	DER 编码的印章数据结构体
	iSealDataLen[in]	印章数据结构体长度
	iVersion[in]	印章数据版本
	pSealInfo[in/out]	印章信息数据结构体，当为 NULL 时，通过 piSealInfoLen 给出长度，印章数据结构定义应符合 B.6 的规定
	piSealInfoLen[in/out]	印章数据结构体长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.1.13 获取错误信息

函数原型	int SEF_GetErrMessage(const int iErrCode,char * pchErrMessage, int * piErrMessageLen);	
功能描述	根据错误代码获取错误信息。	
参数	iErrCode[in]	调用上一次接口返回的错误代码
	puchErrMessage[in/out]	UTF-8 编码的错误信息，当为 NULL 时，通过 piErrMessageLen 给出长度
	piErrMessageLen[in/out]	错误信息长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2 综合接口

7.2.1 文件定位签章

函数原型	int SEF_SealSignByPosition(const int iRequestTime, const char * pchFileName,const int iFileLock,const int iFileDataType, const unsigned char * puchFileData,const int iFileDataLen, const char * pchUserID,const char * pchSiteID, const POSITIONSEALSIGNINFO * pPositionSealSignInfoList, const int iPositionSealSignInfoListLen,const int iAuthType, const unsigned char * puchAuthValue,const int iAuthValueLen, const int iOutFileDataType, unsigned char * puchOutFileData,int * piOutFileDataLen);	
功能描述	根据签章位置参数对文件进行签章，其中印章及印章的位置由调用方设定，坐标以左上角为原点，坐标值宜采用比率，即坐标位置与整个页面高度/宽度的百分比。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileLock[in]	是否锁定文件；0：不锁定；1：签章方式锁定；2：签章后再添加消息签名方式锁定。
	iFileDataType[in]	数据电文类型，0：原数据，1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度

	pchUserID[in]	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
	pchSiteID[in]	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
	pPositionSealSignInfoList[in]	定位签章印章信息列表，定位签章印章信息数据结构定义应符合 B.7 的规定
	iPositionSealSignInfoList Len[in]	定位签章印章信息列表数组个数
	iAuthType[in]	认证类型， 0：无认证；1：token 认证；2：签名值认证
	puchAuthValue[in]	认证数据
	iAuthValueLen[in]	认证数据长度
	iOutFileDataType[in]	输出数据电文类型，0：数据电文流，1：数据地址
	puchOutFileData[in/out]	输出二进制数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 iOutFileDataType 来确定。当为 NULL 时，通过 piOutFileDataLen 给出长度
	piOutFileDataLen[in/out]	输出数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.2 文件关键字签章

函数原型	int SEF_SealSignOnKeywords(const int iRequestTime, const char * pchFileName,const int iFileLock,const int iFileDataType, const unsigned char * puchFileData,const int iFileDataLen, const char * pchUserID, const char * pchSiteID, const KEYWORDSEALSIGNINFO * pKeyWordSealSignInfoList, const int iKeyWordSealInfoListLen,const int iAuthType, const unsigned char * puchAuthValue,const int iAuthValueLen, const int iOutFileDataType, unsigned char * puchOutFileData,int * piOutFileDataLen);
功能描述	给定关键字，加盖印章到关键字所在位置。当有多个关键字时，可指定加盖第几个关键字，默认加盖所有关键字。

参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileLock[in]	是否锁定文件；0：不锁定；1：签章方式锁定；2：签章后再添加消息签名方式锁定。
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pchUserID[in]	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL

	pchSiteID[in]	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
	pKeyWordSealSignInfoList[in]	关键字签章印章信息列表，关键字签章印章信息数据结构定义应符合 B.8 的规定
	iKeyWordSealSignInfoListLen[in]	关键字签章印章信息列表数据长度
	iAuthType[in]	认证类型，0：无认证；1：token 认证；2：签名值认证
	puchAuthValue[in]	认证数据
	iAuthValueLen[in]	认证数据长度
	iOutFileDataType[in]	输出数据电文类型，0：数据电文流，1：数据地址
	puchOutFileData[in/out]	输出二进制数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 iOutFileDataType 来确定。当为 NULL 时，通过 piOutFileDataLen 给出长度
	piOutFileDataLen[in/out]	输出数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.3 文件加盖骑缝章

函数原型

```
int SEF_SealSignOnCrosspages(const int iRequestTime,
const char * pchFileName,const int iFileLock,const int iFileDataType,
const unsigned char * puchFileData,const int iFileDataLen,
const char * pchUserID,const char * pchSiteID,
const PAGECROSSSEALSIGNINFO * pPageCrossSealInfoList,
const int iPageCrossSealInfoListLen,const int iAuthType,
const unsigned char * puchAuthValue,const int iAuthValueLen,
const int iOutFileDataType,
unsigned char * puchOutFileData,int * piOutFileDataLen)
```

功能描述　对文件加盖骑缝章，可设置骑缝章纵坐标位置，以及一个整章切分的份数，同时可设置是否加盖主章，如果存在主章，需要指定主章所在页面的位置坐标。

参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileLock[in]	是否锁定文件；0：不锁定；1：签章方式锁定；2：签章后再添加消息签名方式锁定。
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pchUserID[in]	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
	pchSiteID[in]	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章

	pPageCrossSealSignInfoList[in]	骑缝签章印章信息列表，骑缝章印章信息数据结构定义应符合 B.9 的规定
	iPageCrossSealSignInfoListLen[in]	骑缝签章印章信息列表数据长度
	iAuthType[in]	认证类型，0：无认证；1：token 认证；2：签名值认证
	puchAuthValue[in]	认证数据
	iAuthValueLen[in]	认证数据长度
	iOutFileDataType[in]	输出数据电文类型，0：数据电文流，1：数据地址
	puchOutFileData[in/out]	输出二进制数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 iOutFileDataType 来确定。当为 NULL 时，通过 piOutFileDataLen 给出长度
	iOutFileDataLen[in/out]	输出数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.4 文件添加消息签名

函数原型	int SEF_CertSign(const int iRequestTime,const char * pchFileName, const int iFileLock,const int iFileDataType, const unsigned char * puchFileData,const int iFileDataLen, const char * pchUserID,const char * pchSiteID,const char * pchCertSN, const int iAuthType,const unsigned char * puchAuthValue, const int iAuthValueLen,const int iOutFileDataType, unsigned char * puchOutFileData,int * piOutFileDataLen);	
功能描述	对文件添加消息签名，文件可以被锁定签名，也可以不被锁定签名。如果使用 SM2，消息签名数据应符合 GB/T 35275—2017 中第 8 章规定的签名数据类型。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileLock[in]	是否锁定文件，0：不锁定；1：锁定。
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pchUserID[in]	用户 ID，分散式用章时可为 NULL，如果是集中式用章，不能为 NULL
	pchSiteID[in]	外部应用 ID，当有多个外部应用时可区分不同外部应用，适用于集中式签章
	pchCertSN[in]	证书 ID
	iAuthType[in]	认证类型，0：无认证；1：token 认证；2：签名值认证
	puchAuthValue[in]	认证数据
	iAuthValueLen[in]	认证数据长度

	iOutFileDataType[in]	输出数据电文类型，0：数据电文流，1：数据地址
	puchOutFileData[in/out]	输出二进制数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 iOutFileDataType 来确定。当为 NULL 时，通过 piOutFileDataLen 给出长度
	piOutFileDataLen[in/out]	输出数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.5 文件添加文字或图片

函数原型	int SEF_AddAnnot (const int iRequestTime,const char * pchFileName, const int iFileDataType,const unsigned char * puchFileData, const int iFileDataLen,const ANNOTDATAINFO * pAnnotDataInfoList, const int iAnnotDataInfoListLen,const int iOutFileDataType, unsigned char * puchOutFileData,int * piOutFileDataLen);	
功能描述	对文件添加注释，包括文字、图片、水印。支持同时添加多条文字、图片，以及文字和图片混合添加。页面坐标以左上角为原点，可指定文字、图片的页面位置，否则默认添加到所有页面。图片水印和文字水印可设置透明度。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，具体根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pAnnotDataInfoList[in]	注释数据信息列表，注释数据信息数据结构定义应符合 B.10 的规定
	iAnnotDataInfoListLen[in]	注释数据信息列表长度
	iOutFileDataType[in]	输出数据电文类型，0：数据电文流；1：数据地址
	puchOutFileData[in/out]	输出二进制数据电文流，或者 UTF-8 编码的输出数据地址，具体根据入参 iOutFileDataType 来确定。当为 NULL 时，通过 piOutFileDataLen 给出长度
	piOutFileDataLen[in/out]	输出数据电文长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.6 文件签章验证

函数原型	int SEF_VerifySealSignedFile(const int iRequestTime, const char * pchFileName,const int iFileDataType, const unsigned char * puchFileData,const int iFileDataLen, FILEVERIFYINFO * pFileVerifyInfo,int * piFileVerifyInfoLen);	
功能描述	验证已签署文件里所有签章，返回验证结果和签章结构体数据。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验

	pchFileName[in]	UTF-8 编码的文件名
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，具体根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pFileVerifyInfo[in/out]	文件签章验证结果，文件验证结果数据结构定义应符合 B.11 的规定，当为 NULL 时，通过 piFileVerifyInfoLen 给出长度
	piFileVerifyInfoLen [in/out]	签名验证结果数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.7 文件签名验证

函数原型	int SEF_VerifyCertSignedFile(const int iRequestTime, const char * pchFileName,const int iFileDataType, const unsigned char * puchFileData,const int iFileDataLen, FILEVERIFYINFO * pFileVerifyInfo,int * piFileVerifyInfoLen)；	
功能描述	验证已签署文件里所有消息签名，返回验证结果和所有消息签名结构体。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，具体根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pFileVerifyInfo[in/out]	文件签名验证结果，文件验证结果数据结构定义应符合 B.11 的规定，当为 NULL 时，通过 piFileVerifyInfoLen 给出长度
	piFileVerifyInfoLen[in/out]	签名验证结果数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.8 获取文件签章/消息签名个数

函数原型	int SEF_GetSignCount(const int iRequestTime,const char * pchFileName, const int iFileDataType,const unsigned char * puchFileData, const int iFileDataLen,int * piSignCount)；	
功能描述	获取文件中的签章和消息签名个数。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，具体根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度

	piSignCount[out]	签名和签章总数
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.9 文件指定签章/消息签名验证

函数原型	int SEF_VerifySignByIndex(const int iRequestTime, const char * pchFileName,const int iFileDataType, const unsigned char * puchFileData,const int iFileDataLen, const int iSignIndex, VERIFYSIGNINFO * pVerifySignInfo,int * piVerifySignInfoLen)；	
功能描述	验证文件中指定签章或者消息签名，验证成功后返回验证结果和对应的签章/消息签名结构体。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，具体根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	iSignIndex[in]	指定签章/消息签名索引
	pVerifySignInfo[in/out]	文件签章/消息签名验证结果，签章/消息签名验证结果数据结构定义应符合 B.12 的规定，当为 NULL 时，通过 piVerifySignInfoLen 给出长度
	piVerifySignInfoLen[in/out]	验证结果数据长度
返回值	SEF_SUCCESS：	成功
	其他：	错误码

7.2.10 文件转图像

函数原型	int SEF_FileToImage(const int iRequestTime,const char * pchFileName, const int iFileDataType,const unsigned char * puchFileData, const int iFileDataLen,const char * pchImageType,const int iDpi, const int iPageStart,const int iPageEnd, PAGEIMAGEINFO * pPageImageInfoList,int * piPageImageInfoListLen)；	
功能描述	将版式文件转换成图像数据，每一页转成一张图像，至少支持 JPG 和 PNG 两种格式。	
参数	iRequestTime[in]	时间值，为当前时间的毫秒数，用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileDataType[in]	数据电文类型，0：原数据；1：数据地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址，具体根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pchImageType[in]	图像类型，支持"PNG"、"BMP"和"JPG"
	iDpi[in]	图像像素，默认 120

	iPageStart[in]	起始页索引,0 为首页
	iPageEnd[in]	终止页面索引,-1 表示取章文档的最后一页
	pPageImageInfoList[in/out]	页面图像信息列表,页码图像信息数据结构定义应符合 B.13 的规定,当为 NULL 时,通过 piPageImageInfoListLen 给出长度
	piPageImageInfoListLen [in/out]	页面图像信息列表长度
返回值	SEF_SUCCESS:	成功
	其他:	错误码

7.2.11 文件验证

函数原型	int SEF_VerifySignedFile(const int iRequestTime,const char * pchFileName, const int iFileDataType,const unsigned char * puchFileData,const int iFileDataLen,FILEVERIFYINFO * pFileVerifyInfo,int * piFileVerifyInfoLen)	
功能描述	验证数据电文,包括所有签章和消息签名等。	
参数	iRequestTime[in]	时间值,为当前时间的毫秒数,用于超时校验
	pchFileName[in]	UTF-8 编码的文件名
	iFileDataType[in]	数据电文类型,0:原数据;1:地址
	puchFileData[in]	文件二进制数据流或者 UTF-8 编码的数据地址,具体根据 iFileDataType 确定
	iFileDataLen[in]	数据电文长度
	pFileVerifyInfo[in/out]	数据电文验证结果,数据电文验证结果数据结构定义应符合 B.11 的规定,当为 NULL 时,通过 piFileVerifyInfoLen 给出长度
	piFileVerifyInfoLen[in/out]	签章验证结果数据长度
返回值	SEF_SUCCESS:	成功
	其他:	错误码

附 录 A
（规范性）
自定义 JSON 报文数据结构

A.1 用户印章信息 JSON 报文结构

用户印章信息 JSON 报文结构见表 A.1。

表 A.1 用户印章信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
UserSealInfo Object	sealID	String	是	印章 ID
	sealName	String	是	UTF-8 编码的印章名称
	status	int	是	对印章状态要求， 0：未知；1：有效；2：无效；3：冻结
	version	int	是	印章版本

A.2 印章数据信息 JSON 报文结构

印章数据信息 JSON 报文结构见表 A.2。

表 A.2 印章数据信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
SealDataInfo Object	sealData	String	是	Base64 编码的印章数据
	status	int	是	印章当前状态， 0：未知；1：有效；2：无效；3：冻结
	version	int	是	印章版本

A.3 用户证书信息 JSON 报文结构

用户证书信息 JSON 报文结构见表 A.3。

表 A.3 用户证书信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
UserCertInfo Object	certSN	String	是	证书序列号
	commonName	String	是	UTF-8 编码的证书通用名称
	status	int	是	证书状态， 0：未知；1：有效；2：无效；3：冻结

A.4 签章信息 JSON 报文结构

签章信息 JSON 报文结构见表 A.4。

表 A.4 签章信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
SignedValueInfo Object	version	int	是	签章数据版本
	sealData	String	是	Base64 编码的电子印章数据
	signDateTime	String	是	签章时间，GeneralizedTime 时间格式
	dataHash	String	是	Base64 编码原文杂凑值
	propertyInfo	String	否	UTF-8 编码原文属性信息，如果存在，应解析
	signerCert	String	是	Base64 编码的签章人证书，如果签名使用 SM2 算法，应符合 GB/T 20518 的规定
	signatureAlgOID	String	是	签名算法 OID，应符合 GB/T 33560 的规定
	timeStamp	String	否	Base64 编码的时间戳，应符合 GB/T 20520 的规定

A.5 印章信息 JSON 报文结构

印章信息 JSON 报文结构见表 A.5。

表 A.5 印章信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
SealInfo Object	sealID	String	是	印章标识，对应 GB/T 38540 的 esID
	version	int	是	印章版本
	venderID	String	是	厂商标识，对应 GB/T 38540 的 Vid
	type	int	是	印章类型
	name	String	是	UTF-8 编码的印章名称
	certListType	int	是	签章者证书信息类型， 1：证书；2：证书杂凑值
	certListInfo	String	是	Base64 编码的签章者证书列表信息，证书列表信息采用 sequence 结构封装
	validStart	String	是	有效起始时间， GeneralizedTime 时间格式
	validEnd	String	是	有效结束时间，GeneralizedTime 时间格式
	createDate	String	是	制章日期，GeneralizedTime 时间格式
	makerCert	String	是	Base64 编码的制章人证书

表 A.5 印章信息 JSON 报文结构（续）

自定义类型	参数名称	类型	是否必选	描述
SealInfo Object	signatureAlgOID	String	是	签名算法 OID 标识， 应符合 GB/T 33560 的规定
	imageData	String	是	Base64 编码的印章图像数据
	type	String	是	印章图像类型， 支持“PNG”“BMP”“JPG”
	imageHeight	int	是	印章图像的高度，单位毫米
	imageWidth	int	是	印章图像的宽度，单位毫米

A.6 定位签章印章信息 JSON 报文结构

定位签章印章信息 JSON 报文结构见表 A.6。

表 A.6 定位签章印章信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
PositionSealSign Info Object	sealID	String	是	印章标识
	pageIndex	int	是	文件页码，页码计数从 0 开始
	pageCount	int	是	印章加盖总页数，-1：表示所有页
	widthRatio	float	是	印模左上角对应的 x 坐标与页面宽度的比率
	heightRatio	float	是	印模左上角对应的 y 坐标与页面高度的比率

A.7 关键字签章印章信息 JSON 报文结构

关键字签章印章信息 JSON 报文结构见表 A.7。

表 A.7 关键字签章印章信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
KeyWordSealSignInfo Object	keyWord	String	是	签章关键字字段，UTF-8 编码，文档中要转码后搜索
	sealID	String	是	印章 ID
	pos	array	是	位置编号， 0：全部关键字；X：从前向后找第 X 个关键字；$-X$：从后向前找第 X 个关键字

A.8 骑缝章印章信息 JSON 报文结构

骑缝章印章信息 JSON 报文结构见表 A.8。

表 A.8 骑缝章印章信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
PageCrossSealSignInfoObject	mainSeal	int	是	是否加盖主章。 1:盖;0:不盖
	sealID	String	是	印章标识
	pageIndex	int	否	主章签署的页码, 0 为起始页,mainSeal 为 1 时必选
	widthRatio	float	否	主章左上角对应的 x 坐标与页面宽度的比率,mainSeal 为 1 时必选
	heightRatio	float	否	主章左上角对应的 y 坐标与页面高度的比率,mainSeal 为 1 时必选
	posY	float	是	骑缝章左上角对应的 y 坐标与页面高度的比率
	pageCount	int	是	一个骑缝章印模图片所能覆盖的页数(-1 默认覆盖所有页数)

A.9 注释数据信息 JSON 报文结构

注释数据信息 JSON 报文结构见表 A.9。

表 A.9 注释数据信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
AnnotDataInfoObject	annotType	int	是	注释类型,0:水印,加盖到文档所有页面,透明度 50%;1:普通注释,添加到指定页面指定位置,不透明
	dataType	int	是	注释数据类型,0:文字;1:图片
	annotData	String	是	dataType=0 是表示文字注释数据,UTF-8 编码后再做 Base64 编码;dataType=1 时,表示图片注释数据
	pageIndex	int	否	页面索引,从 0 开始,annotType=1 时必选
	widthRatio	float	否	印模左上角对应的 x 坐标与页面宽度的比率,annotType=1 时必选
	heightRatio	float	否	印模左上角对应的 y 坐标与页面高度的比率,annotType=1 时必选
	imageHeight	float	否	文字/图像在页面上显示高度,单位毫米,annotType=1 时必选
	imageWidth	float	否	文字/图像在页面上显示宽度,单位毫米,annotType=1 时必选

A.10 文件验证结果 JSON 报文结构

文件验证结果 JSON 报文结构见表 A.10。

表 A.10 文件验证结果 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
FileVerifyInfo Object	resultCode	int	是	验证结果，0：全部成功；1：部分成功；2：全部失败；3：文件未签署
	resultMessage	String	是	与 resultCode 对应的文字描述，UTF-8 编码
	totalCount	int	是	签章个数(含签名)，为 0 代表此文件没有被签署
	signInfoList	[VerifySignInfo Object]	是	签章/消息签名验证结果结构体，见表 A.11

A.11 签章/消息签名验证结果 JSON 报文结构

签章/消息签名验证结果 JSON 报文结构见表 A.11。

表 A.11 签章/消息签名验证结果 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
VerifySignInfo Object	signIndex	int	是	签章/签名索引
	pageIndex	[int]	是	页面索引列表，如果一个印章同时加盖在多页上，应返回所有页面，如[0,1,2…]，骑缝章返回-1(因为骑缝章默认加盖所有页面)
	verifyResult	int	是	验证结果，成功返回 0，失败非 0；如果选择不验证，返回 0
	resultMsg	String	是	验证结果的文字描述，UTF-8 编码
	signStructType	int	是	签章或者签名格式类型，0：未知；1：GB/T 38540；2：GB/T 35275；3：GM/T 0031；4：pkcs7-RSA
	signedValue	String	是	Base64 编码的签章/签名数据
	status	int	是	印章当前状态，0：未知；1：有效；2：无效；3：冻结

A.12 页面图像信息 JSON 报文结构

页面图像信息 JSON 报文结构见表 A.12。

表 A.12　页面图像信息 JSON 报文结构

自定义类型	参数名称	类型	是否必选	描述
PageImageInfo Object	pageindex	int	是	页面索引，0 为起始页
	pageHeight	float	是	页面高度，单位毫米
	pageWidth	float	是	页面宽度，单位毫米
	pageImage	String	是	Base64 编码的页面图像数据

附 录 B
（规范性）
自定义 C 语言数据结构

B.1 常量定义

数据常量标识定义了在本文件中用到的常量的取值，见表 B.1。

表 B.1 常量定义

常量	数值	描述
BUF_SIZE	256	通用字节长度
MAX_CERT_SIZE	5120	数字证书最大长度
MAX_SEALDATA_SIZE	51200	印章数据最大长度

B.2 用户印章信息数据结构

a） 类型定义

```
typedef struct SEF_USERSEALINFO{
  char      sealID[BUF_SIZE];
  char      sealName[BUF_SIZE];
  int       status;
  int       version;
}USERSEALINFO, *PUSERSEALINFO;
```

b） 数据项描述见表 B.2

表 B.2 用户印章信息数据结构参数说明

数据项	类型	描述	备注
sealID	char *	印章 ID	—
sealName	char *	UTF-8 编码的印章名称	—
status	int	对印章状态要求，0：未知；1：有效；2：无效；3：冻结	—
version	int	印章版本	—

B.3 印章数据信息数据结构

a） 类型定义

```
typedef struct SEF_SEALDATAINFO{
  int               status;
  int               version;
  unsigned char     sealData[MAX_SEALDATA_SIZE];
  int               sealDateLen;
```

} SEALDATAINFO, * PSEALDATAINFO;

b) 数据项描述见表 B.3

表 B.3 印章数据信息数据结构参数说明

数据项	类型	描述	备注
status	int	印章当前状态， 0:未知、1:有效、2:无效、3:冻结	—
version	int	印章版本	—
sealData	unsigned char *	DER 编码的印章数据	—
sealDateLen	int	印章数据长度	—

B.4 用户证书信息数据结构

a) 类型定义

```
typedef struct SEF_USERCERTINFO{
  char      certSN[BUF_SIZE];
  char      commonName[BUF_SIZE];
  int       status;
} USERCERTINFO, * PUSERCERTINFO;
```

b) 数据项描述见表 B.4

表 B.4 用户证书信息数据结构参数说明

数据项	类型	描述	备注
certSN	char *	证书序列号	—
commonName	char *	UTF-8 编码的证书通用名称	—
status	int	证书状态， 0:未知;1:有效;2:无效;3:冻结	—

B.5 签章信息数据结构

a) 类型定义

```
typedef struct SEF_SIGNEDVALUEINFO{
  int                 version;
  unsigned char       sealData[MAX_SEALDATA_SIZE];
  int                 sealDataLen;
  char                signDateTime[BUF_SIZE];
  unsigned char       dataHash[32];
  int                 dataHashLen;
  unsigned char       signerCert[MAX_CERT_SIZE];
  char                signatureAlgOID[32];
  unsigned char *     propertyInfo;
```

```
    int                     propertyInfoLen;
    unsigned char *         timeStamp;
    int                     timeStampLen;
} SIGNEDVALUEINFO, *PSIGNEDVALUEINFO;
```

b) 数据项描述见表 B.5

表 B.5 签章信息数据结构参数说明

数据项	类型	描述	备注
version	int	签章数据版本	—
sealData	unsigned char *	DER 编码的电子印章数据	—
sealDataLen	int	电子印章数据长度	—
signDateTime	char *	签章时间，GeneralizedTime 时间格式	—
dataHash	unsigned char *	原文杂凑值二进制数据	—
dataHashLen	int	原文杂凑数据长度	—
signerCert	unsigned char *	DER 编码的签章人证书，如果签名使用 SM2 算法，应符合 GB/T 20518 的规定	—
signerCertLen	int	签章人证书数据长度	—
signatureAlgOID	char *	签名算法 OID，应符合 GB/T 33560 的规定	—
propertyInfo	unsigned char *	UTF-8 编码原文属性信息，如果存在，应解析	—
propertyInfoLen	int	原文属性信息长度	—
timeStamp	unsigned char *	DER 编码的时间戳，应符合 GB/T 20520 的规定	—
timeStampLen	int	时间戳数据长度	—

B.6 印章信息数据结构

a) 类型定义

```
typedef struct SEF_SEALINFO{
    char                    sealID[BUF_SIZE];
    int                     version;
    char                    verderID[BUF_SIZE];
    int                     type;
    char                    name[BUF_SIZE];
    int                     certListType;
    unsigned char *         certListInfo;
    int                     certListInfoLen;
    char                    validStart[32];
    char                    validEnd[32];
    char                    createDate[32];
    unsigned char           makerCert[MAX_CERT_SIZE];
```

```
    int                 makerCertLen;
    char                signatureAlgOID[32];
    unsigned char *     imageData;
    int                 imageDataLen;
    char                imageType[16];
    int                 imageHeight;
    int                 imageHeight;
} SEALINFO, * PSEALINFO;
```

b) 数据项描述见表 B.6

表 B.6 印章信息数据结构参数说明

数据项	类型	描述	备注
sealID	char *	印章标识，对应 GB/T 38540 的 csID	—
version	int	印章版本	—
venderID	char *	厂商标识，对应 GB/T 38540 的 Vid	—
type	int	印章类型，应符合 GB/T 38540 的规定	—
name	char *	UTF-8 编码的印章名称	—
certListType	int	签章者证书信息类型， 1：证书；2：证书杂凑值	—
certListInfo	unsigned char *	DER 编码的签章者证书列表信息，证书列表信息采用 sequence 结构封装	—
certListInfoLen	int	签章者证书列表信息数据长度	—
validStart	char *	有效起始时间，GeneralizedTime 时间格式	—
validEnd	char *	有效结束时间，GeneralizedTime 时间格式	—
createDate	char *	制章日期，GeneralizedTime 时间格式	—
makerCert	unsigned char *	DER 编码的制章人数字证书	—
makerCertLen	int	制章人数字证书数据长度	—
signatureAlgOID	char *	签名算法 OID 标识，应符合 GB/T 33560 的规定	—
imageData	unsigned char *	印章图像二进制数据	—
imageDataLen	int	印章图像数据长度	—
imageType	char *	印章图像类型，支持“PNG”“BMP”“JPG”	—
imageHeight	int	印章图像的高度，单位毫米	—
imageWidth	int	印章图像的宽度，单位毫米	—

B.7 定位签章印章信息数据结构

a) 类型定义

```
typedef struct SEF_POSITIONSEALSIGNINFO{
```

```
    char        sealID[BUF_SIZE];
    int         pageIndex;
    int         pageCount;
    float       widthRatio;
    float       heightRatio;
} POSITIONSEALSIGNINFO, * PPOSITIONSEALSIGNINFO;
```

b) 数据项描述见表 B.7

表 B.7 定位签章印章信息数据结构参数说明

数据项	类型	描述	备注
sealID	char *	印章标识	—
pageIndex	int	文件页码，页码计数从 0 开始	—
pageCount	int	印章加盖总页数(从 pageIndex 开始签章)，-1 表示所有页	—
widthRatio	float	印模左上角对应的 x 坐标与页面宽度的比率	—
heightRatio	float	印模左上角对应的 y 坐标与页面高度的比率	—

B.8 关键字签章印章信息数据结构

a) 类型定义

```
typedef struct SEF_KEYWORDSEALSIGNINFO{
    char        sealID[BUF_SIZE];
    char        keyWord[BUF_SIZE];
    int         pos[BUF_SIZE];
    int         posNumber;
} KEYWORDSEALSIGNINFO, * PKEYWORDSEALSIGNINFO;
```

b) 数据项描述见表 B.8

表 B.8 关键字签章印章信息数据结构参数说明

数据项	类型	描述	备注
sealID	char *	印章 ID	—
keyWord	char *	签章关键字字段，UTF-8 编码，文档中要转码后搜索	—
pos	int *	位置编号， 0：所有的关键字； X：从前向后找第 X 个关键字； $-X$：从后向前找第 X 个关键字	—
posNumber	int	关键字位置数目	最大为 BUF_SIZE

B.9 骑缝章印章信息数据结构

a) 类型定义

```
typedef struct SEF_PAGECROSSSEALSIGNINFO{
char            sealID[BUF_SIZE];
int             mainSeal;
int             pageIndex;
float           widthRatio;
float           heightRatio;
float           posY;
int             pageCount;
} PAGECROSSSEALSIGNINFO, * PPAGECROSSSEALSIGNINFO;
```

b) 数据项描述见表 B.9

表 B.9 骑缝章印章信息数据结构参数说明

数据项	类型	描述	备注
sealID	char *	印章标识	—
mainSeal	int	是否加盖主章，1：盖；0：不盖	—
pageIndex	int	主章签署的页码，0 为起始页，mainSeal 为 1 时必选	—
widthRatio	float	主章左上角对应的 x 坐标与页面宽度的比率，mainSeal 为 1 时必选	—
heightRatio	float	主章左上角对应的 y 坐标与页面高度的比率，mainSeal 为 1 时必选	—
posY	float	骑缝章左上角对应的 y 坐标与页面高度的比率	—
pageCount	int	一个骑缝章印模图片所能覆盖的页数，-1 表示覆盖所有页	—

B.10 注释数据信息数据结构

a) 类型定义

```
typedef struct SEF_ANNOTDATAINFO{
int                 annotType;
int                 dataType;
unsigned char *     annotData;
int                 annotDataLen;
int                 pageIndex;
float               widthRatio;
float               heightRatio;
float                 imageHeight;
float                 imageWidth;
} ANNOTDATAINFO, * PANNOTDATAINFO;
```

b) 数据项描述见表 B.10

表 B.10　注释数据信息数据结构参数说明

数据项	类型	描述	备注
annotType	int	注释类型， 0：水印注释，注释信息透明度 50%； 1：普通注释，不透明	—
dataType	int	注释数据类型，0：文字；1：图片	—
annotData	unsigned char *	dataType = 0 是表示文字注释数据，UTF-8 编码； dataType=1 时，表示二进制的图片数据	—
annotDataLen	int	文字/图片数据长度	—
pageIndex	int	页面索引，从 0 开始计数，-1 代表所有页面	—
widthRatio	float	文字/图片左上角对应的 x 坐标与页面宽度的比率	—
heightRatio	float	文字/图片左上角对应的 y 坐标与页面高度的比率	—
imageHeight	float	文字/图像在页面上显示高度，单位毫米	—
imageWidth	float	文字/图像在页面上显示宽度，单位毫米	—

B.11　文件验证结果数据结构

a)　类型定义

```
typedef struct SEF_FILEVERIFYINFO{
    int                     resultCode;
    char                    resultMessage[BUF_SIZE];
    int                     totalCount;
    VERIFYSIGNINFO * signInfoList;
    int                     signInfoListLen;
} FILEVERIFYINFO, * PFILEVERIFYINFO;
```

b)　数据项描述见表 B.11

表 B.11　文件验证结果数据结构参数说明

数据项	类型	描述	备注
resultCode	int	验证结果，0：全部成功；1：部分成功；2：全部失败；3：文件未签署	—
resultMessage	char *	错误消息，与 resultCode 对应的文字描述，UTF-8 编码	—
totalCount	int	签章个数（含签名），为 0 代表此文件没有被签署	—
signInfoList	VERIFYSIGNINFO	签章/消息签名验证结果数据结构体，见 B.12	—
signInfoListLen	int	文件验证结果结构体数据长度	—

B.12　签章/消息签名验证结果数据结构

a)　类型定义

```
typedef struct SEF_VERIFYSIGNINFO{
    int             signIndex;
    int             pageIndex[BUF_SIZE];
    int             pageIndexNum;
    int             verifyResult;
    char            resultMsg[BUF_SIZE];
    int             signStructType;
    unsigned char * signedValue;
    int             signedValueLen;
    int             status;
} VERIFYSIGNINFO, * PVERIFYSIGNINFO;
```

b) 数据项描述见表 B.12

表 B.12 签章/消息签名验证结果数据结构参数说明

数据项	类型	描述	备注
signIndex	int	签章/签名索引	—
pageIndex	int *	页面索引列表，如果一个印章同时加盖在多页上，应返回所有页面，如[0,1,2…]，骑缝章返回-1(因为骑缝章默认加盖所有页面)	—
pageIndexNum	int	页面索引列表有效数组个数	—
verifyResult	int	验证结果，成功返回 0，失败非 0	—
resultMsg	char *	验证结果的文字描述，UTF-8 编码	—
signStructType	int	签章或者签名格式类型，0：未知；1：GB/T 38540；2：GB/T 35275；3：GM/T 0031；4：pkcs7-RSA	—
signedValue	unsigned char *	DER 编码的签章/签名数据	—
signedValueLen	int	签章/签名数据长度	—
status	int	印章当前状态，0：未知；1：有效；2：无效；3：冻结	—

B.13 页面图像信息数据结构

a) 类型定义

```
typedef struct SEF_PAGEIMAGEINFO{
    int             pageIndex;
    int             pageHeight;
    int             pageWidth;
    unsigned char * pageImage;
    int             pageImageLen;
} PAGEIMAGEINFO, * PPAGEIMAGEINFO;
```

b) 数据项描述见表 B.13

表 B.13　页面图像信息数据结构参数说明

数据项	类型	描述	备注
pageindex	int	页面索引，0 为起始页	—
pageHeight	int	页面高度，单位毫米	—
pageWidth	int	页面宽度，单位毫米	—
pageImage	unsigned char *	页面图像二进制数据	—
pageImageLen	int	页面图像数据长度	—

附 录 C
（资料性）
典型服务调用接口示例

C.1 获取签章服务信息

URL 地址，http://server:port/SEF/V1/SEF_GetProviderInfo，POST
响应数据 JSON 样例：


```
{
    "resultCode":"0",
    "resultMessage":"成功",
    "providerName":"golden eseal",
    "providerCompany":"gomain",
    "providerVersion":"1.35"
}
```


C.2 获取用户印章信息列表

URL 地址，http://server:port/SEF/V1/SEF_GetUserSealInfoList，POST
请求数据 JSON 样例：


```
{
    "ruquestTime":1540886034628,
    "userID":"345678313",
    "status":1,
    "authType":1,
    "authValue":"bxLdikRXVbTPdHSM05e5u5sUoXNKd8-41ZO3MhKoyN5OfkWITDGgn
    r2fwJ0m9E8NYzWKVZvdVtaUgWvsdshFKA"

}
```


响应数据 JSON 样例：


```
{
    "resultCode": "0",
    "resultMessage": "成功",
    "sealList": [  {
        "sealID": "1527490410251",
        "sealName": "测试印章 1",
        "status": 1,
        "version": 4,

      ……
      }, {
      "sealID": "1527488995046",
```

```
            "sealName": "测试印章 2"
            "status": 1,
            "version": 4,

        },
        ......
        ]
    }
```

C.3 文件定位签章

URL 地址,http://server:port/SEF/V1/SEF_SealSignByPosition,POST

请求数据 JSON 样例:

```
{
        "requestTime": 1540886052520,
        "filename": "13454546909.ofd",
        "fileDataType":0 ,
        "fileData":"MXBADCGKD....." ,
        "lockFile":0 ,
        "userID": "345678313",
        "authType":1,
        "authValue":"bxLdikRXVbTPdHSM05e5u5sUoXNKd8-41ZO3MhKoyN5OfkWITDGgn
          r2fwJ0m9E8NYzWKVZvdVtaUgWvsdshFKA",
        "outFileDataType":1,
        "sealSignInfolList": [
            {
                "sealID": "44010400060556",
                "pageIndex": 0,//从第 0 页开始,连续盖 3 页
                "pageCount": 3,
                "widthRatio": 0.5,
                "heightRatio": 0.6,
            },
            {
                "sealID": "44010400060556",
                "pageIndex": 10,//从第 10 页开始,连续盖 2 页
                "pageCount": 2,
                "widthRatio": 0.4,
                "heightRatio": 0.5,
            }
        ]
}
```

响应数据 JSON 样例:

```
{
        "resultCode":0,
        "resultMessage": "成功",
        "outFileDataType":1,
        "outFileData ": "signSuccess/13454546909.ofd"

}
```


附 录 D
（资料性）
典型组件调用接口示例

D.1 获取签章服务信息

```
char    pchProviderName[BUF_SIZE]={0};
int     iProviderNameLen=BUF_SIZE;
char    pchProviderCompany[BUF_SIZE]={0};
int     iProviderCompanyLen=BUF_SIZE;
char    pchProviderVersion[BUF_SIZE]={0};
int     iVersionLen=BUF_SIZE;
char    pchProviderExtend[BUF_SIZE]={0};
int     iExtendLen=BUF_SIZE;
int     rv=SEF_GetProviderInfo(pchProviderName,&iProviderNameLen, pchProviderCompany,
&iProviderCompanyLen,pchProviderVersion, &iProviderVersionLen, pchProviderExtend,
&iProviderExtendLen);
if(rv! =0)
{
      printf("SEF_GetProviderInfo fail.\n");
      return rv;
}
else
{
      printf("SEF_GetProviderInfo success. \n");
      return SEF_SUCCESS;
}
```

D.2 获取用户印章信息列表

```
USERSEALINFO * gmSealInfoList=NULL;
int  iSealInfoListLen = 0;
int rv = SEF_GetUserSealInfoList(1540886034628,"345678313",9,NULL,0,1,1,
"bxLdRXVbTPd...vsdshFKA",
128,gmSealInfoList,&iSealInfoListLen );
if(rv! =0)
{
    printf("SEF_GetUserSealInfoList fail.\n");
    return rv;
}
gmSealInfoList= (USERSEALINFO * )malloc(iSealInfoListLen );
if(gmSealInfoList==NULL)
{
```

```
    printf("malloc gmSealInfoList error\n");
    return  SEF_MALLOC_ERROR;
}
memset(gmSealInfoList,0,iSealInfoListLen);
rv = SEF_GetUserSealInfoList(1540886034628,"345678313",9,NULL,0,1,1,"bxLkRXVbTPd...
vsdshFKA",128,gmSealInfoList,&iSealInfoListLen );
if(rv! =0)
{
    printf("SEF_GetUserSealInfoList fail.\n");
    return rv;
}
else
{
    printf("SEF_GetUserSealInfoList success. \n");
    return SEF_SUCCESS;
}
```

D.3 文件定位签章

```
POSITIONSEALSIGNINFO ps_SealSignInfo[2];
memset(ps_SealSignInfo,0,sizeof(POSITIONSEALSIGNINFO) * 2);
ps_SealSignInfo[0].sealID = "44010400060556";
ps_SealSignInfo[0].pageIndex = 0;
ps_SealSignInfo[0].pageCount = 3;
ps_SealSignInfo[0].widthRatio = 0.4;
ps_SealSignInfo[0].widthRatio = 0.4;
ps_SealSignInfo[1].sealID = "44010400060557";
ps_SealSignInfo[1].pageIndex = 3;
ps_SealSignInfo[1].pageCount = 4;
ps_SealSignInfo[1].widthRatio = 0.8;
ps_SealSignInfo[1].widthRatio = 0.8;
int iSealSignInfoLen = 2;
unsigned char * puchOutFileData = NULL;
int iOutFileDataLen = 0;
int rv=  SEF_SealSignByPosition(1540886052520,"13454546909.ofd",15,0,0,
"MXBADCGKD.....",2345980,"345678313",9,NUL,0,ps_SealSignInfo,iSealSignInfoLen,1,
"bxLdikRXVbTPdHS...vsdshFKA",128,0,&puchOutFileData,&iOutFileDataLen);
if(rv! =0)
{
    printf("SEF_SealSignByPosition first error\n");
    return  rv;
}
puchOutFileData = (unsigned char * )malloc(iOutFileDataLen);
if(puchOutFIleData==NULL)
```

```
{
    printf("malloc puchOutFileData   error\n");
    return   SEF_MALLOC_ERROR;
}
rv=   SEF_SealSignByPosition(1540886052520,"13454546909.ofd",15,0,0,
"MXBADCGKD.....",2345980,"345678313",9,NUL,0,ps_SealSignInfo,iSealSignInfoLen,1,
"bxLdikRXVbTPdHS...vsdshFKA",128,0,puchOutFileData,&iOutFileDataLen);
if(rv! =0)
{
    printf("SEF_SealSignByPosition fail.\n");
    return rv;
}
else
{
    printf("SEF_SealSignByPosition success. \n");
    return SEF_SUCCESS;
}
```

附 录 E
（规范性）
错误代码定义

错误码定义见表 E.1。

表 E.1 错误代码定义

宏描述	预定义值	说明
SEF_SUCCESS	0	成功
SEF_JSON_PARSE_ERROR	0X08000001	JSON 格式错误
SEF_PARAMETER_ERROR	0X08000002	参数错误
SEF_PARAM_OUTRANG_ERROR	0X80000003	参数超出范围
SEF_DOC_TYPE_ERROR	0X80000004	文件类型不支持
SEF_DOC_FORMAT_ERROR	0X80000005	文件格式不正确
SEF_DOC_OPEN_ERROR	0X80000006	读取文件失败
SEF_DOC_READ_ERROR	0X80000007	加载文件失败
SEF_DOC_LOAD_ERROE	0X80000008	打开文件失败
SEF_REQUEST_TIMEOUT_ERROR	0X80000009	请求时间超时
SEF_REQUEST_VERIFY_ERROR	0X8000000A	请求数据验证失败
SEF_GET_SEALCOUNT_ERROR	0X8000000B	获取用户印章数量异常
SEF_GET_USERSEAL_ERROR	0X8000000C	获取用户印章信息错误
SEF_GET_SEALDATA_ERROR	0X8000000D	获取用户印章数据错误
SEF_GET_USERCERT_ERROR	0X8000000E	获取用户证书信息错误
SEF_GET_CERTDATA_ERROR	0X8000000F	获取用户证书数据错误
SEF_SEAL_IMAGE_ERROR	0X80000010	获取印章图片数据错误
SEF_SEAL_STATUS_ERROR	0X80000011	获取印章状态失败
SEF_SEAL_STATUS_LOCK_ERROR	0X80000012	印章状态被冻结
SEF_MALLOC_ERROR	0X80000013	内存空间分配错误
SEF_DOC_ENC_ERROR	0X80000014	文件被加密不允许签署
SEF_DOC_LOCK_ERROR	0X80000015	文件被锁定不允许签署
SEF_POINT_PARAM_ERROR	0X80000016	坐标参数错误
SEF_PAGE_PARAM_ERROR	0X80000017	页码参数错误
SEF_KEYWORD_EMPTY_ERROR	0X80000018	关键字不能为空
SEF_KEYWORD_FIND_ERROR	0X80000019	未找到关键字
SEF_GETCONTENT_ERROR	0X8000001A	获取被包含内容失败
SEF_DIGEST_ERROR	0X8000001B	计算杂凑值失败
SEF_SEAL_SERVER_ERROR	0X8000001C	签章服务异常

表 E.1　错误代码定义（续）

宏描述	预定义值	说明
SEF_SIGNPAGE_COUNT_ERROR	0X8000001D	单页文件不能加盖骑缝章
SEF_TRANS_IMG_ERROR	0X8000001E	转换图片失败
SEF_DOC_ADDSIGNATURE_ERROR	0X8000001F	文件添加签章数据结构失败
SEF_GETDOC_SIGNATURECOUNT_ERROR	0X80000020	获取文件签名/签章个数失败
SEF_GETDOC_SIGNATUREINFO_ERROR	0X80000021	获取文件签名/签章信息失败
SEF_SET_SIGNATUREINFO_ERROR	0X80000022	添加签名/签章信息失败
SEF_GETDOC_PAGEINFO_ERROR	0X80000023	获取文件页码信息失败
SEF_SET_WATERMARK_ERROR	0X80000024	添加水印失败
SEF_GET_DOCDATA_ERROR	0X80000025	获取数据电文失败
SEF_DELETE_SIGNATURE_ERROR	0X80000026	删除签名/签章信息失败
SEF_DOC_MODIFIED	0X80000027	文件内容被篡改
SEF_DOC_NO_SIGN	0X80000028	未签署文件
SEF_SIGNEDVALUE_DECODE_ERROR	0X80000029	签章数据解码失败
SEF_SIGNEDVALUE_FORMAT_ERROR	0X8000002A	签章数据格式不符合标准
SEF_SIGNVALUE_LENTH_ERROR	0X8000002B	签名值长度不正确
SEF_SIGNVALUE_VERIFY_ERROR	0X8000002C	签章数据签名值验证失败
SEF_PACK_SIGNEDVALUE_ERROR	0X8000002D	组装签章结构体失败
SEF_CERTCHAIN_NULL	0X8000002E	证书链不存在
SEF_DATASIGN_ERROR	0X8000002F	数据签名失败
SEF_SIGNEDMESSAGE_AUTHATTRR_ERROR	0X80000030	组装消息签名认证属性失败
SEF_SIGNEDMESSAGEPACK_ERROR	0X80000031	组装消息签名数据结构失败
SEF_GETSIGNVALUE_NULL	0X80000032	获取到签名/签章数据为空
SEF_SEALDECODE_ERROR	0X80000033	印章数据解码失败
SEF_SEALFORMAT_ERROR	0X80000034	印章数据不符合标准
SEF_SEALVERIFY_ERROR	0X80000035	印章数据签名值验证失败
SEF_SEAL_INVALID	0X80000036	当前时间不在印章有效期
SEF_SIGNTIME_INVALID	0X80000037	签章时间不在证书有效期
SEF_Base64DECODE_ERROR	0X80000038	Base64 解码失败
SEF_Base64ENCODE_ERROR	0X80000039	Base64 编码失败
SEF_MAKERCERTVERIFY_ERROR	0X8000003A	制章人证书签名值验证失败
SEF_SIGNERCERTVERIFY_ERROR	0X8000003B	签章人证书签名值验证失败
SEF_MAKERCERT_INVALID	0X8000003C	当前时间不在制章人证书有效期
SEF_SIGNERCERT_INVALID	0X8000003D	当前时间不在签章人证书有效期
SEF_MAKERCERTROOT_ERROR	0X8000003E	制章人证书根证书验证失败

表 E.1　错误代码定义（续）

宏描述	预定义值	说明
SEF_SIGNERCERTROOT_ERROR	0X8000003F	签章人证书根证书验证失败
SEF_SIGNEDMESSAGEDECODE_ERROR	0X80000040	消息签名数据解码失败
SEF_SIGNEDMESSAGE_ERROR	0X80000041	消息签名验签失败
SEF_SEALSIGNTIME_ERROR	0X80000042	签章时间不在印章有效期
SEF_CERTSIGNTIME_ERROR	0X80000043	签章时间不在证书有效期
SEF_SEALSIGNTIME_ERROR	0X80000044	签章时间格式错误
SEF_SEALCREATETIME_ERROR	0X80000045	印章创建时间格式错误
SEF_SEALSTARTTIME_ERROR	0X80000046	印章起始时间格式错误
SEF_SEALENDTIME_ERROR	0X80000047	印章终止时间格式错误
SEF_TIMESTAMP_FORMAT_ERROR	0X80000048	时间戳格式错误
SEF_TIMESTAMP_SIGN_ERROR	0X80000049	时间戳签名值验证失败
SEF_TIMESTAMP_INVALID	0X8000004A	时间戳有效性验证失败
SEF_ALGORITHM_UNSUPPORTED	0X8000004B	不支持的算法类型
SEF_DIGESTOID_ERROR	0X8000004C	杂凑算法标识错误
SEF_SIGNOID_ERROR	0X8000004D	签名算法标识错误
SEF_MODIFIED_LASTSIGN_ERROR	0X8000004E	文件存在签名后未保护内容
SEF_HASHVERIFY_ERROR	0X8000004F	被保护的内容被篡改
SEF_SERVICE_UNSUPPORT	0X80000050	未提供的功能
SEF_UNKNOWN_ERROR	0X80000051	未知错误

参 考 文 献

[1] GB/T 15843.3—2023 信息技术 安全技术 实体鉴别 第3部分:采用数字签名技术的机制

[2] GM/T 0067—2019 基于数字证书的身份鉴别接口规范

[3] RFC 6749 The OAuth 2.0 Authorization Framework