ICS 35.030
CCS L 80

GM

中华人民共和国密码行业标准

GM/T 0125.3—2022

JSON Web密码应用语法规范 第3部分:数据加密

JavaScript Object Notation Web cryptographic application syntax specification—Part 3:Data encryption

2022-11-20 发布　　2023-06-01 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。

本文件是GM/T 0125《JSON Web密码应用语法规范》的第3部分。GM/T 0125已经发布了以下部分：

——第1部分：算法标识；

——第2部分：数字签名；

——第3部分：数据加密；

——第4部分：密钥。

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：广东省电子商务认证有限公司、智巡密码(上海)检测技术有限公司、格尔软件股份有限公司、北京信安世纪科技股份有限公司、北京数字认证股份有限公司、北京国脉信安科技有限公司、中国科学院信息工程研究所、广东南方通信建设有限公司、郑州迪维勒普科技有限公司、河南省鼎信信息安全等级测评有限公司、上海市数字证书认证中心有限公司。

本文件主要起草人：陈树乐、韩玮、郑强、张永强、袁峰、高能、张庆勇、赵敏、刘义、黄志伟、林少柳、梁宁宁、梁家声、傅大鹏、傅鹏、岳志军、陈宇、王维初。

引　言

《JSON Web 密码应用语法规范》旨在以国产商用密码算法为核心，来保证数据机密性和完整性，适用于 JSON Web 密码应用产品的研发与检测，其他使用 JSON 数据交换格式的安全产品，可参考使用。《JSON Web 密码应用语法规范》由四个部分构成。

——第 1 部分：算法标识。定义了 JSON Web 密码应用的算法标识。

——第 2 部分：数字签名。描述了基于 JSON 数据结构来保护消息内容的数字签名或消息鉴别码的语法规范，并给出了相应的生成和验证流程。

——第 3 部分：数据加密。描述了使用身份鉴别和加密来确保数据的机密性和完整性的技术要求。

——第 4 部分：密钥。定义了密钥的 JSON 数据结构表示方法。

本文件为《JSON Web 密码应用语法规范》的第 3 部分，描述了使用身份鉴别和加密来确保数据的机密性和完整性的技术要求。

JSON Web 密码应用语法规范
第 3 部分:数据加密

1 范围

本文件描述了使用身份鉴别和加密来确保数据的机密性和完整性的技术要求。

本文件适用于 JSON Web 密码应用产品的研发与检测,其他使用 JSON 数据交换格式的安全产品,可参考使用。

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 16263.1 信息技术 ASN.1 编码规则 第 1 部分:基本编码规则(BER)、正则编码规则(CER)和非典型编码规则(DER)规范

GB/T 32905 信息安全技术 SM3 密码杂凑算法

GB/T 32907 信息安全技术 SM4 分组密码算法

GB/T 32915 信息安全技术 二元序列随机性检测方法

GB/T 32918(所有部分) 信息安全技术 SM2 椭圆曲线公钥密码算法

GB/T 35276 信息安全技术 SM2 密码算法使用规范

GB/T 36624 信息技术 安全技术 可鉴别的加密机制

GB/T 38635(所有部分) 信息安全技术 SM9 标识密码算法

GM/T 0080 SM9 密码算法使用规范

GM/T 0125.1 JSON Web 密码应用语法规范 第 1 部分:算法标识

GM/T 0125.4 JSON Web 密码应用语法规范 第 4 部分:密钥

3 术语和定义

下列术语和定义适用于本文件。

3.1

带额外数据的可鉴别加密 authenticated encryption with associated data;AEAD

加密明文,可指定额外的可鉴别数据,并对密文和额外可鉴别数据提供完整性校验的加密。

注:使用该机制的算法也称 AEAD 算法,AEAD 算法接受两个输入,明文和额外可鉴别数据,并产生两个输出,分别是密文和鉴别标志。

3.2

额外可鉴别数据 additional authenticated data;AAD

AEAD 算法的输入,受完整性保护但未加密的数据。

3.3

鉴别标志 authentication tag

AEAD 算法的输出,用于确保密文和额外可鉴别数据的完整性。

3.4

内容加密密钥　content encryption key;CEK

AEAD 算法的对称密钥,用于加密明文以产生密文和鉴别标志。

3.5

JWE 密钥密文　JWE encrypted key

使用密钥加密密钥算法加密 CEK 后获得的密文值。

3.6

JWE 初始化向量　JWE initialization vector

加密算法使用的初始化向量值。

3.7

JWE 额外可鉴别数据　JWE AAD

JWE 数据结构的额外可鉴别数据,只有在 JSON 序列化存在。

3.8

JWE 密文　JWE ciphertext

内容加密算法输出结果的密文部分。

3.9

JWE 标志　JWE authentication tag

内容加密算法输出结果的标志部分,用于确保 JWE 密文和 JWE 额外可鉴别数据的完整性。

3.10

JWE 保护头部　JWE protected header

JSON 对象,包含完整性保护的头部参数,由可鉴别加密操作保护这些头部参数的完整性。

注:这些参数适用于 JWE 的所有接收者。对于 JWE 紧凑序列化,它包含整个 JOSE 头部。对于 JWE JSON 序列化,其是 JOSE 头部的一部分。

3.11

JWE 共享无保护头部　JWE shared unprotected header

JSON 对象,包含不受完整性保护的头部参数。

注:这些参数适用于 JWE 的所有接收者并且不受完整性保护,使用 JWE JSON 序列化时存在。

3.12

JWE 单一接收者无保护的头部　JWE per-recipient unprotected header

JSON 对象,包含不受完整性保护的头部参数,是 JWE 的每个接收者特有的参数。

注:这些头部参数值不受完整性保护,使用 JWE JSON 序列化时存在。

3.13

JWE 紧凑序列化　JWE compact serialization

JWE 使用紧凑的 URL 安全字符串形式表示的序列化。

3.14

JWE JSON 序列化　JWE JSON serialization

JWE 使用 JSON 结构表示的序列化。

注:与 JWE 紧凑序列化不同,JSON 序列化可支持多个接收者。

3.15

密钥加密　key encryption

对密钥进行加密的过程。

注:这里的密钥一般指内容加密密钥 CEK。

4 缩略语

下列缩略语适用于本文件。

CA:证书认证机构(Certification Authority)

CCM:带有密文分组链接消息鉴别码的计数器模式(Counter with Cipher Block Chaining-Message Authentication Code)

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

GCM:伽罗瓦/计数器模式(Galois/Counter Mode)

IV:初始化向量(Initialization Vector)

JOSE:JSON 签名和加密(JSON Object Signing and Encryption)

JSON:JavaScript 对象标记(JavaScript Object Notation)

JWE:JSON Web 数据加密(JSON Web Encryption)

PEM:隐私增强邮件(Privacy Enhanced Mail)

URL:统一资源定位(Uniform Resource Locator)

5 符号

下列符号适用于本文件。

ASCII(X):数据 X 对应的 ASCII 编码

base64url(X):对数据 X 进行 base64url 编码

IV:初始化向量

T:鉴别标志

UTF8(X):数据 X 对应的 UTF-8 编码

‖:字符串连接符号

6 JSON Web 数据加密(JWE)

6.1 概述

JWE 是使用 JSON 方式来表示数据加密的数据结构。

JWE 由 JOSE 头部、JWE 密钥密文、JWE 初始化向量、JWE 额外可鉴别数据、JWE 密文和 JWE 鉴别标志组成。

JWE 加密分为两种序列化格式:紧凑序列化和 JSON 序列化。紧凑序列化只支持一个加密接收者,JSON 序列化可支持一个或多个加密接收者。

JWE 使用 AEAD 算法来确保明文消息的机密性、完整性以及 JWE 保护头部和 JWE 额外可鉴别数据的完整性。首先指定或生成 CEK,使用完整性校验的加密算法加密明文数据和额外可鉴别的数据,再通过密钥加密密钥算法加密 CEK 完成整体加密。

6.2 JOSE 头部

6.2.1 通则

JOSE 头部是一个 JSON 对象,JSON 对象的每个参数描述了 JWE 的头部参数,JOSE 头部中的头部参数名称应唯一。JWE 的 JOSE 头部由 JWE 保护头部、JWE 共享无保护头部和 JWE 单一接收者无

保护头部组成。

JWE 共享无保护头部和 JWE 单一接收者无保护头部只有在 JSON 序列化情况下才存在。本文件使用的 DER 编码规则，应符合 GB/T 16263.1。

JWE 头部参数名称有三类：已定义参数名称、扩展头部参数名称和自定义头部参数名称。

应用解析和处理已定义的头部参数时应符合 6.2.2 要求。

6.2.2 已定义头部参数

6.2.2.1 概述

已定义的头部参数名称见表 1。

表 1 JWE 头部参数

参数值	类型	说明	要求
alg	字符串	算法	必选
enc	字符串	加密算法	必选
jku	字符串	密钥资源的 URI	可选
jwk	JSON 对象	JSON Web 密钥	可选
kid	字符串	密钥 ID	可选
x5u	字符串	证书 URL	可选
x5c	数组	证书链	可选
x5t＃sm3	字符串	证书 SM3 杂凑值	可选
zip	字符串	明文压缩算法	可选
typ	字符串	整个 JWE 的媒体类型	可选
cty	字符串	JWE 有效载荷媒体类型	可选
crit	数组	应被处理的关键头部参数列表	可选

6.2.2.2 “alg”(算法)头部参数

“alg”头部参数是一个字符串，用于标识密钥加密密钥算法。该头部参数必选，取值应符合 GM/T 0125.1，如果“alg”值是不支持的算法，或者接收者没有可与该算法一起使用的密钥，则加密的内容不可用。

“alg”值是 ASCII 字符串，区分大小写。如果使用密钥加密密钥算法，该头部参数必选。

6.2.2.3 “enc”(加密算法)头部参数

“enc”头部参数用于标识使用的内容加密算法。该头部参数取值应符合 GM/T 0125.1。如果“enc”的值是不支持的算法，则加密的内容不可用。

“enc”值是 ASCII 字符串，区分大小写，该头部参数必选。

6.2.2.4 “jku”(JWK Set URL)头部参数

“jku”头部参数是一个 URI 的字符串，它引用一组 JSON 结构的公钥，其中每个公钥应编码为 JWK 集合，可用于确定解密 JWE 所需的私钥。该头部参数可选，应设置为保护头部参数。

6.2.2.5 “jwk”(JSON Web 密钥)头部参数

“jwk”头部参数是一个 JSON 对象,用于标识 JWE 加密的公钥或确定解密的私钥。该头部参数可选,取值应符合 GM/T 0125.4。

6.2.2.6 “kid”(密钥 ID)头部参数

“kid”头部参数用于标识 JWE 加密的公钥或标识解密的私钥。此头部参数可选,其值以字符串形式表示,区分大小写。当与 JWK 一起使用时,“kid”值用于匹配 JWK 的“kid”参数值。

6.2.2.7 “x5u”(证书 URL)头部参数

“x5u”头部参数用于标识与密钥加密密钥相关的加密证书 URL,其值是一个 URI 形式的字符串,该资源应提供符合 PEM 编码形式的证书或证书链,每个证书使用以下分隔符隔开:

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

第一个证书应为 JWE 密钥加密密钥相关的加密证书。该头部参数可选,应设置为保护头部参数。

6.2.2.8 “x5c”(证书链)头部参数

“x5c”头部参数用于标识密钥加密密钥相关的加密证书或证书链,其值是一个 JSON 字符串数组,数组中的元素是数字证书 DER 编码的 base64 编码字符串。数组第一个元素应是 JWE 密钥加密密钥相关的加密证书,后面的元素依次是前一个元素颁发证书的 CA 所持有的证书。该头部参数可选。

6.2.2.9 “x5t＃sm3”(证书 SM3 杂凑值)头部参数

“x5t＃sm3”头部参数是一个字符串,用于标识对内容加密密钥加密的证书 SM3 指纹,可用来匹配密钥。该值生成过程:使用数字证书的 DER 编码进行 SM3 杂凑后再进行 base64url 编码。SM3 算法的计算方法和步骤应符合 GB/T 32905,该头部参数可选。

6.2.2.10 “zip”(压缩)头部参数

“zip”头部参数是一个字符串,用于标识明文压缩算法。当该参数存在时,表示明文在加密前经过压缩编码处理,此头部参数可选,应放在保护头部。参数值见表 2。

表 2 “zip”参数值

“zip”参数值	说明
DEF	DEFLATE 压缩算法

6.2.2.11 “typ”(类型)头部参数

“typ”头部参数是一个字符串,用于标识整个 JWE 的媒体类型,该头部参数可选。

6.2.2.12 “cty”(内容类型)头部参数

“cty”头部参数是一个字符串,用于标识 JWE 明文的媒体类型,该头部参数可选。

6.2.2.13 “crit”(关键)头部参数

“crit”头部参数用于标识应解析和处理的头部参数。它是一个 JSON 数组,列出了 JOSE 头部中应

处理的头部参数名称，该头部参数可选。

6.2.3 扩展的头部参数

本文件可扩展定义新的头部参数。为了防止冲突，禁止新定义的头部参数名称与已定义的头部名称重复。应谨慎引入新的头部参数，避免导致 JWE 之间无法互操作。

6.2.4 自定义头部参数

JWE 的应用双方可约定互操作的头部参数名称。与已定义参数名称不同，不同应用的自定义头部参数名称可能会发生冲突，应谨慎使用。

7 JWE 密钥加密密钥算法

7.1 概述

JWE 密钥加密密钥算法是用于对 CEK 进行加密或解密的算法。该算法通过 JOSE 头部的“alg”头部参数来标识，标识取值应符合 GM/T 0125.1。

本文件规定了 SM2 公钥加密和 SM9 加密两种密钥加密密钥算法的实现要求。

7.2 SM2 公钥加密算法

本节描述了 SM2 公钥加密算法的相关要求。

SM2 公钥加密算法的加解密计算方法应符合 GB/T 32918，指定杂凑算法为 SM3 算法，SM3 杂凑算法的运算方法和步骤应符合 GB/T 32905。

该算法加密结果的输出数据格式应符合 GB/T 35276。JWE 密钥密文的值为加密结果的输出。

7.3 SM9 加密算法

本节描述了 SM9 加密算法的相关要求。

SM9 加密算法的加密和解密过程应符合 GB/T 38635。指定杂凑算法为 SM3 算法，SM3 杂凑算法的运算方法和步骤应符合 GB/T 32905。指定消息认证码函数的密钥长度为 256 位。

该算法加密结果的输出数据格式应符合 GM/T 0080。JWE 密钥密文的值为加密结果的输出。

8 JWE 内容加密算法

8.1 概述

内容加密算法使用 AEAD 算法来加密，是提供给 JWE 使用的完整性校验加密算法，可用来加密或解密 JWE 的消息数据和保证密文和额外指定可鉴别数据的完整性。该算法通过 JOSE 头部的“enc”头部参数来标识。

内容加密算法的密钥使用密钥加密密钥算法加密。内容加密算法加密时会输出两个参数，一个是密文，一个是标志，分别对应 JWE 密文和 JWE 标志。

本文件定义了内容加密算法使用指定可鉴别的加密机制算法的实现要求。

8.2 可鉴别的加密机制算法

内容加密算法使用可鉴别的加密机制算法时加解密运算方法和步骤应符合 GB/T 36624。SM4 分组密码运算方法和步骤应符合 GB/T 32907。

a) 如果是可鉴别加密机制 CCM 算法，JWE 密文的值为 CCM 算法加密数据输出的 $C_1 \| C_2 \| \cdots \| C_m$ 部分。JWE 标志的值为加密数据输出的 U 部分。

b) 如果是可鉴别加密机制 GCM 算法，JWE 密文的值为 GCM 算法加密数据输出的 $C_1 \| C_2 \| \cdots \| C_m$ 部分。JWE 标志的值为加密数据输出的 T 部分。

8.3 参数要求

本文件定义了内容加密算法的参数要求，这里输入参数 K 对应内容加密算法的 CEK，输入参数 IV 对应算法的开始变量，输入参数 A 对应额外可鉴别数据，输出参数 T 对应 JWE 标志。参数要求见表 3。

表 3 内容加密算法的参数要求

算法标识	输入参数的长度要求				输出参数的长度要求
	M	K	IV	A	T
SGD_SM4_CCM	小于 2^{59} 比特	128 比特	64 比特	小于 2^{67} 比特	128 比特
SGD_SM4_GCM	小于或等于 $2^{39}-256$ 比特	128 比特	96 比特	小于或等于 $2^{64}-1$ 比特	128 比特

9 JWE 加密和解密

9.1 概述

本章节描述了 JWE 的加密和解密过程。该部分使用国家密码管理局批准的随机数发生器，生成的随机序列应符合 GB/T 32915。

9.2 消息加密过程

设待加密的明文数据为 M，消息加密过程如下。

a) 根据应用设置的保护头部参数，生成 JWE 保护头部的 JSON 结构。

b) 根据应用设置的共享无保护头部参数，生成 JWE 共享无保护头部的 JSON 结构。

c) 如果是 JSON 序列化，根据应用设置每个接收者特有的无保护头部参数，对每个接收者生成单一接收者无保护头部的 JSON 结构。

d) 根据内容加密密钥的算法，在 JOSE 头部设置“enc”头部参数值。

e) 根据密钥加密密钥的算法，在 JOSE 头部设置“alg”头部参数值。

f) 根据需要在 JOSE 头部设置用户密钥或相关标识信息。

g) 根据内容加密算法要求的密钥长度，使用随机数发生器生成随机数，将随机数设置为内容加密密钥 CEK。使用密钥加密密钥算法加密 CEK，并将加密输出结果作为 JWE 密钥密文。

h) 如果使用 JSON 序列化，对每个接收者重复此过程[步骤 e)～g)]。

i) 根据内容加密算法要求的初始化向量长度，使用随机数发生器生成随机数，将随机数设置为 JWE 初始化向量 IV。

j) 设置 JWE 保护头部编码为 base64url(UTF8(JWE 保护头部))，如果保护头部不存在，JWE 保护头部编码设置为空字符串。

k) 如果使用 JSON 序列化并且应用设置了额外可鉴别数据值 A^1，将内容加密算法可鉴别数据 A 设置为：ASCII(JWE 保护头部编码) ‖ '.' ‖ base64url(应用设置的额外可鉴别数据 A^1)，否则将内容加密算法的额外可鉴别数据 A 设置为：ASCII(JWE 保护头部编码)。

l) 如果设置“zip”头部参数，根据压缩算法压缩明文消息 M 生成压缩后的数据 M^Z。设置明文消息 M 为 M^Z。

m) 根据内容加密算法使用 CEK，JWE 初始化向量 IV 和额外可鉴别数据 A 对明文消息 M 进行加密，输出 JWE 密文和 JWE 标志。

n) 输出序列化结果：

1) 紧凑序列化结果是以下级联的字符串 base64url(UTF8(JWE 保护头部)) ‖ '.' ‖ base64url(JWE 密钥密文) ‖ '.' ‖ base64url(JWE 初始化向量) ‖ '.' ‖ base64url(JWE 密文) ‖ '.' ‖ base64url(JWE 标志)；

2) JSON 序列化结果见 12.3。

消息加密的示例见附录 A。

9.3 消息解密过程

消息解密过程与加密过程相反。如果其中任何步骤失败，则解密失败。

当有多个接收者时，由应用程序决定哪些接收者的加密内容在成功解密后接受 JWE。在某些情况下，只有当所有接收者的加密内容都能被成功解密和验证时，JWE 才是有效的，否则被视为无效。在其他情况下，只要成功解密和验证单个接收者的加密内容即可接受 JWE。

为了得到解密明文 M，解密过程如下。

a) 如果是 JWE 紧凑序列化，通过'.'分割符解析顺序得到 JWE 保护头部、JWE 密钥密文、JWE 初始化向量、JWE 密文和 JWE 标志，如果解析失败，则解密失败。

b) 如果是 JWE JSON 序列化，按照 12.3 的描述解析。

c) 检验使用 base64url 解码 JWE 保护头部、JWE 密钥密文、JWE 初始化向量、JWE 密文、JWE 标志和 JWE 额外可鉴别数据是否成功，如果解码失败，则解密失败。

d) 检验 JWE 保护头部是否符合 UTF-8 编码，如果检验失败，则解密失败。

e) 如果使用紧凑序列化，将 JWE 保护头部作为 JOSE 头部。如果使用 JWE JSON 序列化，将 JWE 保护头部、JWE 共享无保护头部和对应的 JWE 单一接收者无保护头部参数的并集作为每一个接收者的 JOSE 头部，所有这些头部应为完全有效的 JSON 对象。JOSE 头部参数名称不能重复，如果存在重复头部参数，则解密失败。

f) 若 JOSE 头部使用“crit”头部参数值，根据参数值定义做相应处理。

g) 从 JOSE 头部获取“enc”头部参数，确定内容加密算法。

h) 从 JOSE 头部获取“alg”头部参数，确定密钥加密密钥算法。

i) 从 JOSE 头部获取用户相关密钥和标识信息。

j) 根据密钥加密密钥算法解密 JWE 密钥密文得到 CEK。CEK 的长度应符合内容加密算法所要求的长度。当存在多个接收者时，每个接收者解密自己对应的 JWE 密钥密文。

k) 如果正在使用 JWE JSON 序列化，重复此过程[步骤 h)～j)]。

l) 计算保护头部编码值为 base64url[UTF8(JWE 保护头部)]。如果 JWE 保护头部不存在，设置为空字符串。

m) 将内容加密算法的额外可鉴别数据输入设置为：ASCII(保护头部编码值)。如果使用 JSON 序列化并且存在 JWE 额外可鉴别数据值，算法的额外可鉴别数据输入设置为：ASCII(保护头部编码值) ‖ '.' ‖ base64url(JWE 额外可鉴别数据)。

n) 根据内容加密算法，使用 CEK、JWE 初始化向量、JWE 额外可鉴别数据和 JWE 标志为算法输入参数解密 JWE 密文，输出解密的明文 M 并验证 JWE 标志，如果 JWE 标志验证不正确，则

解密失败。

o) 如果保护头部存在“zip”,根据压缩算法对解密的明文 M 进行解压缩,生成解压缩的明文。

p) 当没有成功解密的接收者时,JWE 应被视为无效。

10 字符串比较规则

处理 JWE 时,应将已知字符串与 JSON 对象中的头部参数名称和头部参数值进行比较并严格区分大小,该比较规则适用于一般情况下对所有 JSON 字符串的比较。当头部参数的定义明确指出要为该头部参数值使用不同的比较规则时,应遵循头部参数的具体定义。

11 密钥身份标识

本文件被识别的密钥是 JWE 密钥加密密钥算法所使用的公钥。JWE 的接收者应能确定用于解密的密钥身份标识。密钥应使用 6.2.2 中描述的头部参数方法标识。

密钥信息可通过头部参数“jku”“jwk”“kid”“x5u”“x5c”和“x5t＃sm3”来识别。

若应用程序未使用其他方法或约定来确定所使用的密钥,消息创建者应在头部参数中包含足够的信息,以标识所使用的密钥。如果无法识别所使用的密钥,则解密失败。

12 序列化

12.1 概述

JWE 有两种数据表示形式,分别是紧凑序列化和 JSON 序列化,紧凑序列化只支持一个加密接收者,JSON 序列化可支持一个或多个加密接收者。

12.2 紧凑型序列化

JWE 紧凑序列化将加密内容表示为紧凑的 URL 安全字符串。其一般形式如下:

base64url(UTF8(JWE 保护头部)) ‖ '.' ‖

base64url(JWE 密钥密文) ‖ '.' ‖

base64url(JWE 初始化向量) ‖ '.' ‖

base64url(JWE 密文) ‖ '.' ‖

base64url(JWE 标志)

JWE 紧凑序列化仅支持一个接收者,不使用 JWE 共享无保护头部或 JWE 单一接收者无保护头部。在这种情况下,JOSE 头部和 JWE 保护头部是相同的。

12.3 JSON 序列化

JWE JSON 序列化将加密内容表示为 JSON 对象。

JWE JSON 序列化有两种语法,分别是通用语法和扁平化语法。

a) 通用 JSON 序列化语法:

通用 JSON 序列化是一个 JSON 对象,由“protected”“unprotected”“iv”“aad”“tag”“ciphertext”和“recipients”元素组成,通用 JSON 序列化组成参数见表 4,“recipients”元素组成参数见表 5。

表 4　通用 JSON 序列化组成参数

参数值	类型	内容说明	要求
protected	字符串	JWE 保护头部的 JSON 结构字符串，它是先使用 UTF-8 编码，再进行 base64url 编码后获得的字符串。如果设置了保护头，该参数应存在	可选
unprotected	JSON 对象	JWE 的共享无保护头部，是一个 JSON 对象	可选
iv	字符串	JWE 加密使用的初始化向量的 base64url 编码	必选
aad	字符串	JWE 加密使用的额外可鉴别数据 base64url 编码。如果应用没有设置，该参数不存在	可选
tag	字符串	JWE 加密后鉴别标志的 base64url 编码	必选
ciphertext	字符串	JWE 加密密文的 base64url 编码	必选
recipients	数组	JWE 的加密接收者数组，数组每个元素代表一个接收者对象	必选

表 5　recipients 数组每个元素组成参数

参数值	类型	内容说明	要求
header	JSON 对象	JWE 的接收者无保护头部，是一个 JSON 对象	必选
encrypted_key	字符串	JWE 的加密密钥的 base64url 编码	必选

在序列化结构中“header”“protected”和“unprotected”应至少存在一个参数，以确保设置了“alg”和“enc”头部参数值。JWE 保护头部中的头部参数和 JWE 共享无保护头部参数值在所有接收者之间共享。“protected”“unprotected”“header”头部参数的并集构成了 JOSE 头部。并集中的元素名称应不重复。所有接收者使用相同的 JWE 保护头部、JWE 初始化向量、JWE 密文和 JWE 标志。

通用 JSON 序列化的语法如下：

```
{
    "protected":"〈integrity-protected shared header contents〉",
    "unprotected":〈non-integrity-protected shared header contents〉,
    "recipients":[
     {
         "header":〈per-recipient unprotected header 1 contents〉,
         "encrypted_key":"〈encrypted key 1 contents〉"
     },
    ...
     {
         "header":〈per-recipient unprotected header N contents〉,
         "encrypted_key":"〈encrypted key N contents〉"
     }],
    "aad":"〈additional authenticated data contents〉",
    "iv":"〈initialization vector contents〉",
```

```
    "ciphertext":"〈ciphertext contents〉",
    "tag":"〈authentication tag contents〉"
}
```

b) 扁平化 JSON 序列化语法：

扁平化 JSON 序列化语法基于通用语法，用于表示只有一个 JWE 加密接收者。其语法如下：

```
{
    "protected":"〈integrity-protected header contents〉",
    "unprotected":〈non-integrity-protected header contents〉,
    "header":〈more non-integrity-protected header contents〉,
    "encrypted_key":"〈encrypted key contents〉",
    "aad":"〈additional authenticated data contents〉",
    "iv":"〈initialization vector contents〉",
    "ciphertext":"〈ciphertext contents〉",
    "tag":"〈authentication tag contents〉"
}
```

附 录 A
（资料性）
JWE 示例

A.1 综述

本附录中，所有涉及的整数都使用大端格式。

本附录中，明文采用 UTF-8 编码。

本附录中，随机密钥使用 16 进制值表示。

本附录中，为了验证示例的正确性，给出接收者的 SM2 私钥的原文，并使用 base64url 编码表示。在实际运用中，接收者的密钥由密码模块生成并存储在密码模块。

A.2 基于“SGD_SM2_3”和“SGD_SM4_CCM”的 JWE 序列化示例

A.2.1 概述

该示例展示使用“SGD_SM2_3”算法来加密密钥，“x5t＃sm3”来标识证书，“SGD_SM4_CCM”来加密消息内容的 JWE 生成过程和输出结果。

该示例没有使用额外可鉴别数据（紧凑序列化不支持额外可鉴别数据的输入）。

A.2.2 输入参数

输入参数包括以下内容。

a) 待加密数据 M：message encryption
16 进制表示为：6D65737361676520656E6372797074696F6E

b) 加密密钥算法：“SGD_SM2_3”

c) 内容加密算法：“SGD_SM4_CCM”

d) 接收者的 SM2 密钥：
使用如下 JWK 格式来表示：

```
{
    "kty":"EC",
    "crv":"sm2p256v1",
    "x":"yJxuyluqb24nw_VpAMFMP0ZtryQexPUwYhKt79oe0Jw",
    "y":"dEK3Qy1crFnVjlNlac-Tx_CnkJoXhyBB8Y1Jm-FzKKE"
}
```

“x”和“y”值是用椭圆曲线点坐标的字节串进行 base64url 编码。

接收者的私钥 d：“sNPitpyTsL_kJAG55gVlEUu-9MKR_saPkcsvU15RQMY”。

e) SM2 证书：
使用 SM2 密钥的证书 base64 编码如下：

```
MIIBqjCCAU6gAwIBAgIUMEzO577ZywenFSJMKtKRFFgyNVEwDAYIKoEcz1UBg3UFA
DAlMQswCQYDVQQGEwJDTjEWMBQGA1UEAwwNU20yX1Jvb3RfVGVzdDAeFw0yMD
AyMjUwMzQxMjVaFw0zNTAyMjUwMzQxMjVaMB8xCzAJBgNVBAYTAkNOMRAwDg
YDVQQDDAdTTTIgRW5jMFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAEyJxuyluqb24n
w/VpAMFMP0ZtryQexPUwYhKt79oe0Jx0QrdDLVysWdWOU2Vpz5PH8KeQmheHIEHxj
```

Umb4XMooaNgMF4wHwYDVR0jBBgwFoAUqgvWcEhiceHkExasPzhPHS0nevkwHQYDVR0OBBYEFPmJEyELhHTKVBtn7CEiLAGjE/MiMAwGA1UdEwEB/wQCMAAwDgYDVR0PAQH/BAQDAgQwMAwGCCqBHM9VAYN1BQADSAAwRQIhANJueZ6FVPM1+3CuvN9xs7e1TVBJTH5XIfSdHUtw/yOtAiA74TgtAAyHNKQ2q3+q1foC1ckVZYpHZ+fs9gcGatYfhA==

A.2.3 计算过程

计算过程包括以下内容。

a) 使用 SM3 算法计算 SM2 证书 DER 编码的杂凑值，将杂凑值进行 base64url 编码结果为：
AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A

b) 根据加密密钥算法“SGD_SM2_3”和内容加密算法“SGD_SM4_CCM”生成保护头部的 JSON 结构如下：

```
{
    "alg": "SGD_SM2_3",
    "enc": "SGD_SM4_CCM",
    "x5t#sm3": "AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A"
}
```

JSON 结构的字符串使用 UTF-8 编码，进行 base64url 编码，生成的保护头部编码为：
eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0NDTSIsIng1dCNzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfYjlBIn0

c) 按照“SGD_SM4_CCM”算法的密钥要求，产生 16 字节的随机数 CEK，其使用 16 进制表示为：
DDCCCB861885EFB57ECCA05ED862AFC7

d) 根据 SM2 密钥，按照 SM2 加密算法要求加密 CEK，将加密结果进行 base64url 编码，生成密钥密文编码为：
MHgCIEPi3dA4MJD6CEm75iNVP2rupd7MXCcLU35yxPDR2kfvAiAW9aahxfnY-mJaQWoUopqzlsRjPJsXj6-w6BMU2MRapAQgGdx1XN9Np0jLiW9FvPrLbUQ-xY-kLvA-pkfmHB8v9UMEEM0IBk0YapOiAUCX2K8eU-o

e) 按照“SGD_SM4_CCM”算法的初始化向量参数要求，产生 64 位的随机数 IV，其值使用 16 进制表示为：3C2E4B85ECA9C457 进行 base64url 编码后，初始化向量编码为：PC5LheypxFc

f) 设置额外可鉴别数据为保护头部编码，A 的值为：
eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0NDTSIsIng1dCNzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfYjlBIn0

g) 按照 SM4 算法 CCM 可鉴别加密机制，使用 IV，A 和 CEK 对数据 M 进行加密。
将加密密文和鉴别标志进行 base64url 编码后，
JWE 密文编码为：_UK2_3k2Q3P0yxqxJMMBD6_G
JWE 标志编码为：1xyaAMLVoSioY1SIltZH8g

A.2.4 输出结果

根据上面计算的保护头部编码、密钥密文编码、初始化向量编码、JWE 密文编码、JWE 标志编码，输出紧凑序列化和 JSON 序列化结果分别如下。

a) 紧凑序列化：
eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0NDTSIsIng1dCNzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfYjlBIn0.MHgCIEP

i3dA4MJD6CEm75iNVP2rupd7MXCcLU35yxPDR2kfvAiAW9aahxfnY-mJaQWoUopqzlsRj
PJsXj6-w6BMU2MRapAQgGdx1XN9Np0jLiW9FvPrLbUQ-xY-kLvA-pkfmHB8v9UMEEM0
IBk0YapOiAUCX2K8eU-o.PC5LheypxFc._UK2_3k2Q3P0yxqxJMMBD6_G.1xyaAMLVoSi
oY1SIltZH8g

b) JSON 序列化：

```
{
    "protected":"eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0NDTSIsIng1dC
NzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfY
jlBIn0",
    "encrypted_key":"MHgCIEPi3dA4MJD6CEm75iNVP2rupd7MXCcLU35yxPDR2kfvAi
AW9aahxfnY-mJaQWoUopqzlsRjPJsXj6-w6BMU2MRapAQgGdx1XN9Np0jLiW9FvPrLbU
Q-xY-kLvA-pkfmHB8v9UMEEM0IBk0YapOiAUCX2K8eU-o",
    "iv":"PC5LheypxFc",
    "ciphertext":"_UK2_3k2Q3P0yxqxJMMBD6_G",
    "tag":"1xyaAMLVoSioY1SIltZH8g"
}
```

A.3 基于“SGD_SM2_3”和“SGD_SM4_GCM”的 JWE 序列化示例

A.3.1 概述

该示例展示使用“SGD_SM2_3”算法来加密密钥，“x5t＃sm3”来标识证书，“SGD_SM4_GCM”来加密消息内容的 JWE 生成过程和输出结果。

A.3.2 输入参数

输入参数包括以下内容。

a) 待加密数据 M：message encryption
16 进制表示为：6D65737361676520656E6372797074696F6E

b) 加密密钥算法：“SGD_SM2_3”

c) 内容加密算法：“SGD_SM4_GCM”

d) 接收者的 SM2 密钥：
使用如下 JWK 格式来表示：

```
{
    "kty":"EC",
    "crv":"sm2p256v1",
    "x":"yJxuyluqb24nw_VpAMFMP0ZtryQexPUwYhKt79oe0Jw",
    "y":"dEK3Qy1crFnVjlNlac-Tx_CnkJoXhyBB8Y1Jm-FzKKE"
}
```

“x”和“y”值是用椭圆曲线点坐标的字节串进行 base64url 编码。
接收者的私钥 d：“sNPitpyTsL_kJAG55gVlEUu-9MKR_saPkcsvU15RQMY”。

e) SM2 证书：
使用 SM2 密钥的证书 base64 编码如下：
MIIBqjCCAU6gAwIBAgIUMEzO577ywenFSJMKtKRFFgyNVEwDAYIKoEcz1UBg3UFA
DAlMQswCQYDVQQGEwJDTjEWMBQGA1UEAwwNU20yX1Jvb3RfVGVzdDAeFw0yMD

AyMjUwMzQxMjVaFw0zNTAyMjUwMzQxMjVaMB8xCzAJBgNVBAYTAkNOMRAwDgYDVQQDDAdTTTIgRW5jMFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAEyJxuyluqb24nw/VpAMFMP0ZtryQexPUwYhKt79oe0Jx0QrdDLVysWdWOU2Vpz5PH8KeQmheHIEHxjUmb4XMooaNgMF4wHwYDVR0jBBgwFoAUqgvWcEhiceHkExasPzhPHS0nevkwHQYDVR0OBBYEFPmJEyELhHTKVBtn7CEiLAGjE/MiMAwGA1UdEwEB/wQCMAAwDgYDVR0PAQH/BAQDAgQwMAwGCCqBHM9VAYN1BQADSAAwRQIhANJueZ6FVPM1+3CuvN9xs7e1TVBJTH5XIfSdHUtw/yOtAiA74TgtAAyHNKQ2q3+q1foC1ckVZYpHZ+fs9gcGatYfhA==

A.3.3 计算过程

计算过程包括以下内容。

a) 使用 SM3 算法计算 SM2 证书 DER 编码的杂凑值，将杂凑值进行 base64url 编码结果为：AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A；

b) 根据加密密钥算法“SGD_SM2_3”和内容加密算法“SGD_SM4_GCM”生成保护头部的 JSON 结构如下：

```
{
    "alg":"SGD_SM2_3",
    "enc":"SGD_SM4_GCM",
    "x5t#sm3":"AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A"
}
```

JSON 结构的字符串使用 UTF-8 编码，进行 base64url 编码，生成的保护头部编码为：
eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0dDTSIsIng1dCNzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfYjlBIn0

c) 按照“SGD_SM4_GCM”算法的密钥要求，产生 16 字节的随机数 CEK，其使用 16 进制表示为：029C34C2F114E59F8AB5C7DF12B79B42

d) 根据 SM2 密钥，按照 SM2 加密算法要求加密 CEK，将加密结果进行 base64url 编码，生成密钥密文编码为：
MHgCIBuBvS07jcCnMOZg4mWfBhah4oEe3hiYYNb7B3xRtEbjAiAwbMORPg1uHTK-6uBvF398gKNbJoR9kkG8wmBSYGcKKQQgSs2ytBO_DsUZ1DveL7sSCnFCatLNFYGD1N8GVBQ4VdkEEI05kUtyQeGl-6XE6D_rAr4

e) 按照“SGD_SM4_GCM”算法的初始化向量参数要求，产生 96 位的随机数 IV，其值使用 16 进制表示为：405E24AF23CBE84CB575A285 进行 base64url 编码后，初始化向量编码为：
QF4kryPL6Ey1daKF

f) 设置额外可鉴别数据为保护头部编码，A 的值为：
eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0dDTSIsIng1dCNzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfYjlBIn0

g) 按照 SM4 算法 GCM 可鉴别加密机制，使用 IV，A 和 CEK 对数据 M 进行加密。
将加密密文和鉴别标志进行 base64url 编码后，
JWE 密文编码为：ex9LBeJQDo5xIWmFhzdJh7yy
JWE 标志编码为：Poor7hT6qAdlVWUCVJuzQQ

A.3.4 输出结果

根据上面计算的保护头部编码、密钥密文编码、初始化向量编码、JWE 密文编码、JWE 标志编码，

输出紧凑序列化和 JSON 序列化结果分别如下。

a) 紧凑序列化：

eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0dDTSIsIng1dCNzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfYjlBIn0.MHgCIBuBvS07jcCnMOZg4mWfBhah4oEe3hiYYNb7B3xRtEbjAiAwbMORPg1uHTK-6uBvF398gKNbJoR9kkG8wmBSYGcKKQQgSs2ytBO_DsUZ1DveL7sSCnFCatLNFYGD1N8GVBQ4VdkEEI05kUtyQeGl-6XE6D_rAr4.QF4kryPL6Ey1daKF.ex9LBeJQDo5xIWmFhzdJh7yy.Poor7hT6qAdlVWUCVJuzQQ

b) JSON 序列化：

{
"protected":"eyJhbGciOiJTR0RfU00yXzMiLCJlbmMiOiJTR0RfU000X0dDTSIsIng1dCNzbTMiOiJBblJqNzR5U0RkNUM0OGZVdVNud0NZVXU3QTVFb0NmaWdGZWVwb3pfYjlBIn0",
"encrypted_key":"MHgCIBuBvS07jcCnMOZg4mWfBhah4oEe3hiYYNb7B3xRtEbjAiAwbMORPg1uHTK-6uBvF398gKNbJoR9kkG8wmBSYGcKKQQgSs2ytBO_DsUZ1DveL7sSCnFCatLNFYGD1N8GVBQ4VdkEEI05kUtyQeGl-6XE6D_rAr4",
"iv":"QF4kryPL6Ey1daKF",
"ciphertext":"ex9LBeJQDo5xIWmFhzdJh7yy",
"tag":"Poor7hT6qAdlVWUCVJuzQQ"
}

A.4 基于“SGD_SM2_3”和“SGD_SM4_CCM”的 JWE 多个接收者示例

A.4.1 概述

该示例展示了将同一个消息加密后发给两个独立的接收者，使用 SM2 加密算法来分别加密两个接收者各自使用的 SM4 对称密钥，使用“x5t＃sm3”来分别标识两个接收者的证书。在计算 SM2 加密完成后，使用 SM4 算法 CCM 可鉴别机制加密消息内容。

A.4.2 输入参数

输入参数包括以下内容。

a) 待加密数据 M：message encryption
16 进制表示为：6D65737361676520656E6372797074696F6E

b) 额外可鉴别数据 AAD：aad data
16 进制表示为：6161642064617461

c) 加密密钥算法："SGD_SM2_3"

d) 内容加密算法："SGD_SM4_CCM"

e) 第一个接收者的 SM2 密钥：
使用如下 JWK 格式来表示：

{
"kty":"EC",
"crv":"sm2p256v1",
"use":"enc",
"x":"yJxuyluqb24nw_VpAMFMP0ZtryQexPUwYhKt79oe0Jw",

"y":"dEK3Qy1crFnVjlNlac-Tx_CnkJoXhyBB8Y1Jm-FzKKE"
}

“x”和“y”值是用椭圆曲线点坐标的字节串进行 base64url 编码。

第一个接收者的私钥 d 使用 base64url 编码为：

sNPitpyTsL_kJAG55gVlEUu-9MKR_saPkcsvU15RQMY

f) 第一个接收者 SM2 证书：

使用 SM2 密钥的证书 base64 编码如下：

MIIBqjCCAU6gAwIBAgIUMEzO577ZywenFSJMKtKRFFgyNVEwDAYIKoEcz1UBg3UFA
DAlMQswCQYDVQQGEwJDTjEWMBQGA1UEAwwNU20yX1Jvb3RfVGVzdDAeFw0yMD
AyMjUwMzQxMjVaFw0zNTAyMjUwMzQxMjVaMB8xCzAJBgNVBAYTAkNOMRAwDg
YDVQQDDAdTTTIgRW5jMFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAEyJxuyluqb24n
w/VpAMFMP0ZtryQexPUwYhKt79oe0Jx0QrdDLVysWdWOU2Vpz5PH8KeQmheHIEHxj
Umb4XMooaNgMF4wHwYDVR0jBBgwFoAUqgvWcEhiceHkExasPzhPHS0nevkwHQYDV
R0OBBYEFPmJEyELhHTKVBtn7CEiLAGjE/MiMAwGA1UdEwEB/wQCMAAwDgYDVR
0PAQH/BAQDAgQwMAwGCCqBHM9VAYN1BQADSAAwRQIhANJueZ6FVPM1＋3Cuv
N9xs7e1TVBJTH5XIfSdHUtw/yOtAiA74TgtAAyHNKQ2q3＋q1foC1ckVZYpHZ＋fs9gcG
atYfhA＝＝

g) 第二个接收者的 SM2 密钥：

使用如下 JWK 格式来表示：

{
"kty":"EC",
"crv":"sm2p256v1",
"use":"enc",
"x":"EVIjXSs9YhjUsHcGOuyVyYbIF8wFQ2wCp9SIBn529zY",
"y":"2fANOVwJQkGKpGkZRNQfi__Jxlpf2lBs3UDo3MQ5ZXQ"
}

“x”和“y”值是用椭圆曲线点坐标的字节串进行 base64url 编码。

第二个接收者的私钥 dd 使用 base64url 编码为：

nBy-Ofbm-O0nTdQbPvGyiWY3MR3JVjXzziuOJ4T8gRo

h) 第二个接收者的 SM2 证书：

使用 SM2 密钥的证书 base64 编码如下：

MIIBqzCCAU＋gAwIBAgIUQrP4x64Mf＋Jag4RUbjlt2NGU8d8wDAYIKoEcz1UBg3UFAD
AlMQswCQYDVQQGEwJDTjEWMBQGA1UEAwwNU20yX1Jvb3RfVGVzdDAeFw0yMD
AzMDIxMjA3NTJaFw0zNTAzMDIxMjA3NTJaMCAxCzAJBgNVBAYTAkNOMREwDwY
DVQQDDAhTTTIgRW5jMjBZMBMGByqGSM49AgEGCCqBHM9VAYItA0IABBFSI10rP
WIY1LB3BjrslcmGyBfMBUNsAqfUiAZ＋dvc22fANOVwJQkGKpGkZRNQfi//Jxlpf2lBs3U
Do3MQ5ZXSjYDBeMB8GA1UdIwQYMBaAFKoL1nBIYnHh5BMWrD84Tx0tJ3r5MB0GA1
UdDgQWBBTcsHJ0G94B7iReffFLTl7X5qZUajAMBgNVHRMBAf8EAjAAMA4GA1UdDw
EB/wQEAwIEMDAMBggqgRzPVQGDdQUAA0gAMEUCIAPxfU8G3BKV2mgJVF4nMfIf
124KGfgOKX5Ka7ODQYhkAiEA1rrtg3gyvSUSEOr34kMxMRUWm7v8m1X6LW/bai7＋GYo＝

A.4.3 计算过程

计算过程包括以下内容。

a) 根据“SGD_SM4_CCM”算法标识生成保护 JSON 结构如下：

```
{
    "enc":"SGD_SM4_CCM"
}
```

JSON 结构的字符串使用 UTF-8 编码，进行 base64url 编码，生成保护头部编码为：
eyJlbmMiOiJTR0RfU000X0NDTSJ9

b) 按照“SGD_SM4_CCM”算法的密钥要求，产生 256 位的随机数 CEK，其值使用 16 进制表示为：
183931D67B44A69F72950AE4DB0F2A0D

c) 第一个接收者相关信息计算：

1) 根据 SM2 证书，使用 SM3 算法计算证书 DER 编码的杂凑值，将杂凑值进行 base64url 编码，结果为：AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A

2) 根据加密密钥算法标识“SGD_SM2_3”生成接收者无保护头部的 JSON 结构如下：

```
{
    "alg":"SGD_SM2_3",
    "x5t#sm3":"AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A"
}
```

3) 根据 SM2 密钥，按照 SM2 加密算法要求对 CEK 加密，将加密结果进行 base64url 编码，生成密钥密文编码为：
MHkCICVOQ9b9QgGiKH1sW5ajNjQR_64xGV92GeFidDFadOe4AiEAjuG47wEyijtZdF
TtQkkQfxw5eO3wROTk2VQ85fj1Xr0EIErOGdVBXTRaQDO0eYElgXqQ9TGLmxnH
nNN3-hvhRrnyBBALav22M3OtRXivx0n_zHNl

d) 第二个接收者相关信息计算：

1) 使用 SM3 算法计算 SM2 证书 DER 编码的杂凑值，将杂凑值进行 base64url 编码，结果为：
fi1NwzSZ4rS09NKYpo82lKYZe9N8bk3uMWANL8U71qw

2) 根据加密密钥算法“SGD_SM2_3”生成接收者无保护头部的 JSON 结构如下：

```
{
    "alg":"SGD_SM2_3",
    "x5t#sm3":"fi1NwzSZ4rS09NKYpo82lKYZe9N8bk3uMWANL8U71qw"
}
```

3) 根据 SM2 密钥，按照 SM2 加密算法要求对 CEK 加密，将加密结果进行 base64url 编码，生成密钥密文编码为：
MHkCIC3vpLe7X-DSYOPHsEpyhxyjZ-ol_iZTC8NmNewZTKJfAiEAzOl4Qg1yKs7yA
cMmlboUERdnZI9yy9usrnTyY-VigyUEIEJLyhkgiBDkPlphAffeQmlHLpNevE-PdMvq
CAzKvvRTBBC2bK-R0qOgJmfMpOunlgzO

e) 按照“SGD_SM4_CCM”算法的初始化向量参数要求，产生 64 位的随机数 IV，其值使用 16 进制表示为：3C2E4B85ECA9C457 进行 base64url 编码后，初始化向量编码为：A0D6F3C3BE182616

f) 额外可鉴别数据 AAD 进行 base64url 编码，额外可鉴别数据编码为：YWFkIGRhdGE
设置可鉴别加密算法的可鉴别数据输入 A＝(保护头部编码||'.'||额外可鉴别数据编码)，
A 的值为：eyJlbmMiOiJTR0RfU000X0NDTSJ9.YWFkIGRhdGE

g) 按照 SM4 算法 CCM 可鉴别加密机制，使用 IV、A 和 CEK 对待加密数据 M 进行加密。
将加密密文和鉴别标志进行 base64url 编码后，
JWE 密文编码为：nQzvzuKXvNr92Z_WWMvT1Q6U

JWE 标志编码为：ecbQ8fmf4kePMxKowOGg2w

A.4.4 输出结果

根据上面计算的保护头部编码、密钥密文编码、初始化向量编码、JWE 密文编码、JWE 标志编码，输出 JSON 序列化（多人加密不支持紧凑序列化）结果如下：


```
{
    "aad":"YWFkIGRhdGE",
    "ciphertext":"nQzvzuKXvNr92Z_WWMvT1Q6U",
    "iv":"oNbzw74YJhY",
    "protected":"eyJlbmMiOiJTR0RfU000X0NDTSJ9",
    "recipients":[{"encrypted_key":"MHkCICVOQ9b9QgGiKH1sW5ajNjQR_64xGV92GeFidD
FadOe4AiEAjuG47wEyijtZdFTtQkkQfxw5eO3wROTk2VQ85fj1Xr0EIErOGdVBXTRaQDO0eY
ElgXqQ9TGLmxnHnNN3-hvhRrnyBBALav22M3OtRXivx0n_zHNl","header":{"alg":"SGD_S
M2_3","x5t#sm3":"AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A"}},{"encrypted_
key":"MHkCIC3vpLe7X-DSYOPHsEpyhxyjZ-ol_iZTC8NmNewZTKJfAiEAzOl4Qg1yKs7yAcM
mlboUERdnZI9yy9usrnTyY-VigyUEIEJLyhkgiBDkPlphAffeQmlHLpNevE-PdMvqCAzKvvRTB
BC2bK-R0qOgJmfMpOunlgzO","header":{"alg":"SGD_SM2_3","x5t#sm3":"fi1NwzSZ4rS09
NKYpo82lKYZe9N8bk3uMWANL8U71qw"}}],
    "tag":"ecbQ8fmf4kePMxKowOGg2w"
}
```


A.5 基于“SGD_SM2_3”和“SGD_SM4_GCM”的 JWE 多个接收者示例

A.5.1 概述

该示例展示了将同一个消息加密后发给两个独立的接收者，使用 SM2 加密算法来分别加密两个接收者各自使用的 SM4 对称密钥，使用“x5t#sm3”来分别标识两个接收者的证书。在计算 SM2 加密完成后，使用 SM4 算法 GCM 可鉴别机制加密消息内容。

A.5.2 输入参数

输入参数包括以下内容。

a) 待加密数据 M：message encryption
   16 进制表示为：6D65737361676520656E6372797074696F6E

b) 额外可鉴别数据 AAD：aad data
   16 进制表示为：6161642064617461

c) 加密密钥算法："SGD_SM2_3"

d) 内容加密算法："SGD_SM4_GCM"

e) 第一个接收者的 SM2 密钥：
   使用如下 JWK 格式来表示：


```
{
    "kty":"EC",
    "crv":"sm2p256v1",
    "use":"enc",
    "x":"yJxuyluqb24nw_VpAMFMP0ZtryQexPUwYhKt79oe0Jw",
```

"y":"dEK3Qy1crFnVjlNlac-Tx_CnkJoXhyBB8Y1Jm-FzKKE"
}

“x”和“y”值是用椭圆曲线点坐标的字节串进行 base64url 编码。

第一个接收者的私钥 d 使用 base64url 编码为：

sNPitpyTsL_kJAG55gVlEUu-9MKR_saPkcsvU15RQMY

f) 第一个接收者 SM2 证书：

使用 SM2 密钥的证书 base64 编码如下：

MIIBqjCCAU6gAwIBAgIUMEzO577ZywenFSJMKtKRFFgyNVEwDAYIKoEcz1UBg3UFA
DAlMQswCQYDVQQGEwJDTjEWMBQGA1UEAwwNU20yX1Jvb3RfVGVzdDAeFw0yMD
AyMjUwMzQxMjVaFw0zNTAyMjUwMzQxMjVaMB8xCzAJBgNVBAYTAkNOMRAwDg
YDVQQDDAdTTTIgRW5jMFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAEyJxuyluqb24n
w/VpAMFMP0ZtryQexPUwYhKt79oe0Jx0QrdDLVysWdWOU2Vpz5PH8KeQmheHIEHxj
Umb4XMooaNgMF4wHwYDVR0jBBgwFoAUqgvWcEhiceHkExasPzhPHS0nevkwHQYDV
R0OBBYEFPmJEyELhHTKVBtn7CEiLAGjE/MiMAwGA1UdEwEB/wQCMAAwDgYDVR
0PAQH/BAQDAgQwMAwGCCqBHM9VAYN1BQADSAAwRQIhANJueZ6FVPM1+3Cuv
N9xs7e1TVBJTH5XIfSdHUtw/yOtAiA74TgtAAyHNKQ2q3+q1foC1ckVZYpHZ+fs9gcG
atYfhA==

g) 第二个接收者的 SM2 密钥：

使用如下 JWK 格式来表示：

{
"kty":"EC",
"crv":"sm2p256v1",
"use":"enc",
"x":"EVIjXSs9YhjUsHcGOuyVyYbIF8wFQ2wCp9SIBn529zY",
"y":"2fANOVwJQkGKpGkZRNQfi__Jxlpf2lBs3UDo3MQ5ZXQ"
}

“x”和“y”值是用椭圆曲线点坐标的字节串进行 base64url 编码。

第二个接收者的私钥 dd 使用 base64url 编码为：

nBy-Ofbm-O0nTdQbPvGyiWY3MR3JVjXzziuOJ4T8gRo

h) 第二个接收者的 SM2 证书：

使用 SM2 密钥的证书 base64 编码如下：

MIIBqzCCAU+gAwIBAgIUQrP4x64Mf+Jag4RUbjlt2NGU8d8wDAYIKoEcz1UBg3UFAD
AlMQswCQYDVQQGEwJDTjEWMBQGA1UEAwwNU20yX1Jvb3RfVGVzdDAeFw0yMD
AzMDIxMjA3NTJaFw0zNTAzMDIxMjA3NTJaMCAxCzAJBgNVBAYTAkNOMREwDwY
DVQQDDAhTTTIgRW5jMjBZMBMGByqGSM49AgEGCCqBHM9VAYItA0IABBFSI10rP
WIY1LB3BjrslcmGyBfMBUNsAqfUiAZ+dvc22fANOVwJQkGKpGkZRNQfi//Jxlpf2lBs3U
Do3MQ5ZXSjYDBeMB8GA1UdIwQYMBaAFKoL1nBIYnHh5BMWrD84Tx0tJ3r5MB0GA1
UdDgQWBBTcsHJ0G94B7iReffFLTl7X5qZUajAMBgNVHRMBAf8EAjAAMA4GA1UdDw
EB/wQEAwIEMDAMBggqgRzPVQGDdQUAA0gAMEUCIAPxfU8G3BKV2mgJVF4nMfIf1
24KGfgOKX5Ka7ODQYhkAiEA1rrtg3gyvSUSEOr34kMxMRUWm7v8m1X6LW/bai7+GYo=

A.5.3 计算过程

计算过程包括以下内容。

a) 根据“SGD_SM4_GCM”算法标识生成保护 JSON 结构如下：

```
{
    "enc":"SGD_SM4_GCM"
}
```

JSON 结构的字符串使用 UTF-8 编码，进行 base64url 编码，生成保护头部编码为：
eyJlbmMiOiJTR0RfU000X0dDTSJ9

b) 按照“SGD_SM4_GCM”算法的密钥要求，产生 256 位的随机数 CEK，其值使用 16 进制表示为：4D2EFA31D1BC99BE2AFA4DD8D95EA363

c) 第一个接收者相关信息计算：

1) 根据 SM2 证书，使用 SM3 算法计算证书 DER 编码的杂凑值，将杂凑值进行 base64url 编码，结果为：AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A

2) 根据加密密钥算法标识“SGD_SM2_3”生成接收者无保护头部的 JSON 结构如下：

```
{
    "alg":"SGD_SM2_3",
    "x5t#sm3":"AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A"
}
```

3) 根据 SM2 密钥，按照 SM2 加密算法要求对 CEK 加密，将加密结果进行 base64url 编码，生成密钥密文编码为：
MHoCIQDgzRxlstZez2nk3JJJ7RdsEmAew9uytuvW6hVzT6xKJwIhALZYKS1z271KYN
mRC9exhHmBKJ_HEw_RKvyGBA8-k6rdBCCZ14wc5aHZGpo7oemqh_fup3e5tHlJv6s6
ttrODGtJDwQQu_0iEC3jlxPmXdwW9SpSOg

c) 第二个接收者相关信息计算：

1) 使用 SM3 算法计算 SM2 证书 DER 编码的杂凑值，将杂凑值进行 base64url 编码，结果为：
fi1NwzSZ4rS09NKYpo82lKYZe9N8bk3uMWANL8U71qw

2) 根据加密密钥算法"SGD_SM2_3"生成接收者无保护头部的 JSON 结构如下：

```
{
    "alg":"SGD_SM2_3",
    "x5t#sm3":"fi1NwzSZ4rS09NKYpo82lKYZe9N8bk3uMWANL8U71qw"
}
```

3) 根据 SM2 密钥，按照 SM2 加密算法要求对 CEK 加密，将加密结果进行 base64url 编码，生成密钥密文编码为：
MHoCIQD4dazgtperMhiUv_8G8OGIne9Ixu5bSZ3mqR6KcmUgjgIhAKarINqw2RqEujK
ZFjz9SS6Ny-ifs8N75UxFLz1hCiEnBCBNbe8pDDt4TmUSWLJwf-2AlxbQS6ro-EW_V
O-hBkr-HgQQmGTdu9jbrwgZMxn2M2uRrA

e) 按照“SGD_SM4_GCM”算法的初始化向量参数要求，产生 96 位的随机数 IV，其值使用 16 进制表示为：D2572C4D9B179C2B0189B251 进行 base64url 编码后，初始化向量编码为：
0lcsTZsXnCsBibJR

f) 额外可鉴别数据 AAD 进行 base64url 编码，额外可鉴别数据编码为：YWFkIGRhdGE
设置可鉴别加密算法的可鉴别数据输入 A＝(保护头部编码||'.'||额外可鉴别数据编码)，
A 的值为：eyJlbmMiOiJTR0RfU000X0dDTSJ9.0lcsTZsXnCsBibJR

g) 按照 SM4 算法 GCM 可鉴别加密机制，使用 IV、A 和 CEK 对待加密数据 M 进行加密。
将加密密文和鉴别标志进行 base64url 编码后，
JWE 密文编码为：L2SvvvGSY0R-qbN4LVgCPApH

JWE 标志编码为：ObSyN69IBVRqigJYEOHNrw

A.5.4 输出结果

根据上面计算的保护头部编码、密钥密文编码、初始化向量编码、JWE 密文编码、JWE 标志编码，输出 JSON 序列化结果如下：


```
{
    "aad":"YWFkIGRhdGE",
    "ciphertext":"L2SvvvGSY0R-qbN4LVgCPApH",
    "iv":"0lcsTZsXnCsBibJR",
    "protected":"eyJlbmMiOiJTR0RfU000X0dDTSJ9",
    "recipients":[
    {"encrypted_key":"MHoCIQDgzRxlstZez2nk3JJJ7RdsEmAew9uytuvW6hVzT6xKJwIhALZYKS1z271KYNmRC9exhHmBKJ_HEw_RKvyGBA8-k6rdBCCZ14wc5aHZGpo7oemqh_fup3e5tHlJv6s6ttrODGtJDwQQu_0iEC3jlxPmXdwW9SpSOg","header":{"alg":"SGD_SM2_3","x5t#sm3":"AnRj74ySDd5C48fUuSnwCYUu7A5EoCfigFeepoz_b9A"}},
    {"encrypted_key":"MHoCIQD4dazgtperMhiUv_8G8OGIne9Ixu5bSZ3mqR6KcmUgjgIhAKarINqw2RqEujKZFjz9SS6Ny-ifs8N75UxFLz1hCiEnBCBNbe8pDDt4TmUSWLJwf-2AlxbQS6ro-EW_VO-hBkr-HgQQmGTdu9jbrwgZMxn2M2uRrA","header":{"alg":"SGD_SM2_3","x5t#sm3":"fi1NwzSZ4rS09NKYpo82lKYZe9N8bk3uMWANL8U71qw"}}],
    "tag":"ObSyN69IBVRqigJYEOHNrw"
}
```