

JCATT ファイルフォーマット仕様書

RSASSA-PKCS1-v1_5

2018 年 11 月

独立行政法人情報処理推進機構

目次

1	はじめに	3
2	RSASSA-PKCS1-v1_5	4
2.1	JCATT2 互換ファイルフォーマット	4
2.1.1	パラメータファイル (*.par)	5
2.1.2	リクエストファイル (*.req)	6
2.1.3	Facts ファイル (*.fax)	7
2.1.4	レスポンスファイル (*.rsp)	9
2.2	CAVS 準互換ファイルフォーマット	12
2.2.1	パラメータファイル (*.par)	12
2.2.2	リクエストファイル (*.req)	13
2.2.3	Facts ファイル (*.fax)	14
2.2.4	レスポンスファイル (*.rsp)	15

1 はじめに

暗号アルゴリズム実装試験ツール (以下 JCATT と略記する) が使用する各種ファイルのフォーマット規則を記述する。JCATT が使用するファイルには次のようなものがある。

ファイルの種類

- パラメータファイル (*.par)
試験項目の設定を記述する。JCATT を用いて作成する。
- リクエストファイル (*.req)
暗号モジュール開発ベンダに対する要求を記述する。JCATT を用いて作成する。
- Facts ファイル (*.fax)
テストベクタを記述する。JCATT を用いて作成する。
- レスポンスファイル (*.rsp)
ベンダからの回答を記述する。リクエストファイルおよび本稿で指定するファイルフォーマットに基づいてベンダが作成する。
- 結果ファイル (*.out)
試験結果を記述する。JCATT を用いて作成する。

これらのファイルの名前は、次の規則に従ってつけること。

ファイル名の規則

- 拡張子は、上記 () 内に指定したものを使用すること。
- 拡張子以外の名前は、試験対象実装ごとに同じ名称をつけること。
リクエストファイル (*.req) と Facts ファイル (*.fax) の生成時には、リクエストファイル (*.req) と Facts ファイル (*.fax) に対してパラメータファイル (*.par) と同じ名称を JCATT が自動的につける。
試験実行時には、同じ名称のレスポンスファイル (*.rsp) と Facts ファイル (*.fax) に対して試験が行われる。また、試験実行時は、結果ファイル (*.out) に対して、Facts ファイル (*.fax) と同じ名称を JCATT が自動的につける。

ファイルフォーマット詳細は次章以降に記述する。各ファイルに共通の規則は次の通りである。

共通規則

- JCATT 互換ファイルフォーマットの選択時, [] で囲まれた“タグ”の次の行に値を記述する。
- CAVS 準互換ファイルフォーマットの選択時, < タグ > = < 値 > の形式で 1 行で記述する。
- ヘッダ部分については各行について [< タグ > = < 値 >] の形式で 1 行で記述する。
- レスポンスファイルにおいては、【出力】と記述したタグが、試験対象実装が出力するデータを記述する箇所である。
- 半角英数字を用いること。
- タグおよび値は大文字小文字の区別をするので、大文字小文字を含めて正確に記述すること。ただし、数値を 16 進数で記述する場合は、大文字小文字は区別しない。
- 一文字目が # (半角) で始まるコメント行を自由に書き込むことができる。
- 平文、暗号文、鍵などのデータの区切り文字は改行 (CR+LF または LF) とする。
- 平文、暗号文、署名、鍵などのデータは 16 進表記とする。
- ビット数、個数などの数値は 10 進表記とする。
- ACSII コードを使用すること。
- 各行には必ず改行を入れること (最後のデータと EOF との間にも改行を入れること)。

2 RSASSA-PKCS1-v1_5

RSASSA-PKCS1-v1_5 の暗号アルゴリズム実装試験のためのファイルフォーマットを記述する。

2.1 JCATT2 互換ファイルフォーマット

各表中，プライベート鍵識別子，公開鍵指数 e 識別子，ハッシュ関数識別子は下表の通りである。

表1 プライベート鍵識別子

識別子	対応するプライベート鍵
TYPE1	プライベート鍵は $(p, q, dP, dQ, qInv)$
TYPE2	プライベート鍵は (d, n)

表2 公開鍵指数 e 識別子

識別子	対応する公開鍵指数 e
TYPE1	$e = 65,537$
TYPE2	e はランダム

表3 ハッシュ関数識別子

ハッシュ関数識別子	対応するハッシュ関数
M_Hash_SHA1	SHA-1
M_Hash_SHA256	SHA-256
M_Hash_SHA384	SHA-384
M_Hash_SHA512	SHA-512

2.1.1 パラメータファイル (*.par)

表4 RSASSA-PKCS1-v1_5 パラメータファイル

機能	タグ	内容
(共通)	[Algorithm Name]	RSASSA-PKCS1-v1_5
署名生成	[Function Name]	Sign
	[Bitlength of Modulus]	法 n のビット長
	[Hash]	ハッシュ関数識別子
	[Secret Key Type]	プライベート鍵識別子
	[Public Key Type]	公開鍵指数 e 識別子
	[Seed P]	ランダムな平文を生成するための擬似乱数生成関数用乱数シード
	[Bitlength of Seed P]	Seed P のビット長
	[Seed K]	鍵ペア生成のための擬似乱数生成用乱数シード
	[Bitlength of Seed K]	Seed K のビット長
	[Bitlength of Plaintexts]	平文のビット長
	[Number of Plaintexts]	平文の数
署名検証	[Function Name]	Verification
	[Bitlength of Modulus]	署名生成と同じ
	[Hash]	
	[Secret Key Type]	
	[Public Key Type]	
	[Seed P]	
	[Bitlength of Seed P]	
	[Seed K]	
	[Bitlength of Seed K]	
	[Bitlength of Plaintexts]	
	[Number of Plaintexts]	
	[Rate of Fail Data]	署名検証が不合格になる割合
鍵ペア生成	[Function Name]	Key Generation
	[Bitlength of Modulus]	法 n のビット長
	[Secret Key Type]	プライベート鍵識別子
	[Public Key Type]	公開鍵指数 e 識別子
	[Number of Keys]	鍵の数

2.1.2 リクエストファイル (*.req)

表5 RSASSA-PKCS1-v1_5 リクエストファイル

機能	タグ	内容
(共通)	[Algorithm Name]	RSASSA-PKCS1-v1_5
署名生成	[Function Name]	Sign
	[Bitlength of Modulus]	法 n のビット長 [10 進数表記]
	[Hash]	ハッシュ関数識別子
	[Secret Key Type]	プライベート鍵識別子
	[p] ¹	素数 p [16 進数表記]
	[q] ¹	素数 q [16 進数表記]
	[d] ¹	プライベート鍵 d [16 進数表記]
	[n] ¹	法 n [16 進数表記]
	[dP] ¹	$d \bmod p - 1$ [16 進数表記]
	[dQ] ¹	$d \bmod q - 1$ [16 進数表記]
	[qInv] ¹	$q^{-1} \bmod p$ [16 進数表記]
	[Bitlength of Plaintexts]	平文のビット長 [10 進数表記]
	[Number of Plaintexts]	平文の数 [10 進数表記]
	[Plaintexts] ²	平文 [16 進数表記]
署名検証	[Function Name]	Verification
	[Bitlength of Modulus]	法 n のビット長 [10 進数表記]
	[Hash]	ハッシュ関数識別子
	[n]	法 n [16 進数表記]
	[Number of Signatures]	署名の数 [10 進数表記]
	[e] ³	公開鍵指数 e [16 進数表記]
	[Signatures] ³	署名 [16 進数表記]
	[Bitlength of Plaintexts]	平文のビット長 [10 進数表記]
	[Plaintexts] ³	平文 [16 進数表記]
鍵ペア生成	[Function Name]	Key Generation
	[Bitlength of Modulus]	法 n のビット長 [10 進数表記]
	[Secret Key Type]	プライベート鍵識別子
	[Public Key Type]	公開鍵指数 e 識別子
	[Number of Keys]	鍵の数 [10 進数表記]

注

1. プライベート鍵識別子に応じて、([d],[n]) の組または ([p],[q],[dP],[dQ],[qInv]) の組いずれか一方を記述する。
2. [Number of Plaintexts] 個の平文を記述する。
3. [Number of Signatures] 個の公開鍵指数 e 、署名および平文を記述する。

2.1.3 Facts ファイル (*.fax)

表6 RSASSA-PKCS1-v1_5 Facts ファイル

機能	タグ	内容
(共通)	[Algorithm Name]	RSASSA-PKCS1-v1_5
署名生成	[Function Name]	Sign
	[Bitlength of Modulus]	法 n のビット長
	[Hash]	ハッシュ関数識別子
	[Secret Key Type]	プライベート鍵識別子
	[Public Key Type]	公開鍵指数 e 識別子
	[p] ¹	素数 p
	[q] ¹	素数 q
	[d] ¹	プライベート鍵 d
	[n] ¹	法 n
	[dP] ¹	$d \bmod p - 1$
	[dQ] ¹	$d \bmod q - 1$
	[qInv] ¹	$q^{-1} \bmod p$
	[e]	公開鍵指数 e
	[Bitlength of Plaintexts]	平文のビット長
	[Number of Plaintexts]	平文の数
	[Plaintexts] ²	平文
	[Signatures] ²	署名

注

1. プライベート鍵識別子に応じて, $([d],[n])$ の組または $([p],[q],[n],[dP],[dQ],[qInv])$ の組いずれか一方を記述する.
2. [Number of Plaintexts] 個の平文および署名を記述する.

表7 RSASSA-PKCS1-v1_5 Facts ファイル (続き)

機能	タグ	内容
(共通)	[Algorithm Name]	RSASSA-PKCS1-v1_5
署名検証	[Function Name]	Verification
	[Bitlength of Modulus]	法 n のビット長
	[Hash]	ハッシュ関数識別子
	[Secret Key Type]	プライベート鍵識別子
	[Public Key Type]	公開鍵指数 e 識別子
	[p] ¹	素数 p
	[q] ¹	素数 q
	[d] ¹	プライベート鍵 d
	[n] ¹	法 n
	[dP] ¹	$d \bmod p - 1$
	[dQ] ¹	$d \bmod q - 1$
	[qInv] ¹	$q^{-1} \bmod p$
	[Number of Signatures]	署名の数
	[e] ²	公開鍵指数 e
	[Signatures] ²	署名
	[Bitlength of Plaintexts]	平文のビット長
	[Plaintexts] ²	平文
	[Results] ²	署名検証結果. 署名検証合格の場合 0, 不合格の場合 1 と記述する. [Signatures] および [Plaintexts] に記述した検証対象署名および平文の順に 1 または 0 を記述すること.
鍵ペア生成	[Function Name]	Key Generation
	[Bitlength of Modulus]	法 n のビット長
	[Secret Key Type]	プライベート鍵識別子
	[Public Key Type]	公開鍵指数 e 識別子
	[Number of Keys]	鍵の数

注

1. プライベート鍵識別子に応じて, $([d],[n])$ の組または $([p],[q],[n],[dP],[dQ],[qInv])$ の組いずれか一方を記述する.
2. [Number of Signatures] 個の公開鍵指数 e , 署名, 平文, および署名検証結果を記述する.

2.1.4 レスポンスファイル (*.rsp)

表8 RSASSA-PKCS1-v1_5 レスポンスファイル

機能	タグ	内容
(共通)	[Algorithm Name]	RSASSA-PKCS1-v1_5
署名生成	[Function Name]	Sign
	[Bitlength of Modulus]	法 n のビット長 [10 進数表記]
	[Hash]	ハッシュ関数識別子
	[Secret Key Type]	プライベート鍵識別子
	[p] ¹	素数 p [16 進数表記]
	[q] ¹	素数 q [16 進数表記]
	[d] ¹	プライベート鍵 d [16 進数表記]
	[n] ¹	法 n [16 進数表記]
	[dP] ¹	$d \bmod p - 1$ [16 進数表記]
	[dQ] ¹	$d \bmod q - 1$ [16 進数表記]
	[qInv] ¹	$q^{-1} \bmod p$ [16 進数表記]
	[Bitlength of Plaintexts]	平文のビット長 [10 進数表記]
	[Number of Plaintexts]	平文の数 [10 進数表記]
	[Plaintexts] ²	平文 [16 進数表記]
	[Signatures] ²	【出力】署名 [16 進数表記]
署名検証	[Function Name]	Verification
	[Bitlength of Modulus]	法 n のビット長 [10 進数表記]
	[Hash]	ハッシュ関数識別子
	[n]	法 n [16 進数表記]
	[Number of Signatures]	署名の数 [10 進数表記]
	[e] ³	公開鍵指数 e [16 進数表記]
	[Signatures] ³	署名 [16 進数表記]
	[Bitlength of Plaintexts]	平文のビット長 [10 進数表記]
	[Plaintexts] ³	平文 [16 進数表記]
	[Results] ³	【出力】署名検証結果。検証合格の時 0, 不合格の時 1 と記述する。

注

1. プライベート鍵識別子に応じて、([d],[n]) の組または ([p],[q],[dP],[dQ],[qInv]) の組いずれか一方を記述する。
2. [Number of Plaintexts] 個の平文および署名を記述する。
3. [Number of Signatures] 個の公開鍵指数 e , 署名, 平文, および署名検証結果を記述する。

表9 RSASSA-PKCS1-v1_5 レスポンスファイル (続き)

機能	タグ	内容
(共通)	[Algorithm Name]	RSASSA-PKCS1-v1_5
鍵ペア生成	[Function Name]	Key Generation
	[Bitlength of Modulus]	法 n のビット長 [10 進数表記]
	[Secret Key Type]	プライベート鍵識別子
	[Number of Keys]	鍵の数 [10 進数表記]
	[p] ¹	【出力】素数 p [16 進数表記]
	[q] ¹	【出力】素数 q [16 進数表記]
	[d] ¹	【出力】プライベート鍵 d [16 進数表記]
	[e] ¹	【出力】公開鍵指数 e [16 進数表記]
	[n] ¹	【出力】法 n [16 進数表記]
	[dP] ¹	【出力】 $d \bmod p - 1$ [16 進数表記]
	[dQ] ¹	【出力】 $d \bmod q - 1$ [16 進数表記]
	[qInv] ¹	【出力】 $q^{-1} \bmod p$ [16 進数表記]

注

1. プライベート鍵のタイプに応じて、([p], [q], [d], [e], [n]) の組または ([p], [q], [dP], [dQ], [qInv], [e], [n]) の組いずれか一方を [Number of Keys] 個記述する。

プライベート鍵識別子に TYPE1 が指定されているとき、鍵 ([p], [q], [dP], [dQ], [qInv], [e], [n]) は次の例に従って記述すること。すなわち、2 組以上の鍵を記述する時は、まず 1 組目の ([p], [q], [dP], [dQ], [qInv], [e], [n]) をこの順番で記述し、次に 2 組目の ([p], [q], [dP], [dQ], [qInv], [e], [n]) をこの順番で記述する。(以降同様)

[p]
... # 1 組目の素数 p を記述する。
[q]
... # 1 組目の素数 q を記述する。
[dP]
... # 1 組目の $d \bmod p - 1$ を記述する。
[dQ]
... # 1 組目の $d \bmod q - 1$ を記述する。
[qInv]
... # 1 組目の $q^{-1} \bmod p$ を記述する。
[e]
... # 1 組目の公開鍵指数 e を記述する。
[n]
... # 1 組目の法 n を記述する。
[p]
... # 2 組目の素数 p を記述する。
[q]
... # 2 組目の素数 q を記述する。
[dP]
... # 2 組目の $d \bmod p - 1$ を記述する。
[dQ]
... # 2 組目の $d \bmod q - 1$ を記述する。
[qInv]
... # 2 組目の $q^{-1} \bmod p$ を記述する。
[e]

... # 2 組目の公開鍵指数 e を記述する.

[n]

... # 2 組目の法 n を記述する.

注

- 試験合格の場合, `< Results >` に OK と表示される.
- 試験不合格の場合, `< Results >` に何らかの形式で NG と表示される. また, `< Results >` には, レスポンスファイル内の不合格となったデータが記述されている何番目 (COUNT, # 等の記号で番号を表す) のデータが不合格となったかが表示される. 不合格となったデータが記述されているタグ名は, 前記のレスポンスファイル仕様に【出力】と記述したタグである. ただし, 【出力】と記述したタグが 1 つしかない場合, タグ名は省略することがある.

- 鍵ペア生成機能に対する試験において試験不合格の場合, 下記のようにどの条件が合格 (ok) でどの条件が不合格 (NG) となったかも表示される.

NG(#1 : p is prime ?)

ok(#1 : q is prime ?)

例えば, ok(#1 : q is prime ?) は, q は素数であるという条件を満たしていることを示し, NG(#1 : p is prime ?) は, p は素数であるという条件を満たしていないことを示す. 詳細は別紙の試験項目を記述した文書を参照のこと.

2.2 CAVS 準互換ファイルフォーマット

この章で取り扱うファイルフォーマットでは、公開鍵指数 e 識別子として、表10に記載された表現、プライベート鍵識別子として、表11に記載された表現、ハッシュ関数識別子として、表12に記載された表現を用いる。

表10 公開鍵指数 e 識別子

識別子	対応する公開鍵指数 e
TYPE1	$e = 65,537$
TYPE2	e はランダム

表11 プライベート鍵識別子

識別子	対応するプライベート鍵
TYPE2	プライベート鍵は (d, n)

表12 ハッシュ関数識別子

ハッシュ関数識別子	対応するハッシュ関数
SHA1	SHA-1
SHA224	SHA-224
SHA256	SHA-256
SHA384	SHA-384
SHA512	SHA-512
SHA512224	SHA-512/224
SHA512256	SHA-512/256

2.2.1 パラメータファイル (*.par)

表13 PKCS#1 に記載された RSASSA-PKCS1-v1_5 パラメータファイル

機能	タグ	内容	表記
共通	全体ヘッダ	TargetFunction	KeyGeneration, SignatureGeneration, 又は SignatureVerification
		TypeOfPublicKey	公開鍵指数 e 識別子
		TypeOfPrivateKey	プライベート鍵識別子
		mod	法 n のビット長
	鍵ペア生成ヘッダ	NumberOfKeyPairs	生成する鍵ペアの数
	署名生成ヘッダ	HashAlgorithm	ハッシュ関数識別子
		BitLengthOfPlainText	平文のビット長
		NumberOfPlainTexts	平文の数
	署名検証ヘッダ	HashAlgorithm	ハッシュ関数識別子
		BitLengthOfPlainText	平文のビット長
		NumberOfPlainTexts	平文の数
		RatioOfInvalidData	署名検証の選択時、署名検証が不合格になる割合

2.2.2 リクエストファイル (*.req)

表14: PKCS#1 に記載された RSASSA-PKCS1-v1_5 リクエストファイル

機能	分類	タグ	内容	暗号アルゴリズム実装試験仕様書		例示
				—公開鍵— 上の表記との対応	値の表記	
共通	全体ヘッダ	TargetFunction	KeyGeneration, SignatureGeneration, 又は SignatureVerification.		文字列	[TargetFunction = SignatureGeneration]
		TypeOfPublicKey	公開鍵指数 e 識別子		文字列	[TypeOfPublicKey = TYPE2]
		TypeOfPrivateKey	プライベート鍵識別子		文字列	[TypeOfPrivateKey = TYPE2]
		mod	法 n のビット長		10 進数表記	[mod = 2048]
鍵ペア生成	ヘッダ	NumberOfKeyPairs	生成する鍵ペアの数		10 進数表記	[NumberOfKeyPairs = 20]
		Table for M-R Test	Miller-Rabin probabilistic primality test の設定		文字列	[Table for M-R Test = C.2]
署名生成	ヘッダ	HashAlgorithm	ハッシュ関数識別子	ハッシュ関数	文字列	[HashAlgorithm = SHA256]
		BitLengthOfPlainText	平文のビット長		10 進数表記	[BitLengthOfPlainText = 1024]
		NumberOfPlainTexts	平文の数		10 進数表記	[NumberOfPlainTexts = 2048]
		n	法 n		16 進数表記	n = B373...DFE5
		e	公開鍵指数 e		16 進数表記	e = 8bb7 ... b24f
		d	プライベート鍵 d		16 進数表記	d = 0415 ... fca1
	本体*1	COUNT	0 以上 NumberOfPlainTexts 未満の整数		10 進数表記	COUNT = 0
		SHAAlg	ハッシュ関数識別子	ハッシュ関数	文字列	SHAAlg = SHA256
		Msg	入力メッセージ	平文	16 進数表記	Msg = d172...b13f
		S	署名生成機能の出力であるバイト列のプレースホルダ	署名	16 進数表記	S = ?
署名検証	ヘッダ	HashAlgorithm	ハッシュ関数識別子	ハッシュ関数	文字列	[HashAlgorithm = SHA256]
		BitLengthOfPlainText	平文のビット長		10 進数表記	[BitLengthOfPlainText = 1024]
		NumberOfPlainTexts	平文の数		10 進数表記	[NumberOfPlainTexts = 2048]
		RatioOfInvalidData	署名検証の選択時, 署名検証が不合格になる割合. それ以外は省略.		浮動小数点表記	[RatioOfInvalidData = 0.8]
	本体*2	n	法 n		16 進数表記	n = B373...DFE5
		COUNT	0 以上 NumberOfPlainTexts 未満の整数		10 進数表記	COUNT = 0
		SHAAlg	ハッシュ関数識別子	ハッシュ関数	文字列	SHAAlg = SHA256
		e	公開鍵指数 e		16 進数表記	e = 010001
		d	プライベート鍵 d . 署名検証には使用されない.		16 進数表記	d = 0
		Msg	入力メッセージ	平文	16 進数表記	Msg = 2f9a ... 3664
		S	署名検証機能の入力であるバイト列	署名	16 進数表記	S = b7f3 ... 9744
		Result	署名検証結果のプレースホルダ. 検証合格の時 P, 不合格の時 F と記述する.		文字列	Result = ?

*1 NumberOfPlainTexts 個の各データの組を以下のように記述する.

COUNT = 0
SHAAlg = SHA256
Msg = 2f9a ... 3664
S = ?

i = 0 のデータの組について記述する.
i = 0 に対応するハッシュ関数識別子を記述する.
i = 0 に対応する入力メッセージ.
i = 0 に対応する署名生成機能の出力であるバイト列のプレースホルダ.

COUNT = 1
SHAAlg = SHA256
Msg = 2f9a ... 3664
S = ?

i = 1 のデータの組について記述する.
i = 1 に対応するハッシュ関数識別子を記述する.
i = 1 に対応する入力メッセージ.
i = 1 に対応する署名生成機能の出力であるバイト列のプレースホルダ.

...

...

COUNT = \langle NumberOfPlainTexts - 1 \rangle
SHAAlg = SHA256
Msg = 2f9a ... 3664
S = ?

i = \langle NumberOfPlainTexts - 1 \rangle のデータの組について記述する.
i = \langle NumberOfPlainTexts - 1 \rangle に対応するハッシュ関数識別子を記述する.
i = \langle NumberOfPlainTexts - 1 \rangle に対応する入力メッセージ.
i = \langle NumberOfPlainTexts - 1 \rangle に対応する署名生成機能の出力であるバイト列のプレースホルダ.

*2 NumberOfPlainTexts 個の各データの組を以下のように記述する.

COUNT = 0
SHAAlg = SHA256
e = 010001
d = 0
Msg = 2f9a ... 3664
S = b7f3 ... 9744
Result = ?

i = 0 のデータの組について記述する.
i = 0 に対応するハッシュ関数識別子を記述する.
i = 0 に対応する公開鍵指数 e を記述する.
i = 0 に対応するプライベート鍵のダミーデータ.
i = 0 に対応する入力メッセージ.
i = 0 に対応する署名検証機能の入力であるバイト列.
i = 0 に対応する署名検証結果のプレースホルダ.

COUNT = 1
SHAAlg = SHA256
e = 010001
d = 0
Msg = 2f9a ... 3664
S = b7f3 ... 9744
Result = ?

i = 1 のデータの組について記述する.
i = 1 に対応するハッシュ関数識別子を記述する.
i = 1 に対応する公開鍵指数 e を記述する.
i = 1 に対応するプライベート鍵のダミーデータ.
i = 1 に対応する入力メッセージ.
i = 1 に対応する署名検証機能の入力であるバイト列.
i = 1 に対応する署名検証結果のプレースホルダ.

...

...

COUNT = \langle NumberOfPlainTexts - 1 \rangle
SHAAlg = SHA256
e = 010001
d = 0
Msg = 2f9a ... 3664
S = b7f3 ... 9744
Result = ?

i = \langle NumberOfPlainTexts - 1 \rangle のデータの組について記述する.
i = \langle NumberOfPlainTexts - 1 \rangle に対応するハッシュ関数識別子を記述する.
i = \langle NumberOfPlainTexts - 1 \rangle に対応する公開鍵指数 e を記述する.
i = \langle NumberOfPlainTexts - 1 \rangle に対応するプライベート鍵のダミーデータ.
i = \langle NumberOfPlainTexts - 1 \rangle に対応する入力メッセージ.
i = \langle NumberOfPlainTexts - 1 \rangle に対応する署名検証機能の入力であるバイト列.
i = \langle NumberOfPlainTexts - 1 \rangle に対応する署名検証結果のプレースホルダ.

2.2.3 Facts ファイル (*.fax)

表15: PKCS#1 に記載された RSASSA-PKCS1-v1_5 Facts ファイル

機能	分類	タグ	内容	暗号アルゴリズム実装試験仕様書 —公開鍵— 上の表記との対応	値の表記	例示
共通	全体ヘッダ	TargetFunction	KeyGeneration, SignatureGeneration, 又は SignatureVerification.		文字列	[TargetFunction = SignatureGeneration]
		TypeOfPublicKey	公開鍵指数 e 識別子		文字列	[TypeOfPublicKey = TYPE2]
		TypeOfPrivateKey	プライベート鍵識別子		文字列	[TypeOfPrivateKey = TYPE2]
		mod	法 n のビット長		10 進数表記	[mod = 2048]
鍵ペア生成	ヘッダ	NumberOfKeyPairs	生成する鍵ペアの数		10 進数表記	[NumberOfKeyPairs = 20]
		Table for M-R Test	Miller-Rabin probabilistic primality test の設定		文字列	[Table for M-R Test = C.2]
署名生成	ヘッダ	HashAlgorithm	ハッシュ関数識別子	ハッシュ関数	文字列	[HashAlgorithm = SHA256]
		BitLengthOfPlainText	平文のビット長		10 進数表記	[BitLengthOfPlainText = 1024]
		NumberOfPlainTexts	平文の数		10 進数表記	[NumberOfPlainTexts = 2048]
		n	法 n		16 進数表記	$n = \text{B373...DFE5}$
		e	公開鍵指数 e		16 進数表記	$e = \text{8bb7 ... b24f}$
		d	プライベート鍵 d		16 進数表記	$d = \text{0415 ... fca1}$
		p	素数 p		16 進数表記	$p = \text{0415 ... fca1}$
		q	素数 q		16 進数表記	$q = \text{1504 ... a1fc}$
	本体*1	COUNT	0 以上 NumberOfPlainTexts 未満の整数		10 進数表記	COUNT = 0
		SHAAlg	ハッシュ関数識別子	ハッシュ関数	文字列	SHAAlg = SHA256
		Msg	入力メッセージ	平文	16 進数表記	Msg = d172...b13f
		HashAlgorithm	ハッシュ関数識別子	ハッシュ関数	文字列	[HashAlgorithm = SHA256]
		BitLengthOfPlainText	平文のビット長		10 進数表記	[BitLengthOfPlainText = 1024]
		NumberOfPlainTexts	平文の数		10 進数表記	[NumberOfPlainTexts = 2048]
		RatioOfInvalidData	署名検証の選択時, 署名検証が不合格になる割合. それ以外は省略.		浮動小数点表記	[RatioOfInvalidData = 0.8]
		n	法 n		16 進数表記	$n = \text{B373...DFE5}$
署名検証	ヘッダ	e	公開鍵指数 e		16 進数表記	$e = \text{8bb7 ... b24f}$
		d	プライベート鍵 d		16 進数表記	$d = \text{0415 ... fca1}$
		p	素数 p		16 進数表記	$p = \text{0415 ... fca1}$
		q	素数 q		16 進数表記	$q = \text{1504 ... a1fc}$
	本体*2	COUNT	0 以上 NumberOfPlainTexts 未満の整数		10 進数表記	COUNT = 0
		SHAAlg	ハッシュ関数識別子	ハッシュ関数	文字列	SHAAlg = SHA256
		e	公開鍵指数 e		16 進数表記	$e = \text{010001}$
		d	プライベート鍵 d . 署名検証には使用されない.		16 進数表記	$d = 0$
		Msg	入力メッセージ	平文	16 進数表記	Msg = 2f9a ... 3664
		S	署名検証機能の入力であるバイト列	署名	16 進数表記	S = b7f3 ... 9744
		Result	署名検証結果の期待値. 検証合格の時 P, 不合格の時 F と記述する.		文字列	Result = P
		HashAlgorithm	ハッシュ関数識別子	ハッシュ関数	文字列	[HashAlgorithm = SHA256]
		BitLengthOfPlainText	平文のビット長		10 進数表記	[BitLengthOfPlainText = 1024]
		NumberOfPlainTexts	平文の数		10 進数表記	[NumberOfPlainTexts = 2048]
		RatioOfInvalidData	署名検証の選択時, 署名検証が不合格になる割合. それ以外は省略.		浮動小数点表記	[RatioOfInvalidData = 0.8]
		n	法 n		16 進数表記	$n = \text{B373...DFE5}$
		e	公開鍵指数 e		16 進数表記	$e = \text{8bb7 ... b24f}$
		d	プライベート鍵 d		16 進数表記	$d = \text{0415 ... fca1}$
		p	素数 p		16 進数表記	$p = \text{0415 ... fca1}$
		q	素数 q		16 進数表記	$q = \text{1504 ... a1fc}$

*1 NumberOfPlainTexts 個の各データの組を以下のように記述する.

```
COUNT = 0                                #  $i = 0$  のデータの組について記述する.
SHAAlg = SHA256                          #  $i = 0$  に対応するハッシュ関数識別子を記述する.
Msg = 2f9a ... 3664                      #  $i = 0$  に対応する入力メッセージ.

COUNT = 1                                #  $i = 1$  のデータの組について記述する.
SHAAlg = SHA256                          #  $i = 1$  に対応するハッシュ関数識別子を記述する.
Msg = 2f9a ... 3664                      #  $i = 1$  に対応する入力メッセージ.

:
COUNT = < NumberOfPlainTexts - 1 >      #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  のデータの組について記述する.
SHAAlg = SHA256                          #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応するハッシュ関数識別子を記述する.
Msg = 2f9a ... 3664                      #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応する入力メッセージ.
```

*2 NumberOfPlainTexts 個の各データの組を以下のように記述する.

```
COUNT = 0                                #  $i = 0$  のデータの組について記述する.
SHAAlg = SHA256                          #  $i = 0$  に対応するハッシュ関数識別子を記述する.
 $e = \text{010001}$                             #  $i = 0$  に対応する公開鍵指数  $e$  を記述する.
 $d = 0$                                     #  $i = 0$  に対応するプライベート鍵のダミーデータ.
Msg = 2f9a ... 3664                      #  $i = 0$  に対応する入力メッセージ.
S = b7f3 ... 9744                        #  $i = 0$  に対応する署名検証機能の入力であるバイト列.
Result = P                              #  $i = 0$  に対応する署名検証結果の期待値.

COUNT = 1                                #  $i = 1$  のデータの組について記述する.
SHAAlg = SHA256                          #  $i = 1$  に対応するハッシュ関数識別子を記述する.
 $e = \text{010001}$                             #  $i = 1$  に対応する公開鍵指数  $e$  を記述する.
 $d = 0$                                     #  $i = 1$  に対応するプライベート鍵のダミーデータ.
Msg = 2f9a ... 3664                      #  $i = 1$  に対応する入力メッセージ.
S = b7f3 ... 9744                        #  $i = 1$  に対応する署名検証機能の入力であるバイト列.
Result = P                              #  $i = 1$  に対応する署名検証結果の期待値.

:
COUNT = < NumberOfPlainTexts - 1 >      #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  のデータの組について記述する.
SHAAlg = SHA256                          #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応するハッシュ関数識別子を記述する.
 $e = \text{010001}$                             #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応する公開鍵指数  $e$  を記述する.
 $d = 0$                                     #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応するプライベート鍵のダミーデータ.
Msg = 2f9a ... 3664                      #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応する入力メッセージ.
S = b7f3 ... 9744                        #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応する署名検証機能の入力であるバイト列.
Result = F                              #  $i = \langle \text{NumberOfPlainTexts} - 1 \rangle$  に対応する署名検証結果の期待値.
```

2.2.4 レスポンスファイル (*.rsp)

表16: PKCS#1 に記載された RSASSA-PKCS1-v1_5 レスポンスファイル

機能	分類	タグ	内容	暗号アルゴリズム実装試験仕様書 —公開鍵— 上の表記との対応	値の表記	例示
共通	全体ヘッダ	TargetFunction	KeyGeneration, SignatureGeneration, 又は SignatureVerification.		文字列	[TargetFunction = SignatureGeneration]
		TypeOfPublicKey	公開鍵指数 e 識別子		文字列	[TypeOfPublicKey = TYPE2]
		TypeOfPrivateKey	プライベート鍵識別子		文字列	[TypeOfPrivateKey = TYPE2]
		mod	法 n のビット長		10 進数表記	[mod = 2048]
鍵ペア生成	ヘッダ	NumberOfKeyPairs	生成する鍵ペアの数		10 進数表記	[NumberOfKeyPairs = 20]
		Table for M-R Test	Miller-Rabin probabilistic primality test の設定		文字列	[Table for M-R Test = C.2]
	本体*1	COUNT	【出力】 0 以上 NumberOfKeyPairs 未満の整数		10 進数表記	COUNT = 0
		e	【出力】 鍵ペア生成機能の出力である公開鍵指数 e	e	16 進数表記	e = 9CB975
		p	【出力】 鍵ペア生成機能の出力である素数 p	p	16 進数表記	p = F394 ... 3063
		q	【出力】 鍵ペア生成機能の出力である素数 q	q	16 進数表記	q = CBED ... 8DE7
		n	【出力】 鍵ペア生成機能の出力である法 n	n	16 進数表記	n = C209 ... 3055
		d	【出力】 鍵ペア生成機能の出力であるプライベート指数 d	d	16 進数表記	d = 24A9 ... FDED
	署名生成	ヘッダ	HashAlgorithm	ハッシュ関数	文字列	[HashAlgorithm = SHA256]
			BitLengthOfPlainText	平文のビット長	10 進数表記	[BitLengthOfPlainText = 1024]
			NumberOfPlainTexts	平文の数	10 進数表記	[NumberOfPlainTexts = 2048]
			n	法 n	16 進数表記	n = B373 ... DFE5
			e	公開鍵指数 e	16 進数表記	e = 8bb7 ... b24f
			d	プライベート鍵 d	16 進数表記	d = 0415 ... fca1
		本体*2	COUNT	0 以上 NumberOfPlainTexts 未満の整数	10 進数表記	COUNT = 0
			SHAAlg	ハッシュ関数識別子	文字列	SHAAlg = SHA256
			Msg	入力メッセージ	16 進数表記	Msg = d172 ... b13f
			S	【出力】 署名生成機能の出力であるバイト列	16 進数表記	S = b7f3 ... 9744
署名検証	ヘッダ	HashAlgorithm	ハッシュ関数識別子	ハッシュ関数	文字列	[HashAlgorithm = SHA256]
		BitLengthOfPlainText	平文のビット長		10 進数表記	[BitLengthOfPlainText = 1024]
		NumberOfPlainTexts	平文の数		10 進数表記	[NumberOfPlainTexts = 2048]
		RatioOfInvalidData	署名検証の選択時, 署名検証が不合格になる割合. それ以外は省略.		浮動小数点表記	[RatioOfInvalidData = 0.8]
	本体*3	n	法 n		16 進数表記	n = B373 ... DFE5
		COUNT	0 以上 NumberOfPlainTexts 未満の整数		10 進数表記	COUNT = 0
		SHAAlg	ハッシュ関数識別子	ハッシュ関数	文字列	SHAAlg = SHA256
		e	公開鍵指数 e		16 進数表記	e = 010001
		Msg	入力メッセージ	平文	16 進数表記	Msg = 2f9a ... 3664
		S	署名検証機能の入力であるバイト列	署名	16 進数表記	S = b7f3 ... 9744
		Result	【出力】 署名検証結果. 検証合格の時 P, 不合格の時 F と記述する.		文字列	Result = P

*1 NumberOfKeyPairs 個の各データの組を以下のように記述する.

COUNT = 0	# $i = 0$ のデータの組について記述する.
e = 9CB975	# $i = 0$ に対応して, IUT が生成した (又は前提とする) 公開鍵指数 e .
p = F394 ... 3063	# $i = 0$ に対応して, IUT が生成した素数 p .
q = CBED ... 8DE7	# $i = 0$ に対応して, IUT が生成した素数 q .
n = C209 ... 3055	# $i = 0$ に対応して, IUT が生成した法 n .
d = 24A9 ... FDED	# $i = 0$ に対応して, IUT が生成したプライベート指数 d .
COUNT = 1	# $i = 1$ のデータの組について記述する.
e = 9CB975	# $i = 1$ に対応して, IUT が生成した (又は前提とする) 公開鍵指数 e .
p = F394 ... 3063	# $i = 1$ に対応して, IUT が生成した素数 p .
q = CBED ... 8DE7	# $i = 1$ に対応して, IUT が生成した素数 q .
n = C209 ... 3055	# $i = 1$ に対応して, IUT が生成した法 n .
d = 24A9 ... FDED	# $i = 1$ に対応して, IUT が生成したプライベート指数 d .
:	
:	
COUNT = \langle NumberOfKeyPairs - 1 \rangle	# $i = \langle$ NumberOfKeyPairs - 1 \rangle のデータの組について記述する.
e = 9CB975	# $i = \langle$ NumberOfKeyPairs - 1 \rangle に対応して, IUT が生成した (又は前提とする) 公開鍵指数 e .
p = F394 ... 3063	# $i = \langle$ NumberOfKeyPairs - 1 \rangle に対応して, IUT が生成した素数 p .
q = CBED ... 8DE7	# $i = \langle$ NumberOfKeyPairs - 1 \rangle に対応して, IUT が生成した素数 q .
n = C209 ... 3055	# $i = \langle$ NumberOfKeyPairs - 1 \rangle に対応して, IUT が生成した法 n .
d = 24A9 ... FDED	# $i = \langle$ NumberOfKeyPairs - 1 \rangle に対応して, IUT が生成したプライベート指数 d .

*2 NumberOfPlainTexts 個の各データの組を以下のように記述する.

COUNT = 0	# $i = 0$ のデータの組について記述する.
SHAAlg = SHA256	# $i = 0$ に対応するハッシュ関数識別子を記述する.
Msg = 2f9a ... 3664	# $i = 0$ に対応する入力メッセージ.
S = b7f3 ... 9744	# $i = 0$ に対応して, IUT が生成した署名.
COUNT = 1	# $i = 1$ のデータの組について記述する.
SHAAlg = SHA256	# $i = 1$ に対応するハッシュ関数識別子を記述する.
Msg = 2f9a ... 3664	# $i = 1$ に対応する入力メッセージ.
S = b7f3 ... 9744	# $i = 1$ に対応して, IUT が生成した署名.
:	
:	
COUNT = \langle NumberOfPlainTexts - 1 \rangle	# $i = \langle$ NumberOfPlainTexts - 1 \rangle のデータの組について記述する.
SHAAlg = SHA256	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応するハッシュ関数識別子を記述する.
Msg = 2f9a ... 3664	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応する入力メッセージ.
S = b7f3 ... 9744	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応して, IUT が生成した署名.

*3 NumberOfPlainTexts 個の各データの組を以下のように記述する.

COUNT = 0	# $i = 0$ のデータの組について記述する.
SHAAlg = SHA256	# $i = 0$ に対応するハッシュ関数識別子を記述する.
e = 010001	# $i = 0$ に対応する公開鍵指数 e を記述する.
Msg = 2f9a ... 3664	# $i = 0$ に対応する入力メッセージ.
S = b7f3 ... 9744	# $i = 0$ に対応する署名検証機能の入力であるバイト列.
Result = P	# $i = 0$ に対応して, IUT が出力した署名検証結果.
COUNT = 1	# $i = 1$ のデータの組について記述する.
SHAAlg = SHA256	# $i = 1$ に対応するハッシュ関数識別子を記述する.
e = 010001	# $i = 1$ に対応する公開鍵指数 e を記述する.
Msg = 2f9a ... 3664	# $i = 1$ に対応する入力メッセージ.
S = b7f3 ... 9744	# $i = 1$ に対応する署名検証機能の入力であるバイト列.
Result = P	# $i = 1$ に対応して, IUT が出力した署名検証結果.
:	
:	
COUNT = \langle NumberOfPlainTexts - 1 \rangle	# $i = \langle$ NumberOfPlainTexts - 1 \rangle のデータの組について記述する.
SHAAlg = SHA256	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応するハッシュ関数識別子を記述する.
e = 010001	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応する公開鍵指数 e を記述する.
Msg = 2f9a ... 3664	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応する入力メッセージ.
S = b7f3 ... 9744	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応する署名検証機能の入力であるバイト列.
Result = F	# $i = \langle$ NumberOfPlainTexts - 1 \rangle に対応して, IUT が出力した署名検証結果.