

JCATT ファイルフォーマット仕様書
FIPS 202 に記載されたハッシュ関数

2018 年 8 月

独立行政法人情報処理推進機構

目次

1	はじめに	3
2	FIPS 202 に記載されたハッシュ関数	4
2.1	CAVS 準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)	4
2.1.2	リクエストファイル (*.req)	5
2.1.3	Facts ファイル (*.fax)	6
2.1.4	レスポンスファイル (*.rsp)	7
2.1.5	結果ファイル (*.out)	8

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 FIPS 202 に記載されたハッシュ関数

ハッシュ関数 SHA3-256, SHA3-384, SHA3-512 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。これらのハッシュ関数のファイルフォーマットは, Algorithm Name の他は各ハッシュ関数で同じである。

Algorithm Name は, それぞれ下記の通り。

- SHA3-256
- SHA3-384
- SHA3-512

各表において, 試験方法に関する以下の略語を使用する。

- SMT: Short Messages Test
- SLMT: Selected Long Messages Test
- PGMT: Pseudorandomly Generated Messages Test

試験方法の詳細は, 暗号アルゴリズム実装試験仕様書を参照のこと。

2.1 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, ハッシュ関数識別子として, 表1に記載された表現を用いる。

表1 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
SHA3-256	SHA3-256
SHA3-384	SHA3-384
SHA3-512	SHA3-512

2.1.1 パラメータファイル (*.par)

表2 FIPS 202 に記載されたハッシュ関数

機能	タグ	内容	表記
ハッシュ関数	全体ヘッダ	AlgorithmName	(ハッシュ関数識別子)
		BitOrientedInputCapability	ビット単位での入力する機能の有無 (有:true, 無:false)
		UpperboundOfSLMT	SLMT で使用されるメッセージの最大ビット長を規定するパラメータ
		NumberOfInnerLoopsInPGMT	PGMT の内側ループの回数
		NumberOfOuterLoopsInPGMT	PGMT の外側ループの回数

2.1.2 リクエストファイル (*.req)

表3: FIPS 202 に記載されたハッシュ関数 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—ハッシュ— 上の表記との対応	値の表記	例示
ハッシュ関数	全体ヘッダ		AlgorithmName	ハッシュ関数識別子.		文字列	[AlgorithmName = SHA3-256]
			BitOrientedInputCapability	ビット単位での入力する機能の有無 (有:true, 無:false)		文字列	[BitOrientedInputCapability = true]
			UpperboundOfSLMT	SLMT で使用されるメッセージの最大ビット長を規定するパラメータ		10 進表記	[UpperboundOfSLMT = 100]
			NumberOfInnerLoopsInPGMT	PGMT の内側ループの回数	innerloop	10 進表記	[NumberOfInnerLoopsInPGMT = 1000]
			NumberOfOuterLoopsInPGMT	PGMT の外側ループの回数	outerloop	10 進表記	[NumberOfOuterLoopsInPGMT = 100]
						10 進数	[L = 256]
	SMT	SMTヘッダ *1 *2	COUNT	ハッシュ関数の入力ブロック長 (ビット数) を m として, ・ビット単位での入力する機能有 (bit-oriented) の場合 0 以上 $r + 1$ 未満の整数 ・ビット単位での入力する機能無 (byte-oriented) の場合 0 以上 $r/8 + 1$ 未満の整数	i	10 進表記	COUNT = 0
			Len	ハッシュ関数の入力メッセージのビット長	$i = \langle \text{COUNT の値} \rangle$ に対応する入力メッセージのビット長	10 進表記	Len = 0
			Msg	ハッシュ関数の入力メッセージ	$i = \langle \text{COUNT の値} \rangle$ に対応する入力メッセージ	16 進表記	Msg = 00
			MD	ハッシュ関数の出力であるビット列	$i = \langle \text{COUNT の値} \rangle$ に対応するメッセージダイジェスト	16 進表記	MD = ?
						10 進数	[L = 256]
						10 進表記	COUNT = 0
	SLMT	SLMT本体 *3	COUNT	0 以上 UpperboundOfSLMT 未満の整数	$i - 1$	10 進表記	COUNT = 0
			Len	ハッシュ関数の入力メッセージのビット長	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応する入力メッセージのビット長	10 進表記	Len = 2184
			Msg	ハッシュ関数の入力メッセージ	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応する入力メッセージ	16 進表記	Msg = 2f9a ... 3664
			MD	ハッシュ関数の出力であるビット列	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応するメッセージダイジェスト	16 進表記	MD = ?
						10 進数	[L = 256]
						10 進表記	COUNT = 0
	PGMT	PGMTヘッダ *4	COUNT	0 以上 NumberOfOuterLoopsInPGMT 未満の整数	j	10 進表記	COUNT = 0
			MD	ハッシュ関数の出力であるビット列	$j = \langle \text{COUNT の値} \rangle$ に対応する MD_0	16 進表記	MD = ?
						10 進数	[L = 256]
						10 進表記	COUNT = 0
						10 進数	[L = 256]
						10 進表記	COUNT = 0

*1 ビット単位での入力する機能有 (bit-oriented) の場合, $m + 1$ 個の各データの組を以下のように記述する.

COUNT = 0

Len = 0

Msg = 00

MD = ?

$i = 0$ のデータの組について記述する.
$i = 0$ に対応する入力メッセージのビット長を記述する.
$i = 0$ に対応する入力メッセージを記述する.
$i = 0$ に対応するメッセージダイジェストのプレースホルダ.

COUNT = 1

Len = 1

Msg = 01

MD = ?

$i = 1$ のデータの組について記述する.
$i = 1$ に対応する入力メッセージのビット長を記述する.
$i = 1$ に対応する入力メッセージを記述する.
$i = 1$ に対応するメッセージダイジェストのプレースホルダ.

⋮

COUNT = $\langle r \rangle$

Len = 1088

Msg = dc6f ... c8f2

MD = ?

$i = \langle r \rangle$ のデータの組について記述する.
$i = \langle r \rangle$ に対応する入力メッセージのビット長を記述する.
$i = \langle r \rangle$ に対応する入力メッセージを記述する.
$i = \langle r \rangle$ に対応するメッセージダイジェストのプレースホルダ.

*2 ビット単位での入力する機能無 (byte-oriented) の場合, $r/8 + 1$ 個の各データの組を以下のように記述する.

COUNT = 0

Len = 0

Msg = 00

MD = ?

$i = 0$ のデータの組について記述する.
$i = 0$ に対応する入力メッセージのビット長を記述する.
$i = 0$ に対応する入力メッセージを記述する.
$i = 0$ に対応するメッセージダイジェストのプレースホルダ.

COUNT = 1

Len = 8

Msg = 30

MD = ?

$i = 1$ のデータの組について記述する.
$i = 1$ に対応する入力メッセージのビット長を記述する.
$i = 1$ に対応する入力メッセージを記述する.
$i = 1$ に対応するメッセージダイジェストのプレースホルダ.

⋮

COUNT = $\langle r/8 \rangle$

Len = 1088

Msg = dc6f ... c8f2

MD = ?

$i = \langle r/8 \rangle$ のデータの組について記述する.
$i = \langle r/8 \rangle$ に対応する入力メッセージのビット長を記述する.
$i = \langle r/8 \rangle$ に対応する入力メッセージを記述する.
$i = \langle r/8 \rangle$ に対応するメッセージダイジェストのプレースホルダ.

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

COUNT = 0

Len = 2184

Msg = 2f9a ... 3664

MD = ?

$i = 1$ のデータの組について記述する.
$i = 1$ に対応する入力メッセージのビット長を記述する.
$i = 1$ に対応する入力メッセージを記述する.
$i = 1$ に対応するメッセージダイジェストのプレースホルダ.

COUNT = 1

Len = 3280

Msg = bbeef ... 1429

MD = ?

$i = 2$ のデータの組について記述する.
$i = 2$ に対応する入力メッセージのビット長を記述する.
$i = 2$ に対応する入力メッセージを記述する.
$i = 2$ に対応するメッセージダイジェストのプレースホルダ.

⋮

COUNT = $\langle \text{UpperboundOfSLMT} - 1 \rangle$

Len = 110688

Msg = dc6f ... c8f2

MD = ?

$i = \langle \text{UpperboundOfSLMT} \rangle$ のデータの組について記述する.
$i = \langle \text{UpperboundOfSLMT} \rangle$ に対応する入力メッセージのビット長を記述する.
$i = \langle \text{UpperboundOfSLMT} \rangle$ に対応する入力メッセージを記述する.
$i = \langle \text{UpperboundOfSLMT} \rangle$ に対応するメッセージダイジェストのプレースホルダ.

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

COUNT = 0

MD = ?

$j = 0$ のデータの組について記述する.
$j = 0$ に対応するメッセージダイジェストのプレースホルダ.

COUNT = 1

MD = ?

$j = 1$ のデータの組について記述する.
$j = 1$ に対応するメッセージダイジェストのプレースホルダ.

⋮

COUNT = $\langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$

MD = ?

$j = \langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$ のデータの組について記述する.
$j = \langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$ に対応するメッセージダイジェストのプレースホルダ.

2.1.3 Facts ファイル (*.fax)

表4: FIPS 202 に記載されたハッシュ関数 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—ハッシュ— 上の表記との対応	値の表記	例示
ハッシュ関数	全体ヘッダ		AlgorithmName	ハッシュ関数識別子.		文字列	[AlgorithmName = SHA3-256]
			BitOrientedInputCapability	ビット単位での入力する機能の有無 (有:true, 無:false)		文字列	[BitOrientedInputCapability = true]
			UpperboundOfSLMT	SLMT で使用されるメッセージの最大ビット長を規定するパラメータ		10 進表記	[UpperboundOfSLMT = 100]
			NumberOfInnerLoopsInPGMT	PGMT の内側ループの回数	innerloop	10 進表記	[NumberOfInnerLoopsInPGMT = 1000]
			NumberOfOuterLoopsInPGMT	PGMT の外側ループの回数	outerloop	10 進表記	[NumberOfOuterLoopsInPGMT = 100]
	SMT	SMTヘッダ	〈メッセージダイジェストのビット数〉			10 進数	[L = 256]
			COUNT	ハッシュ関数の入力ブロック長 (ビット数) を m として, ・ビット単位での入力する機能有 (bit-oriented) の場合 0 以上 $r + 1$ 未満の整数 ・ビット単位での入力する機能無 (byte-oriented) の場合 0 以上 $r/8 + 1$ 未満の整数	i	10 進表記	COUNT = 0
		SMT本体 *1 *2	Len	ハッシュ関数の入力メッセージのビット長	$i = \langle \text{COUNT の値} \rangle$ に対応する入力メッセージのビット長	10 進表記	Len = 0
			Msg	ハッシュ関数の入力メッセージ	$i = \langle \text{COUNT の値} \rangle$ に対応する入力メッセージ	16 進表記	Msg = 00
			MD	ハッシュ関数の出力であるビット列	$i = \langle \text{COUNT の値} \rangle$ に対応するメッセージダイジェスト	16 進表記	MD = e3b0 ... b855
	SLMT	SLMTヘッダ	〈メッセージダイジェストのビット数〉			10 進数	[L = 256]
			COUNT	0 以上 UpperboundOfSLMT 未満の整数	$i - 1$	10 進表記	COUNT = 0
		SLMT本体 *3	Len	ハッシュ関数の入力メッセージのビット長	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応する入力メッセージのビット長	10 進表記	Len = 2184
			Msg	ハッシュ関数の入力メッセージ	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応する入力メッセージ	16 進表記	Msg = 2f9a ... 3664
			MD	ハッシュ関数の出力であるビット列	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応するメッセージダイジェスト	16 進表記	MD = 12df ... 8e43
	PGMT	PGMTヘッダ	Seed	PGMT の Seed	Seed	16 進表記	Seed = 50c2 ... a265
			COUNT	0 以上 NumberOfOuterLoopsInPGMT 未満の整数	j	10 進表記	COUNT = 0
		PGMT本体 *4	MD	ハッシュ関数の出力であるビット列	$j = \langle \text{COUNT の値} \rangle$ に対応する MD_0	16 進表記	MD = 0506 ... 1746

*1 ビット単位での入力する機能有 (bit-oriented) の場合, $m + 1$ 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.
Len = 0	# $i = 0$ に対応する入力メッセージのビット長を記述する.
Msg = 00	# $i = 0$ に対応する入力メッセージを記述する.
MD = e3b0 ... b855	# $i = 0$ に対応するメッセージダイジェストの期待値を記述する.

COUNT = 1	# $i = 1$ のデータの組について記述する.
Len = 1	# $i = 1$ に対応する入力メッセージのビット長を記述する.
Msg = 01	# $i = 1$ に対応する入力メッセージを記述する.
MD = 0a2c ... ab38	# $i = 1$ に対応するメッセージダイジェストの期待値を記述する.

⋮

COUNT = $\langle r \rangle$	# $i = \langle r \rangle$ のデータの組について記述する.
Len = 1088	# $i = \langle r \rangle$ に対応する入力メッセージのビット長を記述する.
Msg = dc6f ... c8f2	# $i = \langle r \rangle$ に対応する入力メッセージを記述する.
MD = 4b53 ... 275b	# $i = \langle r \rangle$ に対応するメッセージダイジェストの期待値を記述する.

*2 ビット単位での入力する機能無 (byte-oriented) の場合, $r/8 + 1$ 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.
Len = 0	# $i = 0$ に対応する入力メッセージのビット長を記述する.
Msg = 00	# $i = 0$ に対応する入力メッセージを記述する.
MD = e3b0 ... b855	# $i = 0$ に対応するメッセージダイジェストの期待値を記述する.

COUNT = 1	# $i = 1$ のデータの組について記述する.
Len = 8	# $i = 1$ に対応する入力メッセージのビット長を記述する.
Msg = d6	# $i = 1$ に対応する入力メッセージを記述する.
MD = 0a2c ... ab38	# $i = 1$ に対応するメッセージダイジェストの期待値を記述する.

⋮

COUNT = $\langle r/8 \rangle$	# $i = \langle r/8 \rangle$ のデータの組について記述する.
Len = 1088	# $i = \langle r/8 \rangle$ に対応する入力メッセージのビット長を記述する.
Msg = dc6f ... c8f2	# $i = \langle r/8 \rangle$ に対応する入力メッセージを記述する.
MD = 4b53 ... 275b	# $i = \langle r/8 \rangle$ に対応するメッセージダイジェストの期待値を記述する.

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

COUNT = 0	# $i = 1$ のデータの組について記述する.
Len = 2184	# $i = 1$ に対応する入力メッセージのビット長を記述する.
Msg = 2f9a ... 3664	# $i = 1$ に対応する入力メッセージを記述する.
MD = 12df ... 8e43	# $i = 1$ に対応するメッセージダイジェストの期待値を記述する.

COUNT = 1	# $i = 2$ のデータの組について記述する.
Len = 3280	# $i = 2$ に対応する入力メッセージのビット長を記述する.
Msg = bbef ... 1429	# $i = 2$ に対応する入力メッセージを記述する.
MD = edcb ... 4ddd	# $i = 2$ に対応するメッセージダイジェストの期待値を記述する.

⋮

COUNT = $\langle \text{UpperboundOfSLMT} - 1 \rangle$	# $i = \langle \text{UpperboundOfSLMT} \rangle$ のデータの組について記述する.
Len = 110688	# $i = \langle \text{UpperboundOfSLMT} \rangle$ に対応する入力メッセージのビット長を記述する.
Msg = c354 ... 7fef	# $i = \langle \text{UpperboundOfSLMT} \rangle$ に対応する入力メッセージを記述する.
MD = 9d88 ... 11a1	# $i = \langle \text{UpperboundOfSLMT} \rangle$ に対応するメッセージダイジェストの期待値を記述する.

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

COUNT = 0	# $j = 0$ のデータの組について記述する.
MD = 0506 ... 1746	# $j = 0$ に対応するメッセージダイジェストの期待値を記述する.

COUNT = 1	# $j = 1$ のデータの組について記述する.
MD = 263e ... 1a33	# $j = 1$ に対応するメッセージダイジェストの期待値を記述する.

⋮

COUNT = $\langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$	# $j = \langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$ のデータの組について記述する.
MD = 8d59 ... 13f9	# $j = \langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$ に対応するメッセージダイジェストの期待値を記述する.

2.1.4 レスponsファイル (*.rsp)

表5: FIPS 202 に記載されたハッシュ関数 レスpons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—ハッシュ— 上の表記との対応	値の表記	例示
ハッシュ関数	全体ヘッダ		AlgorithmName	ハッシュ関数識別子.		文字列	[AlgorithmName = SHA3-256]
			BitOrientedInputCapability	ビット単位での入力する機能の有無 (有:true, 無:false)		文字列	[BitOrientedInputCapability = true]
			UpperboundOfSLMT	SLMT で使用されるメッセージの最大ビット長を規定するパラメータ		10 進表記	[UpperboundOfSLMT = 100]
			NumberOfInnerLoopsInPGMT	PGMT の内側ループの回数	innerloop	10 進表記	[NumberOfInnerLoopsInPGMT = 1000]
			NumberOfOuterLoopsInPGMT	PGMT の外側ループの回数	outerloop	10 進表記	[NumberOfOuterLoopsInPGMT = 100]
	SMT	SMTヘッダ	〈メッセージダイジェストのビット数〉			10 進数	[L = 256]
			COUNT	ハッシュ関数の入力ブロック長 (ビット数) を m として, ・ビット単位での入力する機能有 (bit-oriented) の場合 0 以上 $r + 1$ 未満の整数 ・ビット単位での入力する機能無 (byte-oriented) の場合 0 以上 $r/8 + 1$ 未満の整数	i	10 進表記	COUNT = 0
		SMT本体 *1 *2	Len	ハッシュ関数の入力メッセージのビット長	$i = \langle \text{COUNT の値} \rangle$ に対応する入力メッセージのビット長	10 進表記	Len = 0
			Msg	ハッシュ関数の入力メッセージ	$i = \langle \text{COUNT の値} \rangle$ に対応する入力メッセージ	16 進表記	Msg = 00
			MD	【出力】ハッシュ関数の出力であるビット列	$i = \langle \text{COUNT の値} \rangle$ に対応するメッセージダイジェスト	16 進表記	MD = e3b0 ... b855
			〈メッセージダイジェストのビット数〉			10 進数	[L = 256]
			COUNT	0 以上 UpperboundOfSLMT 未満の整数	$i - 1$	10 進表記	COUNT = 0
	SLMT	SLMT本体 *3	Len	ハッシュ関数の入力メッセージのビット長	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応する入力メッセージのビット長	10 進表記	Len = 2184
			Msg	ハッシュ関数の入力メッセージ	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応する入力メッセージ	16 進表記	Msg = 2f9a ... 3664
			MD	【出力】ハッシュ関数の出力であるビット列	$i = \langle \text{COUNT} + 1 \text{ の値} \rangle$ に対応するメッセージダイジェスト	16 進表記	MD = 12df ... 8e43
			PGMTヘッダ				
			Seed	PGMT の Seed	Seed	16 進表記	Seed = 50c2 ... a265
	PGMT	PGMT *4	COUNT	0 以上 NumberOfOuterLoopsInPGMT 未満の整数	j	10 進表記	COUNT = 0
			MD	【出力】ハッシュ関数の出力であるビット列	$j = \langle \text{COUNT の値} \rangle$ に対応する MD ₀	16 進表記	MD = 0506 ... 1746

*1 ビット単位での入力する機能有 (bit-oriented) の場合, $m + 1$ 個の各データの組を以下のように記述する.

COUNT = 0
Len = 0
Msg = 00
MD = e3b0 ... b855

$i = 0$ のデータの組について記述する.
$i = 0$ に対応する入力メッセージのビット長を記述する.
$i = 0$ に対応する入力メッセージを記述する.
$i = 0$ に対応して生成されたメッセージダイジェスト.

COUNT = 1
Len = 1
Msg = 01
MD = 0a2c ... ab38

$i = 1$ のデータの組について記述する.
$i = 1$ に対応する入力メッセージのビット長を記述する.
$i = 1$ に対応する入力メッセージを記述する.
$i = 1$ に対応して生成されたメッセージダイジェスト.

⋮
COUNT = $\langle r \rangle$
Len = 1088
Msg = dc6f ... c8f2
MD = 4b53 ... 275b

$i = \langle r \rangle$ のデータの組について記述する.
$i = \langle r \rangle$ に対応する入力メッセージのビット長を記述する.
$i = \langle r \rangle$ に対応する入力メッセージを記述する.
$i = \langle r \rangle$ に対応して生成されたメッセージダイジェスト.

*2 ビット単位での入力する機能無 (byte-oriented) の場合, $r/8 + 1$ 個の各データの組を以下のように記述する.

COUNT = 0
Len = 0
Msg = 00
MD = e3b0 ... b855

$i = 0$ のデータの組について記述する.
$i = 0$ に対応する入力メッセージのビット長を記述する.
$i = 0$ に対応する入力メッセージを記述する.
$i = 0$ に対応して生成されたメッセージダイジェスト.

COUNT = 1
Len = 8
Msg = d6
MD = 0a2c ... ab38

$i = 1$ のデータの組について記述する.
$i = 1$ に対応する入力メッセージのビット長を記述する.
$i = 1$ に対応する入力メッセージを記述する.
$i = 1$ に対応して生成されたメッセージダイジェスト.

⋮
COUNT = $\langle r/8 \rangle$
Len = 1088
Msg = dc6f ... c8f2
MD = 4b53 ... 275b

$i = \langle r/8 \rangle$ のデータの組について記述する.
$i = \langle r/8 \rangle$ に対応する入力メッセージのビット長を記述する.
$i = \langle r/8 \rangle$ に対応する入力メッセージを記述する.
$i = \langle r/8 \rangle$ に対応して生成されたメッセージダイジェスト.

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

COUNT = 0
Len = 2184
Msg = 2f9a ... 3664
MD = 12df ... 8e43

$i = 1$ のデータの組について記述する.
$i = 1$ に対応する入力メッセージのビット長を記述する.
$i = 1$ に対応する入力メッセージを記述する.
$i = 1$ に対応して生成されたメッセージダイジェスト.

COUNT = 1
Len = 3280
Msg = bbef ... 1429
MD = edcb ... 4ddd

$i = 2$ のデータの組について記述する.
$i = 2$ に対応する入力メッセージのビット長を記述する.
$i = 2$ に対応する入力メッセージを記述する.
$i = 2$ に対応して生成されたメッセージダイジェスト.

⋮
COUNT = $\langle \text{UpperboundOfSLMT} - 1 \rangle$
Len = 110688
Msg = c354 ... 7fef
MD = 9d88 ... 11a1

$i = \langle \text{UpperboundOfSLMT} \rangle$ のデータの組について記述する.
$i = \langle \text{UpperboundOfSLMT} \rangle$ に対応する入力メッセージのビット長を記述する.
$i = \langle \text{UpperboundOfSLMT} \rangle$ に対応する入力メッセージを記述する.
$i = \langle \text{UpperboundOfSLMT} \rangle$ に対応して生成されたメッセージダイジェスト.

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

COUNT = 0
MD = 0506 ... 1746

$j = 0$ のデータの組について記述する.
$j = 0$ に対応して生成されたメッセージダイジェスト.

COUNT = 1
MD = 263e ... 1a33

$j = 1$ のデータの組について記述する.
$j = 1$ に対応して生成されたメッセージダイジェスト.

⋮
COUNT = $\langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$
MD = 8d59 ... 13f9

$j = \langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$ のデータの組について記述する.
$j = \langle \text{NumberOfOuterLoopsInPGMT} - 1 \rangle$ に対応して生成されたメッセージダイジェスト.

2.1.5 結果ファイル (*.out)

表6: FIPS 202 に記載されたハッシュ関数 結果ファイル

機能	分類		タグ	内容	暗号アルゴリズム実装試験 仕様書—ハッシュ— 上の表記との対応	値の表記	例示
ハッシュ関数	全体ヘッダ		AlgorithmName	ハッシュ関数識別子.		文字列	[AlgorithmName = SHA3-256]
			BitOrientedInputCapability	ビット単位での入力する機能の有無 (有:true, 無:false)		文字列	[BitOrientedInputCapability = true]
			UpperboundOfSLMT	SLMT で使用されるメッセージの最大 ビット長を規定するパラメータ		10 進表記	[UpperboundOfSLMT = 100]
			NumberOfInnerLoopsInPGMT	PGMT の内側ループの回数	innerloop	10 進表記	[NumberOfInnerLoopsInPGMT = 1000]
			NumberOfOuterLoopsInPGMT	PGMT の外側ループの回数	outerloop	10 進表記	[NumberOfOuterLoopsInPGMT = 100]
	SMT	SMT ヘッダ	〈メッセージダイジェストのバイト 数〉			10 進数	[L = 256]
			〈 Results 〉	OK 又は NG		文字列	OK
	SLMT	SLMT ヘッダ	〈メッセージダイジェストのバイト 数〉			10 進数	[L = 256]
			〈 Results 〉	OK 又は NG		文字列	OK
	PGMT	PGMT ヘッダ	Seed	PGMT の Seed	Seed	16 進表記	Seed = 50c2 ... a265
			〈 Results 〉	OK 又は NG		文字列	OK

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。