JCATTファイルフォーマット仕様書

ハッシュ関数

２００８年1月１８日

独立行政法人　情報処理推進機構

目 次

1　はじめに

暗号アルゴリズム試験ツール (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，試験対象暗号モジュール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

- [　] で囲まれた “タグ” の次の行に値を記述する．
- タグは各ファイルフォーマットに記述した順番通りに記述すること．
- レスポンスファイルにおいては,【出力】と記述したタグが，試験対象モジュール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　ハッシュ関数

ハッシュ関数SHA-1，SHA-224，SHA-256，SHA-384，SHA-512，RIPEMD-160 の暗号モジュール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．

- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512
- RIPEMD-160

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

これらの試験方法の詳細は，暗号アルゴリズム試験仕様書を参照のこと．

2.1 パラメータファイル (*.par)

表 1: ハッシュ関数パラメータファイル

機能	タグ	内容
ハッシュ関数	[Algorithm Name]	(ハッシュ関数名)
	[Function Name]	Hash
	[Seed]	SMT および SLMT においてランダムメッセージを生成するための擬似乱数生成関数用シード値
	[Bitlength of Seed]	Seed のビット長
	[Data Format]	メッセージデータが byte oriented であることを示す識別子．M_Hash_Byte と記述すること．
	[Upperbound of SLMT]	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．“メッセージの最大ビット長” = [Upperbound of SLMT]× “ハッシュ関数のブロック長 (ビット)” となる．ハッシュ関数のブロック長は，SHA-1，SHA-224，SHA-256，RIPEMD-160 が 512 ビット，SHA-384 と SHA-512 が 1,024 ビットである．
	[Number of Inner-loop for PGMT]	PGMT の内側ループの回数
	[Number of Outer-loop for PGMT]	PGMT の外側ループの回数
	[Initial Data for PGMT]	PGMT 用初期値
	[Bitlength of Initial Data for PGMT]	PGMT 用初期値のビット長

2.2 リクエストファイル (*.req)

表 2: ハッシュ関数リクエストファイル

機能	タグ	内容
ハッシュ関数	[Algorithm Name]	(ハッシュ関数名)
	[Function Name]	Hash
	[Data Format]	メッセージデータが byte oriented であることを示す識別子．M_Hash_Byte と記述すること．
	[Data of SMT]	SMT 用メッセージ
	[Upperbound of SLMT]	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．“メッセージの最大ビット長” = [Upperbound of SLMT]× “ハッシュ関数のブロック長 (ビット)” となる．ハッシュ関数のブロック長は，SHA-1，SHA-224，SHA-256，RIPEMD-160 が 512 ビット，SHA-384 と SHA-512 が 1,024 ビットである．
	[Data of SLMT]	SLMT 用メッセージ
	[Number of Inner-loop for PGMT]	PGMT の内側ループの回数
	[Number of Outer-loop for PGMT]	PGMT の外側ループの回数
	[Initial Data for PGMT]	PGMT 用初期値
	[Bitlength of Initial Data for PGMT]	PGMT 用初期値のビット長

2.3 Facts ファイル (*.fax)

表 3: ハッシュ関数 Facts ファイル

機能	タグ	内容
ハッシュ関数	[Algorithm Name]	(ハッシュ関数名)
	[Function Name]	Hash
	[Data Format]	メッセージデータが byte oriented であることを示す識別子．M_Hash_Byte と記述すること．
	[Data of SMT]	SMT 用メッセージ
	[Hash Value of SMT]	SMT で生成されたハッシュ値
	[Upperbound of SLMT]	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．“メッセージの最大ビット長” = [Upperbound of SLMT]× “ハッシュ関数のブロック長 (ビット)” となる．ハッシュ関数のブロック長は，SHA-1，SHA-224，SHA-256，RIPEMD-160 が 512 ビット，SHA-384 と SHA-512 が 1,024 ビットである．
	[Data of SLMT]	SLMT 用メッセージ
	[Hash Value of SLMT]	SLMT で生成されたハッシュ値
	[Number of Inner-loop for PGMT]	PGMT の内側ループの回数
	[Number of Outer-loop for PGMT]	PGMT の外側ループの回数
	[Initial Data for PGMT]	PGMT 用初期値
	[Bitlength of Initial Data for PGMT]	PGMT 用初期値のビット長
	[Hash Value of PGMT]	PGMT で生成されたハッシュ値

2.4 レスポンスファイル (*.rsp)

表 4: ハッシュ関数レスポンスファイル

機能	タグ	内容
ハッシュ関数	[Algorithm Name]	(ハッシュ関数名)
	[Function Name]	Hash
	[Data Format]	メッセージデータが byte oriented であることを示す識別子．M_Hash_Byte と記述すること．
	[Data of SMT]	SMT 用メッセージ
	[Hash Value of SMT]	【出力】SMT で生成されたハッシュ値
	[Upperbound of SLMT]	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．“メッセージの最大ビット長” = [Upperbound of SLMT]× “ハッシュ関数のブロック長 (ビット)” となる．ハッシュ関数のブロック長は，SHA-1，SHA-224，SHA-256，RIPEMD-160 が 512 ビット，SHA-384 と SHA-512 が 1,024 ビットである．
	[Data of SLMT]	SLMT 用メッセージ
	[Hash Value of SLMT]	【出力】SLMT で生成されたハッシュ値
	[Number of Inner-loop for PGMT]	PGMT の内側ループの回数
	[Number of Outer-loop for PGMT]	PGMT の外側ループの回数
	[Initial Data for PGMT]	PGMT 用初期値
	[Bitlength of Initial Data for PGMT]	PGMT 用初期値のビット長
	[Hash Value of PGMT]	【出力】PGMT で生成されたハッシュ値

2.5 結果ファイル (*.out)

表 5: ハッシュ関数結果ファイル

タグ	内容
[Algorithm Name]	暗号名
[Function Name]	試験対象機能名
[Results]	試験結果

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タグ名と，そのタグ内の何番目 (No.，#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．