平文		
short 2bytes	暗号化したアプリケーションのバージョン（4xxxx）	2
char[1]	ミスタイプ回数制限（デフォルト：3）	1
char[1]	ファイルを破壊するか否か（デフォルト：false=0）	1
char[16] 16 bytes	ファイル・シグニチャ ="_AttacheCaseData" || "_Atc_Broken_Data" || "_AttacheCase_Rsa" = ブロック暗号 or 破壊されたデータ or 公開鍵暗号	16
int 4bytes	暗号化データバージョン（=140） 105 < x < 200 ※130 : ver.3; 140 : ver.4	4 総計：24 bytes
int 4 bytes	暗号化前のヘッダーデータサイズ	4
byte[16] 16 bytes	GUID	16
byte[8] 8 bytes	ランダムソルト →**Rfc2898DeriveBytesクラスから「key」と「IV」を導出する** 両暗号に使用	8 総計：52 bytes
byte[256] 256 bytes	公開鍵暗号RSA（"_AttacheCase_Rsa"）のときは、ここに暗号化されたパスワードを格納す	256 総計：(308 bytes)
暗号化		
byte[4]	トークン（="atc4"）	4
byte[ ] 可変長	【ヘッダーデータ（暗号化）】 **AesManaged （キー:256bit, ブロック:128bit）CBC、PKCS7** [シリアルデータとして保存する] FileNameSize(2byte) | FilePath(可変) | FileSize(Int64) | FileAttr(int) | UTC更新日時(Int64) | UTC生成日時(Int64) | FileSize==0 ? "" : MD5(16byte)※ | … … … … ※データ改竄チェックではなく、データ破損チェック程度なら低負荷のMD5で十分と判断。	暗号化前のヘッダーサイズを
byte[] 可変長	パディング（0～15byte）	
byte[] 可変長	【本体データ（暗号化）】 **AesManaged （キー:256bit, ブロック:128bit）CBC、PKCS7** **＋** **DeflateStream(.NET Framework 4.5以降はzlib)** … … ※今まではRijndaelManagedのキー256bit, ブロック256bitを使ってきたが、 他へ移植するときの取り回しの良さから、事実上の標準であるAES256を選択。	
byte[] 可変長	パディング（0～15byte）	