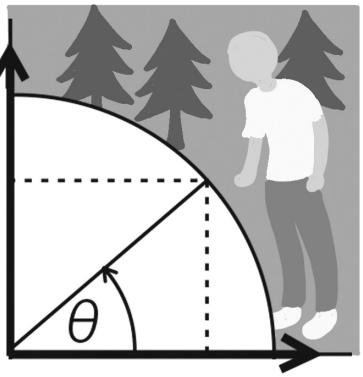
画像や音で理解する数学・物理の世界

数式の歌を聴け



第5回　RSA暗号の応用例…画像の暗号化と電子署名

宮田 賢一

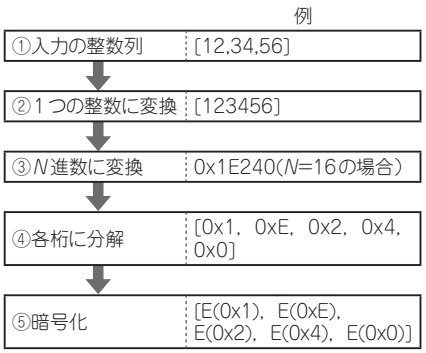


図1　大きなデータの暗号化の手順

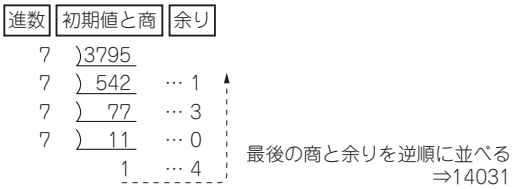


図2　進数変換のプロセス

前回までで，RSA暗号の基本的な考え方と計算方法について理解できたと思います．今回は，実用に近い例として暗号化と復号を可視化します．またRSA暗号のもう1つの応用例である電子署名も取り上げます．使用する言語はPython，開発環境はGoogle Colaboratoryを想定しています．

RSA暗号応用例1…画像の暗号化

● 暗号化には素数と進数変換を使う

RSA暗号は入力の数値を素数の積Nで割ることで暗号化したデータを計算します．つまり入力は必ずN未満でなければなりません．そうしないと復号したときに複数の平文候補が発生してしまうからです．

そこで任意の長さを持つ整数列を暗号化する場合，その整数列全体が1つの巨大な整数であると見なして，これをN進数で表した場合の各桁の数値を暗号化することで暗号文を求めることとします．N進数の各桁の数値は$0 \sim N-1$のN通りなので，最も無駄なく暗号文を求められます(**図1**)．

● 画像を暗号化するプログラム

画像をRSA暗号で暗号化するプログラムを**リスト1**に示します．

▶7～16行目

進数変換を行う関数です．src_base進数で表された数値列をdest_base進数で表された数値列に変換します．内部では一度，入力数値列を1つの整数に変換します(9～10行目)．余りを求める手順を**図2**に示します．数値をNで割って商と余りを求めていきます．割れなくなったら最後の商と余りを逆順に並べると，N進数で表記した場合の各桁の数字を求められます．これを実装したものが12～15行目です．

▶19～34行目

暗号化の対象とする画像を作成します．前回と同じくPython Image Library(Pillow)を使用します．三角関数を使って，2つの同心円に内接する正五角形の頂点を求め，それらの点を結んで星形を描画しています(コラム1)．

▶40～50行目

2つの素数3571と3559を使って公開鍵と秘密鍵を求めます．

▶53行目

入力画像のピクセル・データから，ビッグ・エンディアンの32バイト整数列を作成します．この数列は2^{32}進数(約40億)で表した場合の各桁の数列とみなせます．

▶55行目

入力データ先頭にダミーとして非ゼロの値を挿入します．入力画像の先頭部分が黒の場合，値としてはゼロが並んでいることになります．これをそのまま整数変換すると，黒部分のデータが存在しない場合と区別できません．それを防ぐためのダミー・データです．

▶58～59行目

2^{32}進数の数値列をN進数($3571 \times 3559 = 12709189$進数)の数値列に変換した上で暗号化を適用します．

プログラムは本誌サポート・ページから入手できます．
https://interface.cqpub.co.jp/2301py/