RSA 暗号の暗号化と復号を体験する教材の開発

情報科　山　口　健　二

1. はじめに

暗号は現在のインターネット通信において，安全にデータをやり取りするために用いられる重要な技術である。この技術は，数学における困難さの理論（例えば，2 つの素数からなる大きな合成数を素因数分解する困難さ）を背景として実現される技術であり，暗号は数学や情報を学ぶ上で理想的な教材であるといえる。高校の情報の教科書においては，暗号の理論的側面の説明が十分にカバーされていないため，この技術を学ぶための教材を開発することには，意義があるといえる。

本論考では，Excel で RSA 暗号の暗号化と復号を実装した方法について述べる。これは，お茶の水女子大学の理系女性教育開発共同機構（2022 年 4 月から理系女性育成啓発研究所に改組予定）の支援を受け，2020 年 4 月から 2022 年 3 月にかけて，「暗号技術を学ぶアプリケーションの開発」と「暗号解読に挑戦する体験型教材の開発」でアプリケーションとしても実装している。

2. 新学習指導要領と暗号技術の位置付け

2022 年度から高校の教科「情報」は，新学習指導要領に基づき，これまでの「社会と情報」と「情報の科学」の二つの科目が融合する形で「情報Ⅰ」が新設され，さらに「情報Ⅰ」の内容を発展する形で「情報Ⅱ」が新設される。これらの新しい科目の中で暗号技術に関連する内容があるか確認した。すると，高等学校情報科「情報Ⅰ」教員研修用教材によれば，「第 4 章　情報通信ネットワークとデータの活用」の「情報通信ネットワークの仕組みと役割」の「情報セキュリティ」おいて，取り扱われている。しかし，認証方式と暗号化方式並びに暗号化アルゴリズムの名称と対応については説明されているものの，暗号アルゴリズムの中身については，言及されていない。

実際に暗号アルゴリズムについて，板書などを用いて説明する場合，後述するシーザー暗号などといった，仕組みが簡易な暗号アルゴリズムについて説明が容易であるが，RSA 暗号といった，膨大な指数計算が必要となる暗号アルゴリズムについては説明が難解で，実際の暗号化や復号の計算もコンピューターの力を借りなければならない。また，「情報Ⅰ」は，これまでの「社会と情報」と「情報の科学」の内容を融合したものであるため，授業時間としても暗号アルゴリズムの説明の時間が十分に確保できないかもしれない。これらから推察するに，多くの学校において暗号の説明が時間の都合で省略される可能性が高い。逆を言えば，暗号技術に関する教材を開発することは価値が高いともいえる。

以下，暗号技術について概要を説明する。まず暗号プロトコルの要素として，送信者と受信者，暗号化と復号，平文と暗号文，暗号化と復号で使用される鍵（数値）がある。

そして暗号は大きく分けて「共通鍵暗号方式」と「公開鍵暗号方式」に分類される。

共通鍵暗号方式では，暗号化と復号に同じ鍵（数値）が使われる。その例として教科書でもよく紹介されるのが，古代ローマのジュリアス・シーザーが使ったと言われる「シーザー暗号」である［1］。これは，例えば，送信者と受信者が同じ鍵（ここでは3とする）を他人に知られることなく秘密裏に共有するとする。そして，HELLOという文字列を送る際に，送信者は鍵3を使って暗号化する。具体的には，各文字を鍵の数だけ右シフトさせる。すなわち，H → I → J → K，E → F → G → H，L → M → N → O，L → M → N → O，O → P → Q → R させることで平文HELLOを暗号文KHOORに暗号化する。この暗号文KHOORが通信路（この通信路は誰でも盗聴できる前提のものである）を介して受信者に伝わる。暗号文を受け取った受信者は，KHOORを鍵の数だけ左シフトする。すると，HELLOに復号することができる。しかしこれはよく考え見ると，大文字のアルファベットだけであれば，暗号文が平文と同じ場合も考えると，26通りのシフトをすべて試せば，暗号文が解読されてしまうという弱点がある。ところで，この弱点を解消した共通鍵暗号方式として「単換字暗号」が存在する。これは表1のような換字表を用意し，その換字表に従って，平文から暗号文を作るというものである。これであれば，大文字のアルファベットだけであっても，26! 通り（およそ 4×10^{26} 通り）もの換字表が存在することになり，解読の難易度は一気に上がる。ただし，シーザー暗号においても言えるのだが，暗号文に登場する各文字の登場回数の統計を取ることによる頻度分析からある程度推測することが可能である。例えば，元の文章が英文であればa，i，e，t，n，sといったアルファベットは比較的出現回数が多い。よって，もし暗号文でxの登場回数が多ければ，そのいずれかである可能性が高いことになる。また，それ以外にも，事前に鍵を共有できない場合，共通鍵暗号方式は使用することができない。

表1　換字表

変換前	変換後
A	K
B	Y
C	Q
D	B
E	A
F	I
G	L
H	W
I	E
J	O
K	D
L	M
M	Z
N	J
O	T
P	G
Q	F
R	V
S	C
T	X
U	N
V	P
W	U
X	H
Y	S
Z	R

一方，公開鍵暗号方式は，事前に鍵を共有する必要がない暗号方式で，暗号化に使用する鍵（数値）と復号に使用する鍵（数値）が違うものである。暗号化には（受信者が公にしている）受信者の公開鍵を使用し，復号には（受信者だけが秘密裏に知っている）受信者の秘密鍵を使用する。公開鍵を使って誰でも平文を暗号化して暗号文を作成はできるが，暗号文を平文に復号するには秘密鍵が必要となる。そして，この秘密鍵を知っているのは受信者だけである（公開鍵だけでは暗号文を作ることはできるが解読することはできない）。そのため，基礎となる理論は1976年にDiffieとHellmanによって発表されたDiffie-Hellman鍵交換である［2］。その後，公開鍵暗号方式として，Rivest，Shamir，AdlmanによってRSA暗号が発表された（図1）。

RSA暗号の安全性は，素因数分解の難しさに依存している。とい

うのも，先ほど，公開鍵だけでは暗号文を解読することはできないと述べたが，公開鍵（e, N）から秘密鍵（d）を計算することはできるのだが，d を計算するためには N を素因数分解した p と q が必要になる。N が大きな数の場合，p と q を見つけるには相当の時間（試行回数）がかかる。よって RSA 暗号は計算量的安全性を持つといえる。しかし，最近のコンピューターの性能の向上などにより，より大きな数値の N を使用しなければならずそのため，暗号化や復号にかかる処理が長くなったり，また素因数分解の難しさに依存しているため，素因数分解を簡単に解くことができる解法が発見された場合は，暗号の安全性が脅かされたりするという問題がある。

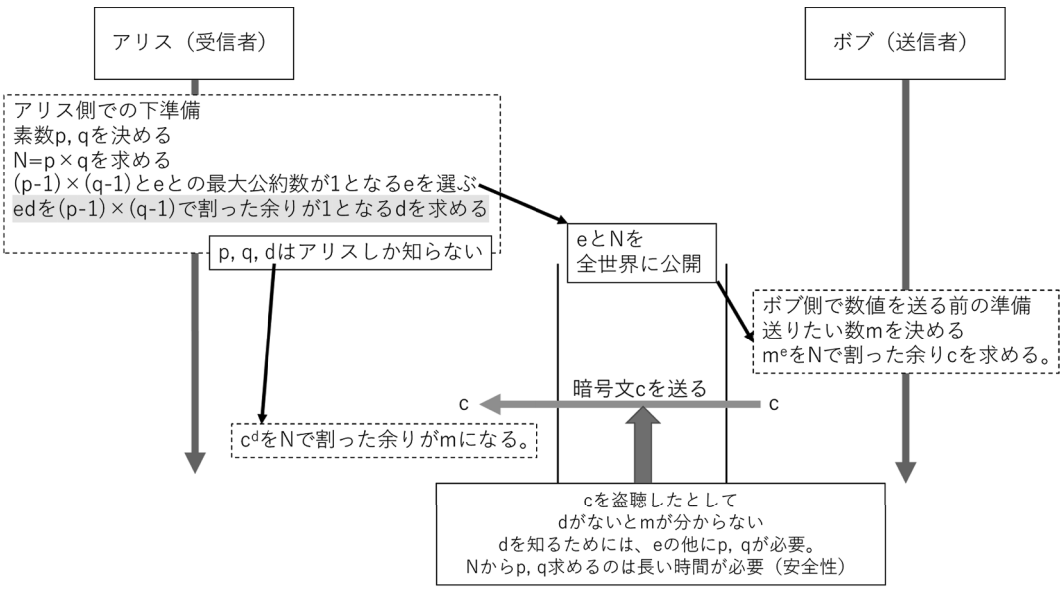

図 1：RSA 暗号のアルゴリズム

また，前述した共通鍵暗号方式と公開鍵暗号方式は組み合わせて使用することもできる。このような暗号はハイブリッド暗号方式とよばれる。例えば，インターネットショッピングで使われる暗号化通信は，SSL/TLS（Secure Sockets Layer/Transport Layer Security）[4] という規格のハイブリッド暗号が使われている。これは，共通鍵暗号方式のメリットである，高速に暗号化や復号が行えるという点と，公開鍵暗号方式のメリットである，送信者と受信者で事前に鍵を共有する必要がないという点を利用したものである。具体的には，まず，公開鍵暗号方式を用いて共通鍵暗号方式の鍵を送信者と受信者で共有する。その後，その鍵を使って共通鍵暗号方式で通信をするというものである。

3. RSA 暗号を Excel で実装する

図 1 で紹介した RSA 暗号の暗号化と復号のアルゴリズムを実現する際に問題になるのが，各変数の桁数である。Excel の数値は，IEEE 754 仕様に準拠しており，16 桁以降の桁が 0 になる。すなわち 12345678901234567890 を入力すると

12345678901234500000 となる [5]。RSA 暗号では，暗号化の際に m^e といった大きな数を取り扱う関係上，IEEE 754 仕様のままだと計算ができない。しかし，m^e が直後に mod N されることを考えると，(a mod N)・(b mod N) = ab mod N の性質を利用すると，

$$c \equiv (m^e) \pmod N$$
$$c \equiv ((m^{e-1} \bmod N) \times m) \pmod N$$
$$c \equiv (((m^{e-2} \bmod N) \times m) \pmod N) \times m) \pmod N$$
$$\cdots\cdots$$
$$c \equiv (\cdots\cdots (((m \bmod N) \times m) \pmod N) \times m) \pmod N) \cdots\cdots \times m) \pmod N$$

となり，指数を用いずに暗号文 c を求めることが可能となる。同様に

$$m \equiv (c^d) \pmod N$$
$$m \equiv ((c^{d-1} \bmod N) \times c) \pmod N$$
$$m \equiv (((c^{d-2} \bmod N) \times c) \pmod N) \times c) \pmod N$$
$$\cdots\cdots$$
$$m \equiv (\cdots\cdots (((c \bmod N) \times c) \pmod N) \times c) \pmod N) \cdots\cdots \times c) \pmod N$$

となり，指数を用いずに平文 m を求めることが可能となる。これを実装したのが，図 2 である。なお，受信者が秘密鍵 d を求める際は，セル G19 以下に表示されている拡張ユークリッドの互除法で求めている [6]。

	A	B	C	D	E	F	G	H	I	J	K	L
1		アリス			ボブ		公開されている情報					
2												
3	素数pを用意する	11	秘密鍵									
4	素数qを用意する	17	秘密鍵									
5	N = p × qを求める	187					N	187	公開鍵			
6	GCD((p-1)(q-1),e)=1 となるeを1つ求める	3					e	3	公開鍵			
7	MOD(ed, (p-1)(q-1))=1 となるdを求める	107	秘密鍵									
8				アリスに伝えたい数mを決める	19	平文						
9				MOD(m^e, N) （mをe回掛けたものを Nで割った余り） をアリスに送る	127	暗号文						
10	ボブからcを受け取る	127	暗号文									
11	MOD(c^d, N)を計算する	19	平文									
12												
13												
14												
15												
16												
17												
18												
19							拡張ユークリッドの互除法					
20	d	c^d		e	m^e		i	q	r	u	v	inv_v
21	1	127		1	19		1		160	1	0	160
22	2	47		2	174		2		3	0	1	161
23	3	172		3	127		3	53	1	1	-53	107
24	4	152		4	169		4	3	0	-3	160	320
25	5	43		5	32		5	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
120	100	67		100	67		100	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
121	101	94		101	151		101	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
122	102	157		102	64		102	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
123	103	117		103	94		103	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
124	104	86		104	103		104	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
125	105	76		105	87		105	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
126	106	115		106	157		106	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
127	107	19		107	178		107	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
128	108	169		108	16		108	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!

図 2：Excel 上で実装した RSA 暗号

例えば，図 2 では，素数 p を 11，素数 q を 17 とし，N = p × q = 187，e を 3 と決めて，公開鍵 (N, e) = (187, 3) を設定した。その後，拡張ユークリッドの互除法により，1 ≡ ed (mod (p-1) (q-1)) を満たす秘密鍵 d = 107 を求めた。そして平文として m = 19 を送る場合，送信者は $c \equiv (m^e) \pmod N$ を計算して，暗号文 c =127【セル E23 を参照】を得て，これを通信路を経由して送る。暗号文 c を受け取った受信者は $m \equiv (c^d) \pmod N$ を計算して，平文 m=19【セル B127 を参照】を得ることができる。このように，合同式の性質を用いることで，指数を実際には使用せずに計算することが可能となる。

今回は，Excel 上で動作するアプリケーション（図 2）と Web 上 (Scratch) で動作するアプリケーション（図 3）を授業教材として用いた。図 3 は，上記を Scratch で実装したもので，Web 上で体験できるようにしたものである（https://scratch.mit.edu/projects/487102676/）。

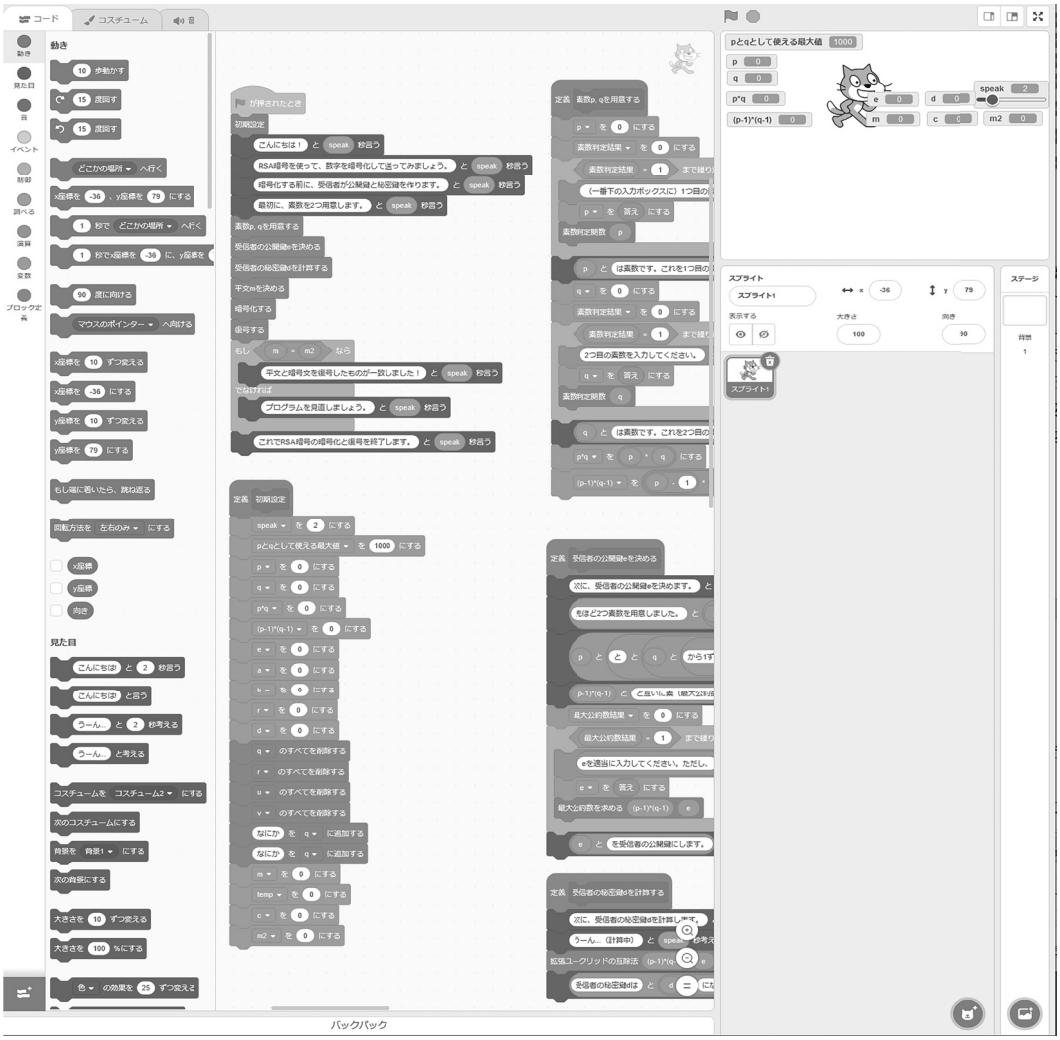

図 3. Scratch 上で実装した RSA 暗号

4. まとめ

本稿では，RSA 暗号の原理と，それを Excel や他のアプリケーションで実現するための方法について理論的な部分をもとに説明した。

これらのアプリケーションは，ただ開発するだけにとどまらず，中身を学習者自身も見ることができるため，暗号に興味を持ち，その背後にある数学的理論を理解し，将来，大学の理数系学部に進学して，暗号アルゴリズムの開発・解析など数理情報科学の分野で活躍できる人材を育成することにつながると考えられる。

今後の目標としては，様々な暗号化アルゴリズムについて，セキュリティの数理的部分を理解し，興味を持てるような教材の開発を目指す。将来的には，暗号化アルゴリズムの開発・解析など，情報セキュリティの分野で研究する人材を育成することを目指したい。

参考文献

[1] 神永正博，現代暗号入門，講談社，2017 年

[2] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” IEEE Transactions on Information Theory, IT-22（6）, pp64-654, 1976.

[3] R. L. Rivest, A. Shamir, L. M. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, MIT-LCSTM-082（1977）.

[4] Internet Engineering Task Force（IETF）, “ The Transport Layer Security（TLS）Protocol Version 1.3, “ RFC8446, https://tools.ietf.org/html/rfc8446, 2018［4］

[5] Microsoft Build, Excel のセルに長い数値を入力すると最後の桁がゼロに変更される , https://docs.microsoft.com/ja-jp/office/troubleshoot/excel/last-digits-changed-to-zeros, 2022 年 4 月 1 日閲覧

[6] IPUSIRON，暗号技術のすべて，翔泳社，2018 年

以上