学習指導案

北海道　　高等学校

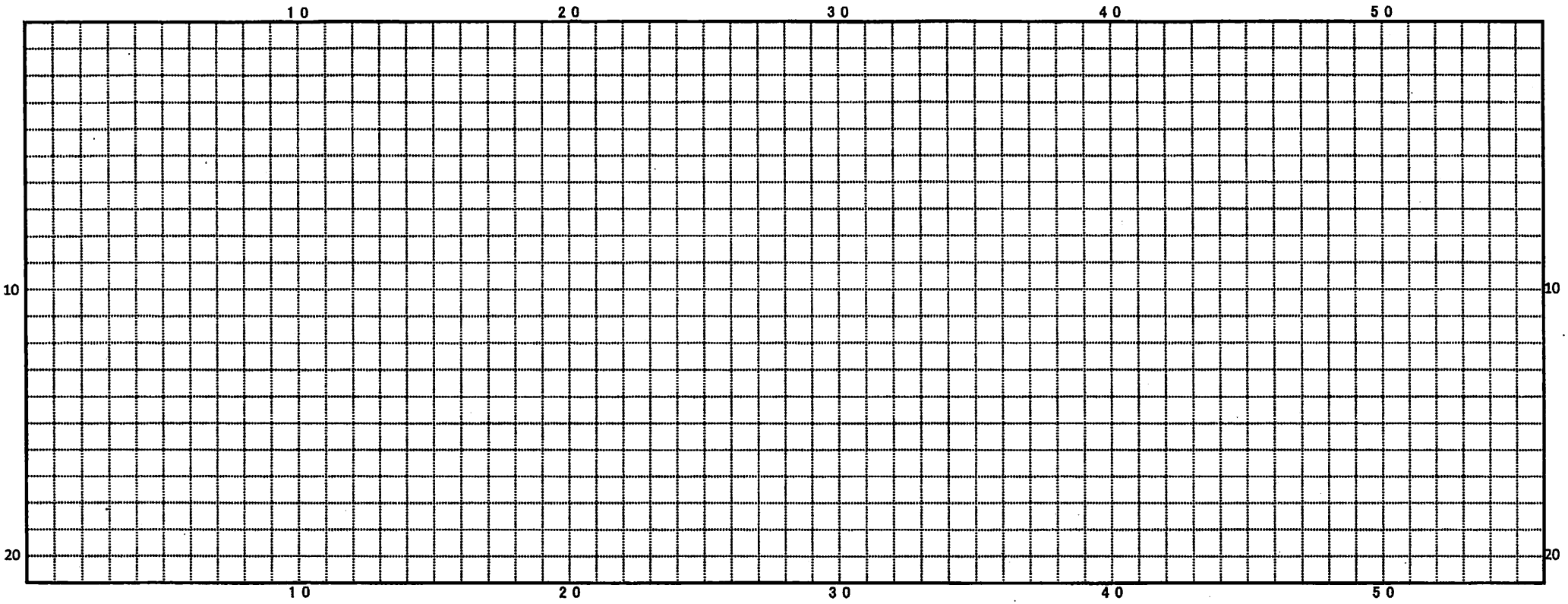
教科・科目	数学・数学A		
実施者	西村 昂介	日時	
対象学年・組	1年A組、1年B組	使用教科書	新編 数学A(数研出版)
実施教室	各HR教室	副 教 材	3TRIAL 数学Ⅰ＋A(数研出版)
単元名	第3章 整数の性質　第2節 ユークリッドの互除法		
単元の指導計画	2つの整数の最大公約数を求めるのに簡便な方法としてユークリッドの互除法を学ぶ。ユークリッドの互除法の応用として、基本的な1次不定方程式のすべての解を得る方法を学ぶ。		
本時の目標	「課題学習」として、既習事項を利用してユークリッドの互除法を、実習を通して理解を深める。また、教室で学んでいる内容が実際の生活でどのように生かされているのかを経験し、より理解を深める。		

段階	学習内容	生徒の活動	教師の活動と指導上の留意点	評価の観点・方法等
導入 5分	・最大公約数の求め方の確認 素因数分解	・班活動	・班を作らせる。 ・活動に集中させる。	【関】、【知】、【技】
展開① 10分	・最大公約数の求め方の確認 ユークリッドの互除法	・班活動	・最大公約数を見つけることの難しさを再確認させる。	
15分	・ユークリッドの互除法の理論をワークシートを切り取ることで理解を深める。	・はさみでワークシートの中でとりうる最大の正方形をとっていく。 ・他の班員と、最小の正方形の一辺について確認する。	・机間指導をし、様子を把握。	【関】、【見】
展開② 10分	インターネット社会における 素数の利用について	・RSA暗号やSSLについてプリントを用いて理解を深める。	・机間巡視をし、様子を把握。 ・レポート用紙を集める。	【関】、【知】、【見】
(計算・考察) 7分	・感想を書く。	・感想を書く。		
まとめ 3分	・発表			

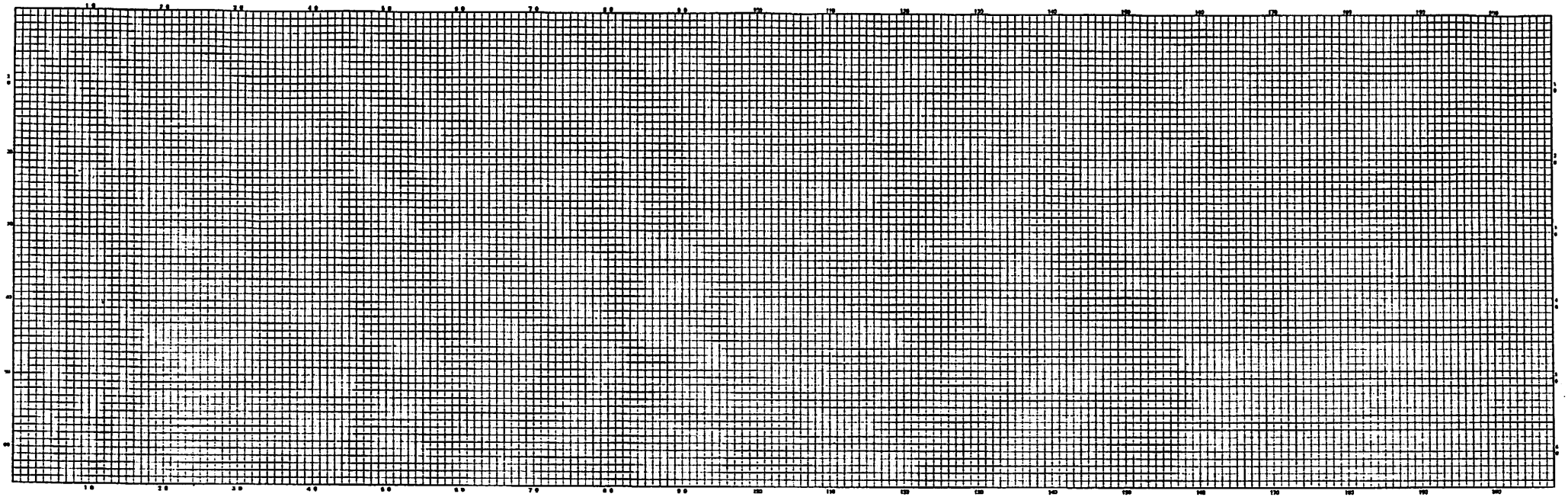
【関】:関心・意欲・態度、【見】:数学的な見方・考え方、【技】数学的な技能、【知】:知識・理解

ユークリッドの互除法の理解を深めるための

21×56　のワークシート



65×208 ワークシート

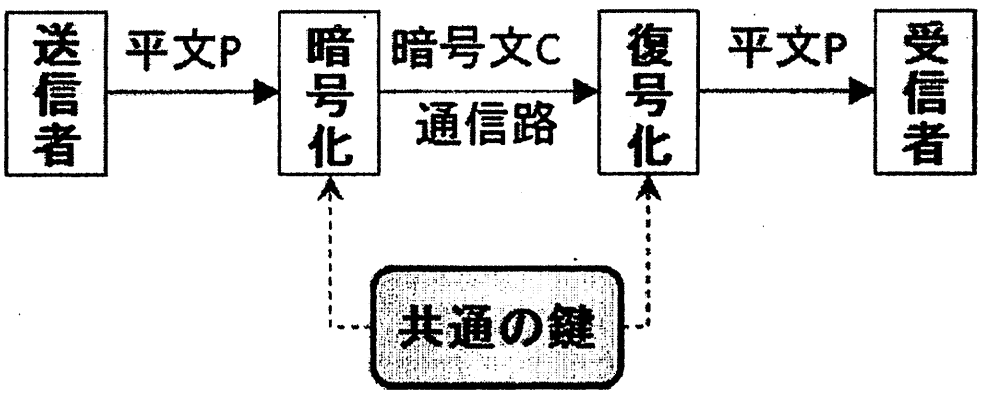


暗号入門

シャノン(アメリカ、1916 年)は、通信路における符号理論だけでなく、暗号理論においても研究を行いました。今日のインターネット社会において、暗号は情報の安全性を確保するために重要な役割を果たしていること皆さんも知っている通りです。そこで今回は、暗号の基礎を学びましょう。

シャノンの暗号化モデル

シャノンは、下図のような暗号化のモデルを示しました。

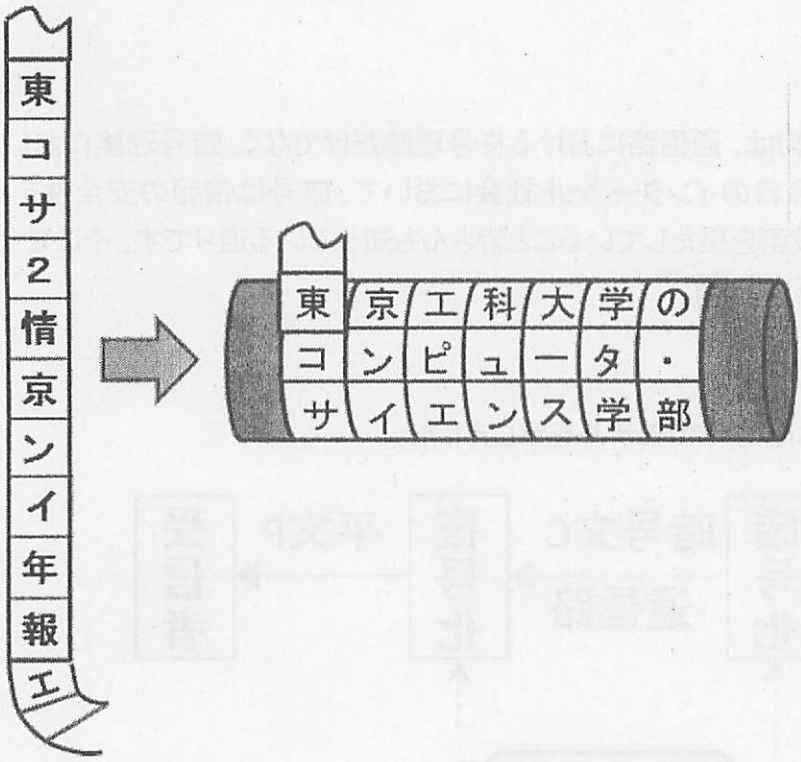


上図において、送信者は平文 P(Plain text)を鍵を使って暗号化し、暗号文 C(Cipher text)を通信路を介して送信します。受信者は、暗号文を元に戻す鍵を使って復号化し、平文を受け取ります。

ここで、問題となるのは、鍵をどのように作って、どのように受信者に渡すかということになります。
歴史的によく知られた、簡単な暗号を次に示します。

簡単な暗号の例

a)スキュタレー暗号
最古の暗号、スキュタレー:ギリシャ語で棒を意味する
鍵:一定の文字間隔で取り出す(歴史では棒に巻きつけて解読)
【例】「東コサ2惰京ンイ年報工ピエ生・科ュンの符大一ス講号学タ学義理の・部の論」を縦書きでテープに書き出し、5文字分の太さの棒に巻きつける



b)シーザー暗号
鍵:X 文字分ずらす
【例】INFOMRATION → GLDMKPYRGML (-2 文字)
ジュリアスシーザーは 3 文字ずらしたと言われる。映画「2001 年宇宙の旅」に「HAL9000」というコンピュータが登場するが、このコンピュータは IBM だという話は有名。
(HAL を 1 文字ずつ後ろにずらす)

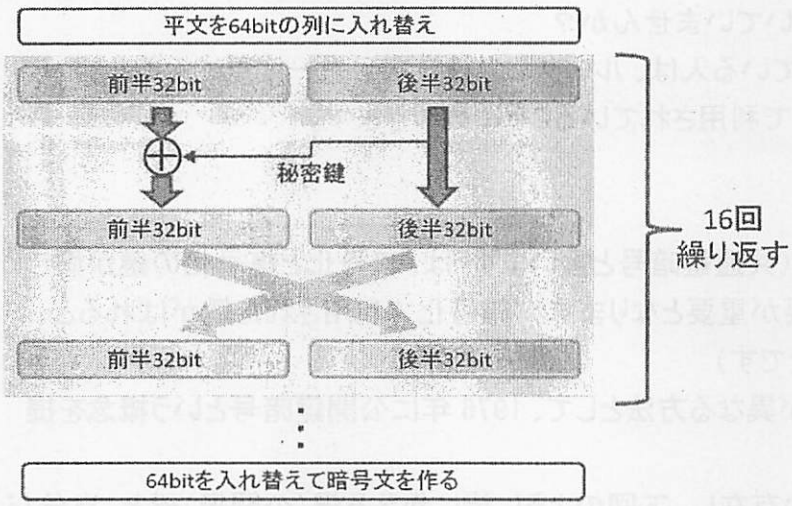
c)転置暗号
鍵:文字の順番を入れ替える
【例】TOKYO UNIVERSITY → YKTOO VIUENTIRYS
(1→3、2→5、3→2、4→1、5→4) 5 文字ブロックで入れ替え

a)や b)のような暗号は単純だが、c)のような暗号はやや複雑となっています。c)のようなn文字毎のブロックで暗号化を行う方法をブロック暗号といい、今日ではもっと複雑な方法ではあるが利用されています。

ブロック暗号
ブロック暗号は、一般的に、平文を一定のビット長のブロックに区切り、ブロックごとに

暗号化を行う方法です。

a)DES(Data Encryption Standard)
1977 年に米国標準暗号として採用された暗号ですが、2005 年に米国標準暗号から取り下げられました。
DES は、平文を 64 ビット毎に分割した後、ビットシフトと XOR(エクスクルーシブオア、コンピュータ概論で習いました)を行い、秘密鍵で取り出す操作を 16 回繰り返すというものです。



複雑に感じますが、その後、解読方法(線形解読法という)が発見され、標準暗号から取り下げられました。
現在でも、この DES を 3 回繰り返し、鍵を 3 倍にしたトリプル DES と呼ばれる暗号化は利用されています。

<u>身近な話</u>
UNIX では、ユーザパスワードの暗号化に古くから DES が採用されてきました。現在の Linux では、MD5(Message Digest 5)と呼ばれるハッシュ関数でパスワードを変換しています。
DES では、8 文字のパスワード制限(64bit ブロックで暗号化するので)があり、現在の Linux ではセキュリティ的に問題となることから、MD5 が標準となっています。MD5 は、パスワードの文字数に特に制限はありませんが、Linux では 256 文字までのパスワードに対応しています。
もし、皆さんが、Linux で MD5 を使っているのに 8 文字程度のパスワードを設定している場合は、セキュリティ的に DES と同じ程度のセキュリティ強度となってしまいます。したがって、8 文字以上のパスワードに設定することが望ましいことになります。

b)AES(Advanced Encryption Standard)

AES は、DES に代わる米国標準暗号として 2000 年に採用された、ブロック暗号です。AES は、平文を 128 ビット毎に分割した後、行列演算と秘密鍵操作により暗号化の処理を行う手法で、DES の解読方法に対応できるように設計されています。現在でも、ブロック暗号の標準として各国で利用されています。

<u>身近な話</u>

皆さんの中で、家で無線ネットワーク環境を利用している場合、使っている無線ルーターの暗号化に AES を用いていませんか？

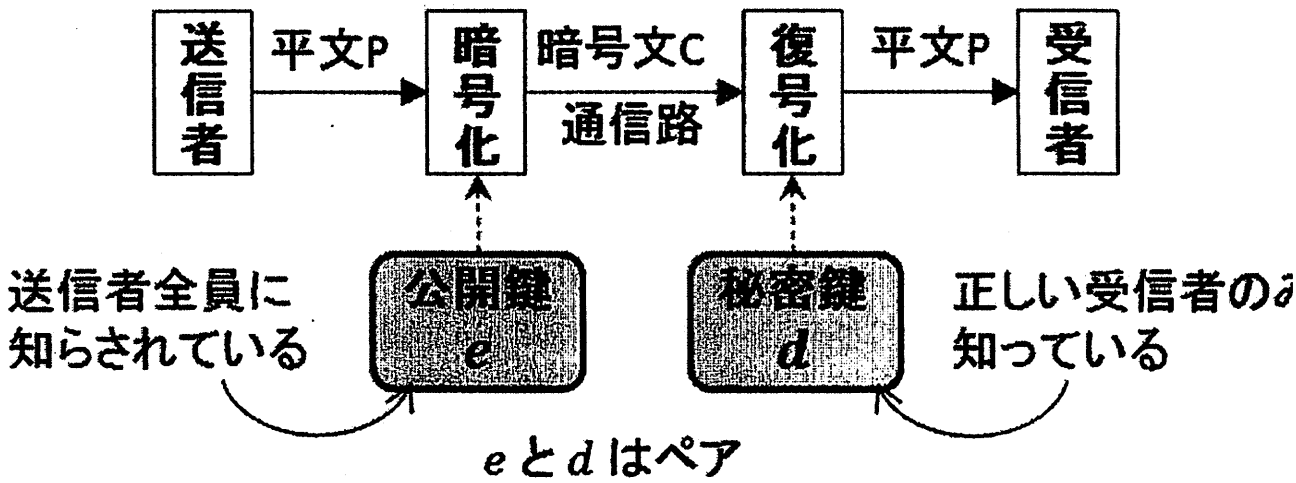
無線ネットワークを利用している人は、ルーターの設定項目を一度見てみるとよいでしょう。意外と身近なところで利用されていることに気づきます。

公開鍵暗号

ここまで紹介してきた暗号（共通鍵暗号と言います）は、暗号化と復号化の鍵が同じため、鍵を秘密にする必要が重要となります。（暗号化で利用された鍵がばれると、解読できてしまうということです）

共通鍵暗号とは、考え方が異なる方法として、1976 年に公開鍵暗号という概念を提案されました。

公開鍵暗号では、鍵が2つ存在し、下図のように皆に教える鍵（公開鍵）鍵と、自分だけが知っている秘密の鍵（秘密鍵）を利用する方法です。



この概念はディフィとヘルマンという2人の研究者によって提案されましたが、鍵の実現方法までは確立されませんでした。その後、次の RSA 暗号によって、鍵の生成方

法が提案されました。

a)RSA

1978 年、リベスト(Rivest)、シャミア(Shamir)、アドルマン(Adleman)の3人によって、公開鍵暗号の具体的な設計方法が提案されました。3人の頭文字から、RSA 暗号と呼ばれ、現在ではこの仕組みが幅広く活用されています。

RSA 暗号では、鍵生成の仕組みに素数の積を利用し、解読の難しさを素因数分解の困難さに帰着させています。

【鍵生成方法】

①異なる2つの素数(p,q)を準備 n＝p×q(現在、100 ケタ程度は危険とされる)

②(p－1)と(q－1)の最小公倍数 k を求める

③k と素な数 e を決める

④e×d＝1 (mod k)を満たす d を決める

⑤(e, n)を公開鍵

(d, n)を秘密鍵とする

【暗号化】

公開鍵(e, n)を用いて、

送りたい平文(ASCII コードなど) P に対して、C = P^e mod n を求め、C を暗号文とする

【復号化】

秘密鍵(d, n)を用いて、

送られてきた暗号文 C に対して、C^d mod n を求めると平文 P に復号できる。

鍵の生成

実際の原理を、短い鍵と簡単な数値で具体的に示してみます。

① p＝11、q＝13、n＝143 (本当は 100 ケタ以上の素数を選びます)

② p-1＝10、q-1＝12 より、最小公倍数 k＝60

③ k=60 と素な数として e＝7 を選択(素な数は任意)

④ d＝43 とする(e×d＝7×43＝301＝1 mod 60 を満たす d を決める)

暗号化:(e=7, n=143)を用いる

ここでは、平文 P=5 を暗号化してみる

$C = P^e \bmod n = 5^7 \bmod 143 = 78125 \bmod 143 = 47$
より、暗号文は C=47 となる。

復号化:(d=43, n=143)を用いる

$P = C^d \bmod n = 47^{43} \bmod 143$

$=(47^2)^{20} \times 47^3 \bmod 143$　　$(47^2 \bmod 143 = 64)$

$=(64^2)^{10} \times 47^3 \bmod 143$　　$(64^2 \bmod 143 = 92)$

$=(92^2)^5 \times 47^3 \bmod 143$　　$(92^2 \bmod 143 = 27)$

$=27^5 \times 47^3 \bmod 143$　　$(27^5 \bmod 143 = 1)$

$=1 \times 47^3 \bmod 143$

$=5$

よって、復号化ができた。

上記より、RSA 暗号の解読はnから、素数pとqを割り出すことができれば、解読されてしまうことになります。上記の例ではn＝143 だから、143 を割り切る素数が見つかれば解読できることになります。素数を小さい方から並べると{3,5,7,11,13,17,19,・・・・・}であるので、143 を小さい素数から割り切れるかどうか確かめていけば良いことになり、この例だと、すぐに 11 で割り切れることが見つかってしまいます。

しかし、nが 50 ケタだったらどうでしょうか？
原始的に総当たりで確認していくとすると、25 ケタまでの素数で割り切れるかどうか確認すればよいことになります。X までの整数の中にある素数は、おおよそ $X \div \ln(X)$ で近似されるので、25 ケタまでに存在する素数は、約 10^{23} 個になります。
ここから、割り切れる素数を見つけるためには、約10^{23}回の割り算が必要になります。
現在の私たちが利用している、コンピュータの CPU は 1GHz(1 秒間に 10^9)ですから、1 クロックで 1 回割り算ができる(割り算には通常数十クロックかかります)と仮定しても、$10^{23} \div (10^9 \times 60 \times 60 \times 24 \times 365)$＝約 317 万年となります。

現在、効率的に素因数分解をする有力な計算法として、数体ふるい法という方法があります。難しい方法なので、厳密な原理ではないですが、nの素因数分解は、$\sqrt{n}$ の近くの値から少しずらした値の組み合わせを作り、その合成から候補を見つけ出すことが可能だというものです。この方法では、nの素因数分解の計算量は
$\exp((64/9)^{1/3}(\ln n)^{1/3}(\ln \ln n)^{2/3})$
となります。

n＝150(150 ケタ)として、計算してみると、計算量は約 10^{19} となり、先ほどと同様に 1GHz の CPU で概算してみると、約 317 年で解読できてしまいます。(十分長いと感じますが、RSA が提案された当初に比べ驚異的な短縮です)

2010 年に、NTT は 300 台の PC を利用して、232 ケタの素因数分解を 3 年で解くことに成功しています。このことから、100 ケタ程度の素数(n＝200 となる)では、私たちの使っているノート PC で解読されてしまう時代となってきています。

b)エルガマル暗号

RSA 暗号は、素数の積(n＝pq)から p と q の組み合わせを求めるのに膨大な時間がかかるという性質を使った暗号化です。これに対して、aを何乗かした値がzとなるとき(a^d＝z (mod p))、何乗したかを求める(d を求める)ことは非常に難しいという性質(離散対数問題といいます)を使った暗号化があります。エルガマル暗号は、この離散対数問題を利用した暗号で、使い捨ての鍵(乱数)を用意し、同じ平文でも毎回異なる鍵を使うことで暗号文が異なるため、強秘匿性をもっています。

離散対数問題

$y=g^x \bmod p$

において、p と g が与えられているとき、x から y を求めることは簡単であるが、y から x を求めるのは難しく、これが離散対数問題といわれる。

もし、mod p という操作がなければ、数学的に x=y÷log(g)で求めることができると思うかもしれないが、mod p の操作が加わったとたんに、この計算では答が求められない問題となる。

簡単な例ではあるが p=7、g=3 が与えられているとき、x=4 なら

$y=3^4 \bmod 7 = 81 \bmod 7 = 4$

として、答はすぐに求まるが、

p=7、g=3 が与えられていて、y=1 となる x は何か？

という問題だと、オーソドックスには、

$3^1 \bmod 7 = 3$

$3^2 \bmod 7 = 2$

$3^3 \bmod 7 = 6$

.

.

と求めていくことになる。

しかし、p の値が 50 桁の数値ともなると、10^{50} 回求めては、答を確かめる必要があり、答を求めるまでに膨大な時間がかかる問題となる。
(実際には、もう少し効率的なアルゴリズムもありますが、指数オーダーの計算が必要になることは変わらず、膨大な時間がかかることになります。)

c)楕円曲線暗号
エルガマル暗号と同様に、鍵の工夫を行った暗号化で、楕円曲線という関数($y^2=x^3+ax+b$)上の2つの点の積から2つの点の値を求めることが非常に難しいことを利用した暗号化で、エルガマル暗号より更に秘匿性を高めた手法です。

SSL のやりとりは複雑だ。事前知識なしにすべてをいきなり理解しようとすると、太刀打ちできない。そこで、SSL の実際のやりとりを見る前に、Lesson2 として SSL のやりとりの概要をざっとつかもう。

暗号通信で使う「鍵」を作る

SSL は、「共通鍵暗号」と「公開鍵暗号」という二つの暗号方式を組み合わせて利用している。実は、SSL の概要をつかむにあたって、これらの暗号方式の知識が欠かせない。最初に押えておこう(図 2-1)。

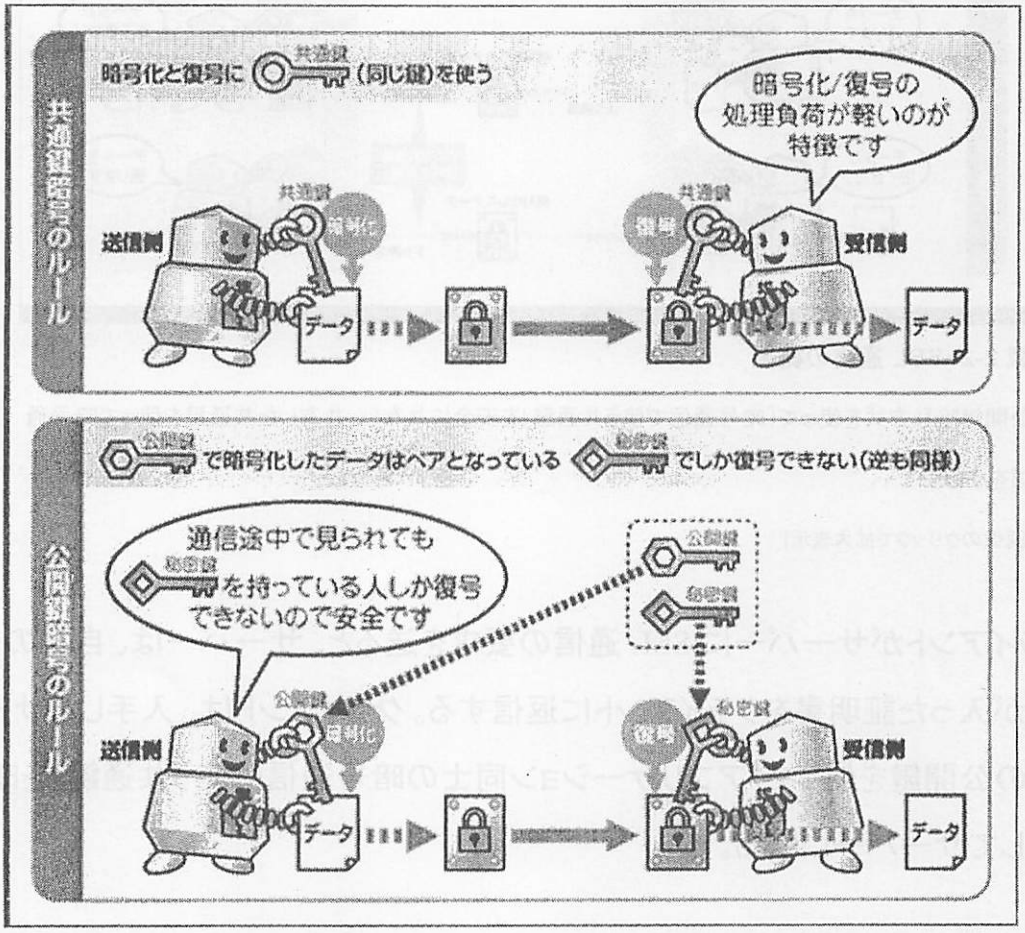


図 2-1●SSL では二つの暗号方式を使う

SSL は、共通鍵暗号方式と公開鍵暗号方式のそれぞれのメリットを組み合わせて実現する。

[画像のクリックで拡大表示]

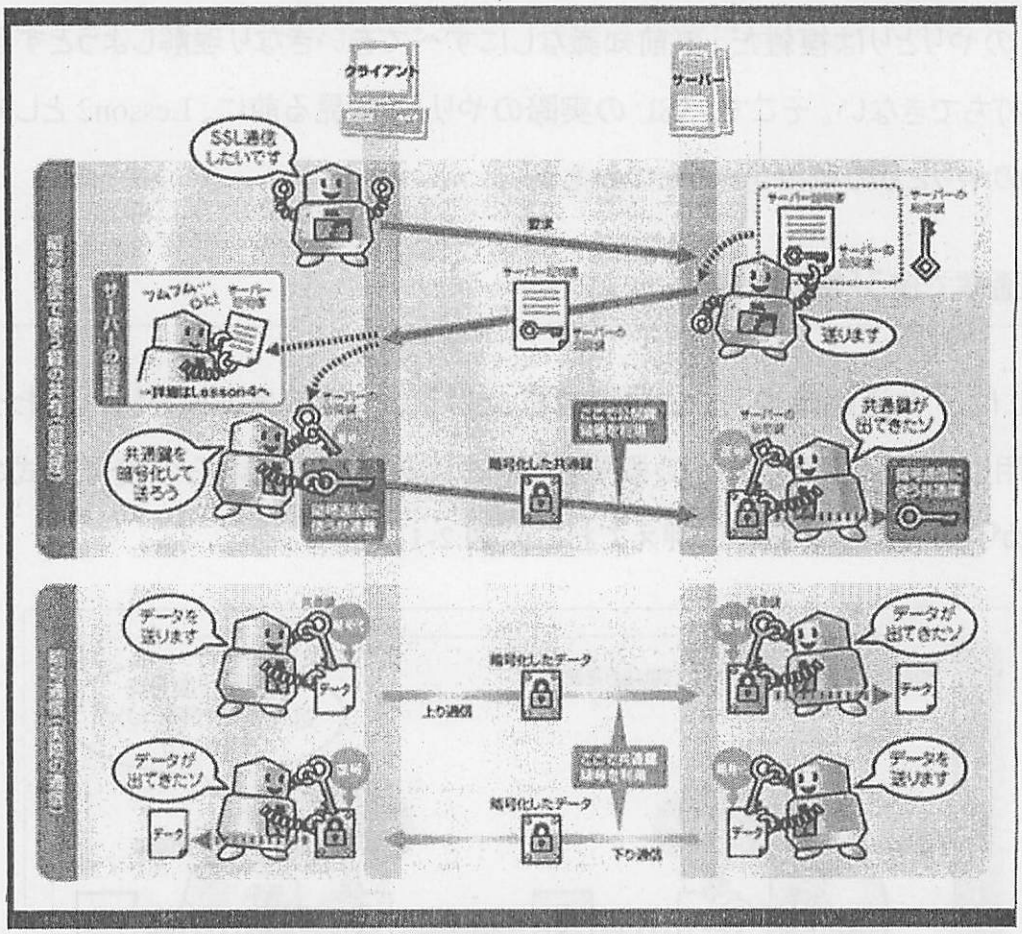


図 2-2●SSL 通信の概要

公開鍵暗号方式を使って「暗号通信で使う共通鍵」を安全に共有し、共有した共通鍵を使って暗号通信をする。

[画像のクリックで拡大表示]

クライアントがサーバーに SSL 通信の要求を送ると、サーバーは、自身の公開鍵が入った証明書をクライアントに返信する。クライアントは、入手したサーバーの公開鍵を使って「アプリケーション同士の暗号通信に使う共通鍵」を暗号化してサーバーに送る。

この暗号化した共通鍵を正しく復号できるのは、暗号化に使った公開鍵のペアとなっている秘密鍵を持つ通信相手のサーバーだけである。このため、ここまでのやりとりが万一盗聴されたとしても、共通鍵が漏えいする心配はない。