それから，秘密用の地図を渡して，子どもたちが正しく解読できることを確かめます．

ここまでくれば，各ペアが自分たち自身で地図を作って，秘密用の地図を隠しておいて公開用の地図を（グループ内の）もう一方のペアに手渡すことができます．あるいは，教室の掲示板に貼り出して「公開」してもかまいません．[*6]

地図を作る原理はツーリストタウンの学習で説明した方法と同じで，解を隠すために余分な道を付け加えてもかまいません．

ただ，「特別な」点につながる道を余分に付け加えないように気をつけてください．

それをしてしまうと，ある交差点からは 2 台のアイスクリーム販売車に一跳びでたどり着けるようになります．そうなると，ツーリストタウンの状況では問題ないのですが，暗号化をするときには混乱を招きます．

なぜかというと，特別な点を使っても，もはや地図は図 18.3 のように重なりのない断片には分割されなくなってしまうからです．重なりのない断片に分割されることは，この仕掛けがうまくいくためにどうしても必要です．

## What's it all about?
## これって，どういうこと？

コンピュータネットワークを通じて，意図された受信者以外の誰も，どんなに賢い人でも，あるいはどんなに努力しても読み取れないように，秘密のメッセージを送るということが，社会において必要である理由は明らかです．

もちろん，送信者と受信者が秘密の暗号表を共有していれば，それを実現する方法はいろいろあります．

でも，公開鍵暗号系のよくできたところは，エイミーはビルに，何ら事前の秘密の打ち合わせをすることなく，単にウェブページのような公開された場所から彼の錠を取ってくるだけで，安全にメッセージを送ることができるという点です．

秘匿は暗号理論の一方の面でしかありません．

もうひとつの面は認証です．

エイミーがビルからのメッセージを受け取ったとき，エイミーはどうすれば，そのメッセージがほんとうにビルが送ったもので，ビルの名をかたる何者かが送ったものではないことを知ることができるでしょうか？

彼女が次の内容の電子メールを受け取ったとしましょう．

「恋人よ，おれは一文無しで路頭に迷っちまった．おれの銀行口座に 100 ドル振り込んでおくれ．口座番号は 0241-45-784329 だ．よろしく．ビル」

彼女はこれがほんとうにビルからのメールなのか，どうすればわかるでしょう？

一部の公開鍵暗号系は，このような目的にも使うことができるのです．

エイミーがビルに秘密のメッセージを送るときにビルの公開鍵を使って暗号化したのと同じ要領で，ビルは，自分の秘密鍵で暗号化することで，彼にしか作りえないメッセージをエイミーに送ることができます．

エイミーがそのメッセージをビルの公開鍵で解読できれば，それは彼が送ったものであるはずです．

もちろん，ほかの誰でも，そのメッセージを解読することができます．なにしろ鍵は公開されているのですから．でも，そのメッセージをエイミーだけに読んでほしければ，ビルはそのメッセージをさらにエイミーの公開鍵で暗号化すればよいのです．

このような二重の暗号化によって，同一の公開鍵と秘密鍵の基本的な仕組みで，秘匿と認証の両方を実現することができます．[*7]

---

[*6] classroom board : 掲示板に貼り出すのか，それとも「黒板に描く」のか？

[*7] dual encoding とは？「ビルの秘密鍵で暗号化してからエイミーの公開鍵で暗号化」という，2 段階の暗号化の手順を踏むことなのか？（上述の訳文はこの解釈）それとも，RSA のように「暗号化と復号化の手順が双対的であること」を意味するのか？（同一の方式を秘匿と認証の両方に使うためにはその性質が必要）

さて，ここで白状します．この学習で紹介したしくみは，産業界で利用される強力な公開鍵暗号化方式とよく似ているのですが，実は，安全な方式ではないのです．たとえ非常に大きな地図を使ったとしてもです．

勝手な地図を見て台数が最少となるアイスクリーム販売車の配置のしかたを見つける方法は知られていないので，この観点からは，この学習で使った方式はたしかに安全です．しかし，それとは全く異なる攻撃の方法があるのです．

生徒が（少なくとも，高校生レベル以下であれば）この着想を得ることはまずありませんが，あなたは少なくともそのような攻撃のしかたがあることを知っておくべきです．

言ってみれば，私たちが見てきたこの仕組みは，生徒が相手なら安全ですが，数学者が相手なら安全ではないのです．

数学に興味がない人は，次のパラグラフは読み飛ばしてください！

地図上の交差点に 1, 2, 3, ... と番号をつけます．

交差点に割り当てられたもとの数を b1, b2, b3, ... で表し，実際に送信された数を t1, t2, t3, ... で表します．

交差点 1 が交差点 2, 3, 4 とつながっているとします．

すると，この交差点について，送信された数は t1 = b1 + b2 + b3 + b4 です．

もちろん，同様の方程式がほかのすべての交差点について作れます．実は，作られる方程式の数は未知数 b1, b2, b3, ... の数と同じです．

盗聴者は公開用の地図と送信された数 t1, t2, t3, ... を知っているので，方程式を書き下して，それらを方程式求解コンピュータプログラムを使って解くことができます．

もとの数が得られたら，メッセージはそれらの合計です．つまり，解読用の地図を発見する必要すらなかったのです．

ガウスの消去法を直接的に使って方程式を解くために必要な計算の手間は，方程式の数の 3 乗に比例します．しかし，これらの方程式は，「まばらな」（大半の係数がゼロの）方程式なので，もっと効率的に解く技法が存在します．[*8]

この計算の手間を，指数的な計算の手間と見比べてください．われわれの知る限り，解読用の地図を見破るために必要な最善の計算の手間は指数的な計算の手間なのです．

「だまされた」と思わないでください！

実は，現実の公開鍵暗号系で使われている手順は私たちが見てきたものとほとんど同じで，暗号化に使っている計算手法だけが違うのです．そして，現実の公開鍵暗号系の計算手法は，手作業で計算するのは現実的ではありません．

最初の公開鍵暗号手法は，現時点で最も安全なもののひとつで，大きな数の素因数分解が困難であることに基づいています．

100 桁の数 9,412,343,607,359,262,946,971,172,136,294, 514,357,528,981,378,983,082,541,347,532,211,942,640,121,301,590,698,634,089,611,468,911,681 の素因数は何でしょう？

あまり長い時間考えてはいけません！

素因数は 86,759,222,313,428,390,812,218,077,095,850,708,048,977 と 108,488,104,853,637,470,612,961,399,842,972,948,4611,525,790,577,216,753 です．

ほかの素因数はありません．これら 2 つの数は素数です．

これらの素因数を見つけるのはたいへんな仕事です．実際，スーパーコンピュータを使っても数ヶ月かかる大仕事です．

実際の公開鍵暗号系では，ビルは 100 桁の数を公開鍵として，そして 2 つの素因数を秘密鍵として使います．

鍵を作り出す方法は，そんなに難しくありません．大きい素数を生成する方法さえあれば十分です．

---

[*8] （保福先生による指摘）実は，例示されている図 18.1 の地図の場合，もっと簡単に破れる．すべての点の次数が 3 であるという特殊性があるので，（送信された数の総和）/4 でメッセージが求められる．この攻撃法は生徒が発見できる可能性がある．

十分に大きい 2 つの素数を見つけて（これはそれほど困難ではありません），それらを掛け合わせれば，ほら，このとおり，公開鍵のできあがり．

巨大な数の掛け算は，コンピュータを使えばわけもない仕事です．

公開鍵がわかっているとしても，スーパーコンピュータを数ヶ月動かし続けない限り，秘密鍵を見破ることはできません．

もし，秘密鍵を見破られてしまうことが心配なら，100 桁でなく 200 桁の素数を使ってください．そうすると，見破るのに何年もかかるようになります！

実際には，512 ビットの鍵が使われています．10 進の桁数で表すとおよそ 155 桁です．[*9]

素数に基づく公開鍵を使ってメッセージを暗号化して，秘密鍵なしには解読できないようにするための方法は，まだ説明していません．

それをするのは，ここまででしてきたことほど単純ではありません．

実は，2 個の素数を秘密鍵として，それらの積を公開鍵としてそのまま使うのでなく，そのかわり，それらから導き出される数を使います．

それでも効果は同じです．数を素因数分解することで暗号を破ることができます．

ともかく，これらの難しさを乗り越えて，実際の暗号化および復号化アルゴリズムの仕組みを作ることは，難しくはありません．でも，ここでは深入りしません．

この学習で行った作業で，十分でしょう！

素数に基づく暗号系はどれほど安全でしょうか？

大きい数の素因数分解は，数百年にわたって，世界中の偉大な数学者の注目を集めている大問題です．そして，可能性のあるすべての素数を試していく総当たり法よりは目に見えて効率の良い手法は発見されていますが，それでも，真に速い（つまり，多項式時間の）アルゴリズムは発見されていません．

（一方，そのようなアルゴリズムが存在しえないということの証明も，なされていません．）

だから，この仕組みは，学校の生徒を相手にして安全だというだけでなく，数学者を相手にしても安全であると考えられます．

でも，注意が必要です．

ツーリストタウン問題を解くことなしにビルの暗号を破る方法があったように，もしかしたら，大きい数の素因数分解をすることなく，素因数分解に基づく暗号を破る方法が見つかるかもしれません．

このことはすでに注意深く検証されていて，どうやら大丈夫のようです．

別の心配は，メッセージが少数しかありえない場合，侵入者がありうるメッセージを順々に公開鍵で暗号化して，実際に送られるメッセージをすべての可能性と比較することです．

エイミーのやり方はこの心配を回避しています．同一のメッセージを暗号化するやり方は，合計が目的の値になるような数の選び方に応じて，たくさんあるからです．

実際には，非常に速いコンピュータが使えるとしても，すべてのメッセージを試す気にはならないぐらいに，あり得るメッセージの数が膨大になるように，暗号系を設計します．

素因数分解問題を解く高速な手法が存在するかどうかは知られていません．

だれひとりとして，それを発明できていません．一方，そのような高速な手法が存在しえないことの証明もなされていません．

この問題を解く高速なアルゴリズムが発見されたら，現在使われている多くの暗号方式が安全でなくなってしまいます．

第 IV 部では NP 完全問題を論じました．NP 完全問題はすべて生死を共にしています．どれかひとつが効率的に解けるなら，すべてがそうなのです．[*10]

---

[*9] これは古い話．今では 1024 ビット（309 桁）が普通で，安全性の観点から 2048 ビット（617 桁）への移行が推奨されている．（参考）ITpro：NTT など、公開鍵暗号の素因数分解問題で 768 ビット整数の分解に成功

[*10] 「第 IV 部」が何をさすのか不明ですが，たぶん 13–15 章．

これらの問題を速く解くアルゴリズムを探すために多大な努力がなされてきた（しかし成功していない）ことから，これらの問題は安全な暗号系を設計するために使える優良な候補であるように思えます．

残念なことに，この方針には困難があります．それで，今のところ，暗号系の設計者は（たとえば素因数分解のような）実は NP 完全問題と比べれば解くのが容易かもしれない（たぶん大幅に易しい）問題に頼らざるを得ないのです．[*11]

ここで述べたような課題に答えることは，産業界にとっては数百万ドルの価値があり，また，国防にとってはきわめて重要な問題だと考えられています．

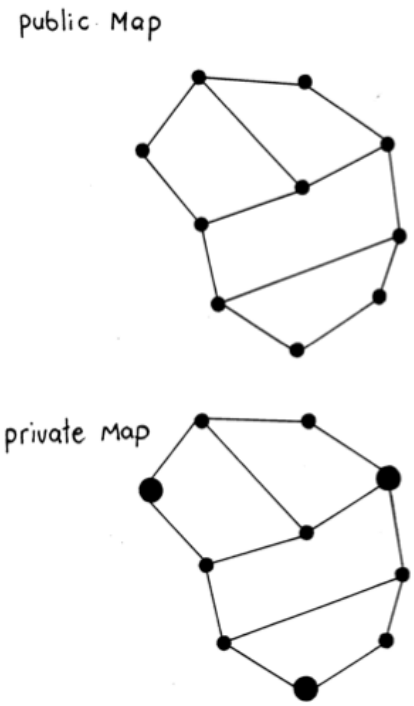
暗号理論はコンピュータサイエンスで現在非常に活発に研究されている分野です．

## Further reading
## 参考文献

Harel の本「Algorithmics」では，公開鍵暗号系を論じています．本では，大きい素数を使って安全な公開鍵暗号系を作り出す方法を説明しています．

暗号理論についての標準的なコンピュータサイエンスの教科書は Dorothy Denning の「Cryptography and data security」です．でも，より実用的な本は Bruce Schneier の「Applied cryptography」です．

Dewdney の「Turing Omnibus」では，公開鍵暗号を実現する別の方式を説明しています．




指示：この地図を使って，本文で説明した要領で，メッセージを暗号化，復元します．

[*11] なぜ困難があるのかは明示されていませんが，NP 完全問題を使った「落とし戸つき一方向関数」が見つからないことが理由と考えられます．公開鍵暗号に使えそうな落とし戸つき一方向関数がうまく見つかるのは，えてして，NP よりずっと小さい（易しい）計算量クラスというのが（現時点での）現実のようです．