公開鍵方式 (RSA 方式) のモデル

全単射 (bijection) を学びました．すでにトーナメント方式の試合数を求める問題に応用しました．ここではさらにインタネットセキュリティにおける公開鍵方式のモデルの説明に用いてみましょう．関数 $f\colon A \longrightarrow B$ が全単射ならば f は逆関数 $f^{-1}\colon B \longrightarrow A$ をもち等式 $f^{-1}f = \mathrm{id}_A$, $ff^{-1} = \mathrm{id}_B$ がともに成り立ちました．ここで $\mathrm{id}_A, \mathrm{id}_B$ はそれぞれ集合 A, B 上の恒等関数でした. この簡単な性質を用いて，公開鍵暗号方式のすばらしいアイデアを説明しようというものです．なお公開鍵方式の数学的詳細は，ここでは目的範囲外ですので，ほとんどふれません．

通信のセキュリティ

真二はインターネットで桃子へメッセージ M を送りたい．他人には読まれたくない二人だけの秘密のメッセージです．しかし，M をそのまま送ると，ネットワークの途中で盗聴（盗読？）される危険性がある．では，どうしたらよいでしょう？

まず，次のような送り方が考えられるでしょう．テキスト M をテキスト M' へと暗号化し，そのままでは他のだれにも読めないようにする．そうすれば，たとえテキスト M' が途中で盗まれても，内容を読まれる心配はありません．桃子だけが，M' を元のテキスト M に復号化できればいいわけです．そのためには，復号化のための鍵を別途桃子に送ればよい．

しかし，ここに問題点があります．まず，この復号化のための鍵が，途中で盗まれたりコピーされたりする危険性です．2 番目の問題は，相手によって異なる復号化鍵を用意しなければならないことです．真二と桃子，真二と花子，真二と圭子，$\cdots$， 桃子と太郎，桃子と次郎，$\cdots$, 各対ごとに異なる復号化鍵が必要になります．

公開鍵方式

このふたつの問題点を解決したのが，公開鍵方式です．公開鍵方式では，復号化のための秘密の鍵を相手に送る必要はありません．しかも真二から桃子に送られた暗号文は，桃子だけにしか元にもどせません

公開鍵方式のアイデアを説明しましょう．各ユーザは二つの鍵を持ちます．ひとつは，公開鍵，他方は秘密鍵と呼びます．公開鍵はすべてのユーザに公開します．誰もが真二の公開鍵を知ることができ，誰もが桃子の公開鍵を知ることができます． 一方，秘密鍵は，誰にも教えてはいけない自分だけが知ってる鍵です．真二の秘密鍵は真二しか知りません．桃子の秘密鍵は桃子しか知りません．

ところで，公開鍵といい秘密鍵といい，そもそも鍵とは具体的にはなんでしょうか？ここではそれに立ち入らず，鍵とはテキストを変換する機能を持つ或るものと理解します．つまり，鍵とはテキスト変換プログラムです．そして異なる鍵は，関数として異なる変換機能を持つとします．

そうすると，鍵 K が表すテキスト変換機能を鍵 K そのものと区別して，前者を $\widehat{K}$ と書きます．つまり，テキストを入力してテキストを出力する関数が K に依存して決まると考え，それを $\widehat{K}$ と書きます．そして，テキスト M をブラックボックス $\widehat{K}$ に入力したときの出力テキストを $\widehat{K}M$ と書きます．

公開鍵方式では，各ユーザは，自分の秘密鍵 S と公開鍵 P を，次の条件を満たすように自由に一組選びます．

M を任意のテキストとして，次の 1, 2 どちらの変換を適用しても M は変わらない．

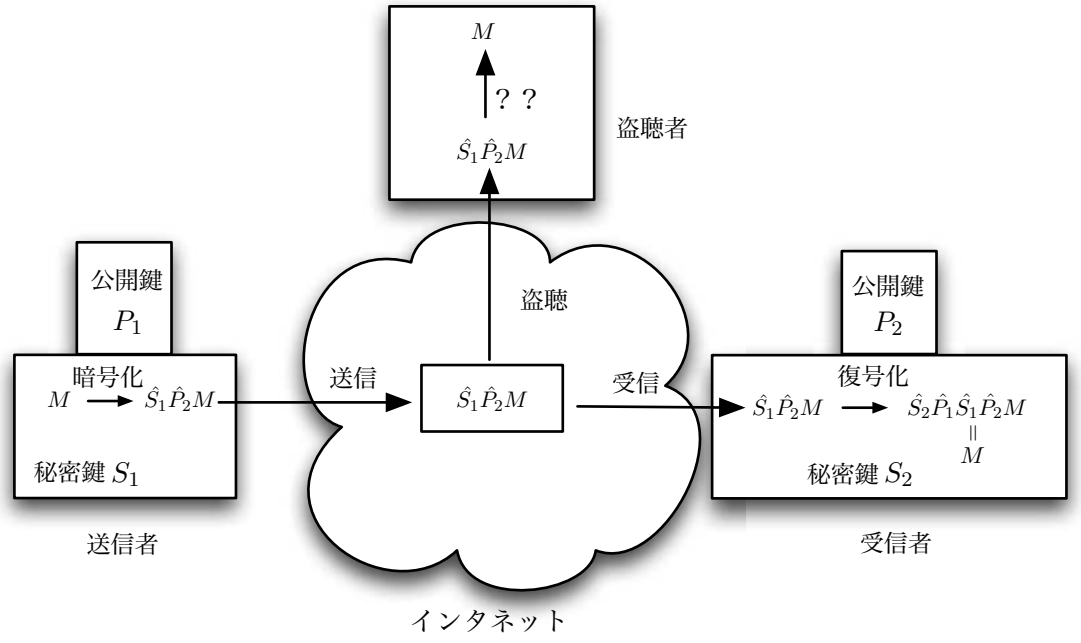
1. M を公開鍵 P で変換して，次に秘密鍵 S で変換する．
2. M を秘密鍵 S で変換して，次に公開鍵 P で変換する．

つまり，

$$\widehat{P}\widehat{S}M = \widehat{S}\widehat{P}M = M.$$

関数の言葉を使えば，$\widehat{P}$ と $\widehat{S}$ は可能なテキストの全体集合上の関数として，互いの他の逆関数になっているということです．

$$\widehat{P}\widehat{S} = \widehat{S}\widehat{P} = \mathrm{id}\,.$$



例として，真二から桃子へ送信する場合を説明しましょう．

$$\widehat{P_1}\widehat{S_1} = \widehat{S_1}\widehat{P_1} = \mathrm{id} \quad (1)$$
$$\widehat{P_2}\widehat{S_2} = \widehat{S_2}\widehat{P_2} = \mathrm{id} \quad (2)$$

ここで，P_1 は真二の公開鍵，　S_1 は真二の秘密鍵を，　P_2 は桃子の公開鍵, S_2 は桃子の秘密鍵です.

さて，真二が桃子に平文（オリジナルなテキスト)M を送りたいとします．真二（送り手）は $M' = \widehat{P_2}\widehat{S_1}M$ と変換し，この M' を桃子宛にネットに送り出します．S_1 は真二の秘密鍵で，P_2 は桃子の公開鍵ですから，真二はこの変換が実行可能です.

一方桃子 (受け手) は，受けとったテキスト M' を　$\widehat{P_1}\widehat{S_2}M'$ と変換します．P_1 は真二の公開鍵で，S_2 は桃子の秘密鍵だから，桃子はこの変換が実行可能です.

結局，桃子は $\widehat{P_1}\widehat{S_2}\widehat{P_2}\widehat{S_1}M$ を得ます．この式は公開鍵と秘密鍵に関する条件と関数の結合律を使って次のように変形できます.

$$\widehat{P_1}\widehat{S_2}\widehat{P_2}\widehat{S_1}M = \widehat{P_1}(\widehat{S_2}\widehat{P_2})\widehat{S_1}M \quad (3)$$
$$= \widehat{P_1}\widehat{S_1}M \quad (4)$$
$$= M \quad (5)$$

つまり，桃子は自分の手元で，真二が書いたオリジナルのテキスト M に復元できました.

ところで，第三者に盗聴される恐れはなかったでしょうか? 受信テキスト $\widehat{P_2}\widehat{S_1}M$ に $\widehat{P_1}\widehat{S_2}$ を適用すればもとに戻ることはわかります．逆に，それ以外の鍵の列の適用では，M に戻せないという保証があれば盗聴されても内容は解読されないことになります．そして，このことは確率的に保証されていなければなりません．ここでは，このことを認めましょう．認めた上で説明を進めます.

P_1 は公開鍵ですから，誰でも使えます．しかし，S_2 は桃子の秘密鍵ですから，S_2 は桃子しか知りません．つまり桃子しか変換 $\widehat{S_2}$ を使えません．したがって，桃子以外の人が関数合成 $\widehat{P_1}\widehat{S_2}$ を使うことはできません．たとえ M を書いた真二といえども，テキスト M' をオリジナルの M に戻すことはできません．なぜなら，S_2 は桃子の秘密鍵なので，真二には利用できないからです.

素因数分解の複雑性

公開鍵方式には根幹となる前提があります．公開鍵 P からその秘密鍵 S を推定することは，事実上不可能なぐらい複雑でなければならないということです．ここで事実上不可能とは，たとえば，どんな早い現実のコンピュータを使っても，計算時間が何万年もかかるぐらいの複雑さです.

この仮定が公開鍵方式の命です．そしてこのような鍵のシステムがすでに提案されており，実際にインタネット上で使われています．たとえばPGP(Pretty Good Privacy) というフリーソフトがそれです．

一般に，自然数のかけ算は簡単に計算できますが，逆に与えられた自然数を因数分解することは困難です．たとえば，13×29 を計算して 377 を得るのは簡単ですが， 377 を因数分解して　13×29 であることを知る方はより手間がかかります．実際，数が大きくなれば，それを因数分解することはもはや計算量的に絶望です．それで，大きな数を公開鍵とし，その素因数を秘密鍵とすることが考えられます．たとえば，377 を公開鍵としてその素因数のひとつ 13 を秘密鍵にします．もちろん，これは説明のための例で，こんな小さな数では，すぐ因数分解されてしまいますので使えません．

数が大きくなれば，因数分解は困難であるという，自然数の素因数分解のこの性質は，実際にも公開鍵方式の有力なひとつの理論的基盤となっています．

このように，公開鍵方式のシステムには，長年数学者が彼ら自身の純粋な興味で積み上げてきた整数論が有効に使われています．ここにも純粋数学の思わぬ有用性の例がみられ，これ自体，興味深いものがあります．しかし，数学的内容については省略します．

署名の認証

インタネットでは「なりすまし」が簡単にできます．だれかが,「真二」と偽って桃子にメールを送るかもしれません．では，受け取ったメッセージが本当に真二から来たのかどうかを確かめるためにはどうしたらよいでしょう？

公開鍵方式での認証は，ユーザ，たとえば真二は，自分の署名 L を秘密鍵 S_1 を使い，$\widehat{S_1}L$ と暗号化してから送ります．こうすれば，暗号化された署名を真二の公開鍵 P_1 で復号化し，それが真二の署名 L であることを目で確認できれば，真二から送られてきたものであることが認証できます．

なぜなら,「なりすましの真二」には，真二の秘密鍵 S_1 が使えないので，暗号化署名 $\widehat{S_1}L$ が作れません．したがって，偽の真二から送られてきた暗号化署名に $\widehat{P_1}$ を適用しても L—-"真二"—になりません．真二の公開鍵で復号して"真二"になるのは，本物の真二から来たきた場合のみであることも明らかです．

公開鍵方式のまとめ

公開鍵方式の数学的なモデルを示します．

D: 可能なメッセージの全体集合を D とおく．

U: ユーザの集合

K: 鍵の集合

$B(D)$: D から D への全単射の全体集合

$\widehat{\cdot}$: $K \longrightarrow B(D)$　　鍵は D 上の全単射として解釈される.

p: $U \longrightarrow K$ (公開鍵)

q: $U \longrightarrow K$ (秘密鍵)

つまり, 次のことが成り立つようにします. $u, v \in U$ は任意のユーザを表します.

- u は公開鍵 $p(u) \in K$ と秘密鍵 $q(u) \in K$ を持つ.
- D 上の関数 $\widehat{p(u)}$ と $\widehat{p(u)}$ は互いに逆関数である.
- u から $v \in U$ へ平文 d を送信するときは, u は平文 d を　$e = \widehat{p(v)}\widehat{q(u)}d$ に暗号化してから v へ送る.
- v が u から暗号文 e を受信したならば, $\widehat{p(u)}\widehat{q(v)}e$ と復号する.

送信者と受信者の間で暗号化と復号化がうまくはたらくことは, すでに述べたように逆関数の性質から明らかでしょう.

この公開鍵方式が安全であるということはどう定義すべきでしょうか？暗号文を第3者 w に復号化されないこととすればよいでしょう. すなわち第3者が, 公開鍵と必要ならば彼自身の秘密鍵をどのよう組み合わせて関数結合しても復号化のための鍵を合成できないこと, あるいはコスト的に不可能であることと定義すればよいでしょう. たとえば, $u, v, w \in U$ について, 次のふたつの条件が成り立っていなければなりません.

1. $q(v)p(w)q(u)p(v) = \mathrm{id}$　　$(w = u)$
2. $q(v)p(w)q(u)p(v) \neq \mathrm{id}$　　$(w \neq u)$

この条件 1 は, u が v へメッセージを暗号して送信するとき, v はそれを復号できることを意味しています. 条件 2 は第 3 者の w がこの暗号-復号の手順にそって「復号」してもオリジナルの平文にはもどらないことを示しています.

注意 1　この条件 2 は, 両辺が関数としては異なるということです. ですから, 論理的には,「もどらない平文が存在する」と読むのが正しい. しかし, 実際には,「ほとんどすべての平文についてもとに戻ることはない」と仮定してよいでしょう.「猫がピアノの鍵盤の上で踊ったときに, そのときに生じる「旋律」がベートーベンの月光と一致する確率」と同じくらいに無視できるといってもよいでしょう. つまり, 条件 2 は現実的には第三者には解読できないことを表現しています.