

本文章は筆者が執筆しDr. Dobb's Journal Japan (翔泳社). 1996年5月号に掲載した「暗号への招待 / すずきひろのぶ」の原稿をベースに作成したものです。

暗号のおはなし

1997年9月バージョン

すずきひろのぶ

hironobu@h2np.suginami.tokyo.jp

暗号技術が何故今クローズアップされるのか？

今まで、暗号という、どちらかと言えば特殊な分野であった技術がクローズアップされてきています。この点に関してまず考えてみましょう。現在、世界中を結んだネットワークであるインターネットには莫大なコンピュータが接続され、これまた莫大な利用者がいます。初期のコンピューティング環境は、ハードウェア的にもソフトウェア的にも自己完結して孤立したシステム上に構築されていました。今日的なコンピューティング環境は、ネットワークとネットワークに接続された複数のハードウェア、およびソフトウェアが混然一体となったシステム上に構築されています。さらには、インターネットという世界中がコンピュータ・ネットワークで結ばれたコンピューティング環境を利用できるまでに至りました。そこにはサイバースペースと呼ばれる巨大な情報空間が生まれ、そこを我々は一般社会の外延としての情報が構築する社会として使うようになりました。

サイバースペースでは、過去のコンピューティング環境の延長線では考えられない状況が現れます。例えば、ネットワークを介してコンピュータ間でデータを交換するという行為は、公道で立ち話しているのと同じ程度の行為にしかありません。故意であろうと、偶然であろうと、誰かに立ち話を聞かれる可能性があるように、サイバースペースでは何も対策を施さなければ、データ転送の途中で内容が漏れる可能性が同様にあります。外部者に見られてはいけない書類やメモを、机の上に出しっぱなしにしたり、道を歩きながら読んだりすれば、故意であろうと、偶然であろうと誰かに読まれたりする可能性があるように、サイバースペースでは何も対策を施さなければ、データが誰かに読まれる可能性があります。

それだけコンピューティング環境というものが、我々の生活や社会の様式に近づいてきたと言えるのかも知れませんが、とにかく、過去のように無頓着でいられなくなった

のは事実のようです。また、エレクトリック・コマースのように外部に開かれたネットワーク上、つまりサイバースペース上で買い物をするような時には、必ず暗号によって、その情報を保護しなければいけません。もし、そうでなければ、公園の人混みの中で自分のクレジットカードを読み上げているのと同じことをサイバースペースで行なっていることになります。

アリスとボブ~サイバースペースの住民

暗号の技術を紹介する前に、サイバースペースで暗号技術を使うとどのようなことができるのかを分かりやすくするために、ストーリーを考えてみましょう。このストーリーでは、サイバースペースの住民であるアリスとボブの二人が主人公です。ここでのサイバースペースの住民とは、それは物理的存在を指しているのではなく、情報的な存在を意味しています。これは、実際のやりとりが人間と人間、プログラムとプログラム、もちろん人間とプログラムでも構わないという意味です。

アリスからのメッセージ

アリスからボブにメッセージ（情報）を送り、そのメッセージをボブが受け取ったとします。しかし、そのメッセージが本当にアリスからのメッセージであることをボブは確信できるのでしょうか？メッセージがアリスを離れ、ボブに伝わるまでのあいだに誰かがメッセージを改ざんしているかも知れません。また、ボブをだますために、他人がアリスを装い書いたメッセージかも知れません。

このような問題はアリスはメッセージにデジタル署名（Digital Signature）を使い署名することによって解決します。ボブはメッセージとデジタル署名を検証すれば、メッセージが本当にアリスからのものか、途中で改ざんされていないか、といったことがわ

かります。

アリスがアリスであること

アリスは自分がアリスであるということを証明するアリスしか知らない情報をもっています。しかし、アリスがアリスであることを証明するために、アリスしか知らない情報を明かしてしまえば、今後、明かされた情報を使って自分がアリスだと偽る人が現れるかも知れません。そうなった場合、アリスと偽アリスのもっている情報はまったく同等であるため、どちらが本当のアリスであるかをボブは判断できません。では、アリスの秘密を誰にも知られずに、かつ、アリスがアリスであることをどうボブに教えることができるのでしょうか？

このような場合、ゼロ知識証明 (Zero-Knowledge Proof) を用いれば、アリスはアリスしか知らない情報を誰にも明かすことなく、ボブにアリスがアリスであることを教えることができます。

ボブだけに教えたい

アリスはボブだけに秘密を伝えたいので、メッセージを暗号化してから送るようにしようと考えました。アリスは、暗号を使うことをボブに提案しました。使う暗号は、暗号化する鍵と復号化する鍵が同じものを使う従来からの暗号方式としました。これは慣用暗号 (Conventional Cryptography) あるいは対称暗号 (Symmetric Cryptography) と呼ばれる方式です。アリスはボブに復号化のために (と、同時に暗号化もできる) 鍵を送ろうとしましたが、思い留まりました。なぜなら、もし万が一、この鍵を送る途中で誰かの手に渡ることがあれば、その鍵の複製を使って、ボブへ送る暗号化したメッセージをいくらでも読むことができることに気がついたのです。そうなれば、暗号は何の役

にも立ちません。

そこでボブは公開鍵暗号法(Public Key Cryptography)を使うことを提案しました。公開鍵暗号法は暗号化する鍵と復号化する鍵が別々になっています。アリス自身だけが復号化する鍵を持ちます。アリスは暗号化する鍵をボブだけではなく、誰に公開しようとも、ボブからの暗号化されたメッセージはアリスしか復号化できません。

このようにサイバースペースでは暗号技術は必要不可欠な技術だということがわかります。

どのような暗号があるのか

先に暗号に関する感覚をつかんでもらうため、技術的な話題より先に利用のストーリーを書いてしまいましたが、改めて、コンピュータ上で使われている暗号アルゴリズムの分類と、代表的な暗号システムを紹介したいと思います。現在、広く使われている暗号アルゴリズムには2つのアルゴリズムがあります。それは、対称アルゴリズムと公開鍵アルゴリズムです。

* 対称アルゴリズム (Symmetric Algorithm)

暗号化する鍵と復号化する鍵が同じ鍵であるようなアルゴリズムです。対称アルゴリズムでは、データの扱いより、さらに2つの方法に分類されます。

- 逐次アルゴリズム(Stream Algorithm)

メッセージをビット列もしくはバイト列として扱う方法です。ヨーロッパのデジタル携帯電話に使われている標準的な暗号方式であるA5、アップル社のAOCEやオラクル社のSecure SQLといった

製品に採用されているRSA社のRC4などが有名です。

- **ブロックアルゴリズム (Block Algorithm)**

メッセージを何ビットかでブロック化し扱う方法です。米国のデータ暗号化規格であるDES、NTTが開発し日本国内外で広く使われているFEAL、RSA社のRC5、PGPに使われているIDEAなどが有名です。

* **公開鍵アルゴリズム (Public-Key Algorithm)**

公開鍵アルゴリズムは非対称アルゴリズム(Asymmetric Algorithm)とも呼ばれ、暗号化する鍵(公開鍵)と復号化する鍵(秘匿鍵)が異なる鍵であるようなアルゴリズムです。1976年にディフィー(Diffie)とヘルマン(Hellman)が公開鍵暗号方式を考えついたことから始まります。

現在の代表的なものとしていくつか紹介します。大きな数の素因数分解の困難さを利用したRSA暗号(発明者のRivest, Shamir, Adlemanの頭文字を取って命名されている)は、デ・ファクト・スタンダードといえるもので、各種のプログラムに応用されるとともに、RSA社の製品として広く市場で利用されています。離散対数問題の困難さを利用したElGamal暗号。離散対数問題の一種である楕円曲線を使った楕円暗号(Elliptic Curve Encryption)はRSA暗号よりも暗号化/復号化の計算量が少ない暗号法です。また、RSA暗号ほど安全性に関する実績はありませんが、安全性の研究もかなりすすんでおり、今後、重要な技術となるでしょう。NeXT社(現アップル)のFEE(Fast Elliptic Encryption)、松下電器産業のMyEllityなどがありますが、その他

にもいくつかの会社が開発を行なっています。

安全に鍵交換を行なうDiffie-Hellman法も公開鍵暗号法として利用できます。

アリスとボブのストーリーの中に、デジタル署名やゼロ知識証明といったテーマを取り上げましたが、公開鍵アルゴリズムは相手に秘密の情報を与えずとも必要な情報交換を安全に行えるという側面をもっています。

対称暗号と公開鍵暗号の使い分け

鍵の管理が安全な公開鍵暗号も残念ながら暗号化 / 復号化の処理速度が遅いという欠点があります。DESとRSAを比べた場合、ソフトウェアで約100倍の処理速度の差があります。DESはハードウェア向きの構造をしているので、各々をチップ化したものを比較すると約1000倍の処理速度の差が出てしまいます。この違いを考慮した上で、実際のシステムでは、例えば、サーバー・クライアント間で安全なデータ転送を行いたい場合、最初のコネクション時に公開鍵暗号方式を使って対称暗号で使う鍵を交換し、実際のデータ転送には高速な対称暗号を使うといった工夫がされています。

暗号の安全性とは何か

現代の暗号の安全性は「暗号を解読するためには莫大な計算が必要になる」ということによって支えられています。その「莫大な計算が必要になる」という意味は、裏を返せば、莫大な計算能力を提供できれば解読できるという意味でもあります。現在の最高の計算技術と可能な限りの計算資源を与えて暗号を解読するのに数万年もの時間がかかるなら文字通り「安全」といえるでしょう。一方、暗号化された10円の価値しかない

(または10円の損害しか与えられない)情報に対し、その解読のコストが10億円かかるとすると、経済的な意味において「安全」と呼んでも構わないでしょう。

しかし、「莫大な計算が必要になる」という証明は、実は非常に難しいのです。暗号の強度や穴がないかを調べることを暗号分析(Cryptanalysis)といますが、暗号分析の結果、あっさり破られるという話も実はよくある話なのです。世の中には次から次へとたくさんの暗号方式が提案されているのですが、その中で無傷で暗号方式が生き残るとするのは非常に大変なことなのです。

ブロック暗号に対し、1990年にビハム(Biham)とシャミア(Shamir)は差分攻撃法(Differential cryptanalysis)を考え出しました。差分攻撃法がターゲットとしていたDESは設計時に既に差分攻撃法を想定した設計がされておりDESの安全性は保たれました。つまり、DESの設計グループは1974年当時に、ビハムとシャミアが考えつく15年以上前に同じ解読法を考えついていたわけです。一方、FEALは差分攻撃法には弱いという問題点が明らかになりました。

そのDESも1993年に日本の暗号研究者である松井充氏(三菱電機)が発表した線形攻撃法によって、その安全性が怪しくなってきました。1995年にNTTの暗号研究のグループが松井氏のアルゴリズムを改良し、より高速に処理ができることを示しました。

理論的分析だけではなく、最近のCPUの演算能力の飛躍的向上や分散処理技術の向上により、具体的な計算可能である範囲も広がってきます。公開鍵暗号の1つであるRSA法は、(大きな2つの素数をかけあわせた、さらに)大きい数を素因数分解するには莫大な計算量が必要であるという事実によって安全性を保っています。RSA法の開発者は、この暗号方法で作った暗号文に賞金を懸けました。その1977年当時、大きい数として選んだ桁数は129桁でした。これで最低でも50年は破られまいと考えたと

言われています。17年後の1994年4月、129桁は破られました。「莫大な計算量」をこなすために、世界中からボランティアを募り、たくさんのコンピュータを使い、分散して処理を行ったために可能になったのです。インターネットで世界中が結ばれていてコンピューティング資源を提供/共有できたという、1977年当時では考えられなかった新しいコンピューティング・スタイルが台頭してきたからこそ可能になったのです(これこそサイバースペースと呼べるものです!)。

また、DESによって暗号化した暗号文を解読するというチャレンジでは、インターネットを通じてパソコンも含めた大量の計算機を動員し、鍵の組み合わせパターンを片っ端から試してみる力づくの方法(ブルートフォースと呼びます)で、147日かけて解読しました。この際、可能な鍵の組み合わせの24.6%を試した時点で見つかったという幸運はあったのですが、しかし24.6%といえども一京八千兆個の(1.8×10^{15} 個)鍵を試す必要がありました。

ちなみに、公開鍵暗号方式のRSA法を用いた場合、通常の使用目的であれば、現在は1024ビット長(大体300桁程度)の大きさで十分に安全性を保てると言われています。しかし、暗号解読に使う技術とハードウェアの進化は速く、2014年には2048ビット長を用いなければ、安全性は保てないと予想されています。ちなみに鍵が短くて済む楕円暗号は、RSA法で1024ビット長にあたる安全性が160ビット長で実現され、またRSA法で2048ビット長にあたる安全性が210ビット長で実現できるという利点があります。

将来、量子コンピュータや、DNAコンピュータなどといった、今までのコンピュータとはまったく別の計算手法が実用化される可能性があります。ですから、今日、安全だといっている暗号が、本当にいつまで安全であるかは、実は誰もわからないのです。

(ひ)

Oct/27/1997

Author: Hironobu SUZUKI