インターネットにおけるセキュリティ――暗号 cryptography

暗号の歴史と重要性

バビントン陰謀事件(1586)

イングランド女王エリザベスに対する反逆罪に問われたスコットランド女王メアリが処刑された事例。イングランド王位に対する継承権も持っていたメアリは、スコットランドにおける政変で失脚しイングランド国内に幽閉されていた。外部の協力者がエリザベスを暗殺してメアリを王位につける計画を持ちかけたところ、彼女はその計画に同意した。しかし陰謀に関して交わした暗号化書簡は、実はエリザベスの治安官吏によってすべて解読されていた。これが犯罪の証明となり、メアリは斬首された。

第二次世界大戦

連合国側・同盟国側とも、相手国の通信内容の解読に積極的に取り組み、かなりの成功を挙げていた。太平洋戦争開戦前に行なわれた日米外交交渉は結局決裂するが、一因として日本の外交暗号が解読されていたことが指摘されている(アメリカ側は日本の想定する「落としどころ」を最初から察知して強気の交渉に臨んだ)。また、前線視察のために搭乗した航空機が撃墜され山本五十六(連合艦隊司令長官)が殉職した事件も、山本の予定を伝える暗号通信が解読されたことが原因と言われている。

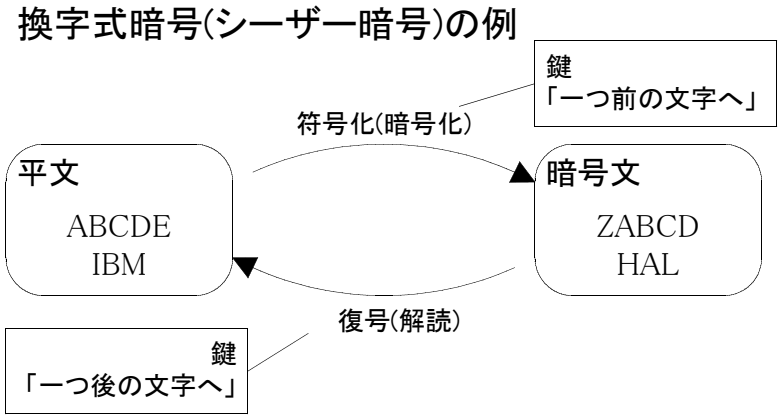
暗号解読と文学

また、暗号解読はミステリの古典的テーマのひとつにもなっている。換字式暗号を頻度分析によって解読している事例としては、エドガー・アラン・ポー「黄金虫」。同様の手法で記号を用いた暗号を解読している例に江戸川乱歩「二銭銅貨」、コナン・ドイル「踊る人形」がある。

初歩的な暗号・共有鍵暗号

簡単な例と用語……シーザー暗号

名称はユリウス・カエサルが使ったと言われていることから。
映画『2001 年宇宙の旅』に登場するスーパーコンピュータ「HAL」は、IBM を一文字分「進歩」させたシーザー暗号だと言われている。



アルゴリズム(暗号化の方法)
鍵(暗号化の際に使われたタネ)
→ この両者がわかれば解読可能。

わからない場合
→ 総当たり方式　　可能な鍵のパターンをすべて試してみることで、力業による解読が可能。
この暗号方式の安全性は、鍵が秘密であることに依存している。

＝秘密鍵暗号方式

共有鍵暗号方式(対称暗号)

暗号化・復号に使う鍵が同じもの(典型がシーザー暗号)。

DES（Data Encryption Standard)：1977 年、アメリカ連邦規格に採用。1981 年、ANSI 標準。56bit の鍵を使用する共有鍵暗号方式。公募によって開発されたが、採用されたのは NSA の提出したアルゴリズム。

スキップジャック：NSA が民間向けに開発した共有鍵暗号方式の非公開アルゴリズム。クリッパーチップに実装される。鍵長 80bit。

公開鍵暗号方式(非対称暗号)

暗号化・復号に使う鍵をあらかじめ公開しておくことができる暗号化方式。

最初の公開鍵暗号: Whitfield Diffie・Martin Hellman（1976)。

RSA 暗号: 1976 年、Ronald Rivest, Adi Shamir, Len Adleman（MIT)により開発された公開鍵方式の暗号。「2 つの大きな素数の積を計算して合成数を作るのは簡単だが、その合成数を元の素因数に分解するのは非常に難しい」。

簡単に説明すると……

準備

それぞれの人が一組の鍵(X, Y)を作る。

X で鍵をかけると、Y でしか開かない。Y で鍵をかけると、X でしか開かない。

片方(X)を公開鍵(誰でも見られる)、もう片方(Y)を秘密鍵(本人だけが知っている)にする。

↓

1）A さんから B さんへの通信を暗号化する。

A さんは送りたい情報を、B さんの公開鍵(Xb)で暗号化する。

B さんは受け取った情報を、B さんの秘密鍵(Yb)で解読する。

…… Yb を持っているのは B さんだけなので、本人だけが読むことができる。

2）A さんから B さんへの通信を本人のものだと証明する(本人認証)。

A さんは送りたい情報を、A さんの秘密鍵(Xa)で暗号化する。

B さんは受け取った情報を、A さんの公開鍵(Ya)で解読する。

…… Ya で開けられる鍵は A さんだけが持つ Xa でしかかけられない → 本人からの通信。

1・2 を組み合わせれば、本人認証と暗号化を同時に行うことも可能。

何が違うのか？―― 両者の比較

鍵伝達の安全性

共有鍵暗号の場合、事前に鍵を相手と共有しておく必要がある。

∵通信を暗号化する鍵を、暗号化されていない通信経路で送るわけにはいかないから。

第二次世界大戦中のドイツ軍最高司令部は、すべてのエニグマ・オペレーターに日鍵を掲載した本を毎月配達しなければならず、これは兵站上の大問題だった。U ボートは長期間基地を離れることになりがちだったが、それでもどうにかして定期的に鍵を届けなければならなかった。(シン 335)

1970 年代、COMSEC の扱う鍵は日々何トン分にものぼった。COMSEC の物資を積んだ船が港に着くと、暗号管理者が続々と船に乗り込み、カードや紙テープ、フロッピーディスクなど、鍵を保存してあるメディアを受け取り、それを正当な受取人に届けていたのである。(シン 335)

鍵配布局を使った場合、そこが侵入の主たる標的となる。

アメリカでは軍・諜報機関対称の鍵配布局だった NSA で暗号鍵を売るスパイ事件が発生している(Ronald W. Pelton)

公開鍵暗号の場合、漏洩が問題となる鍵は一切伝達される必要がない。

総鍵数の有利

共有鍵暗号の場合、必要となる総鍵数は n(n-1)/2。

公開鍵暗号の場合、必要となる総鍵数は 2n。

→ n>5 のとき n(n-1)/2 > 2n。6 人以上の社会では公開鍵暗号が有利。

●暗号の危険性と安全性

RSA 暗号を総当たり方式で解読した実験の例。

1977 年、Martin Gardner の出題(RSA-129: 鍵は 129 桁、429bit)。

1993 年、Arjen Lenstra(Bellcore)・Derek Atikins(MIT)が実験を組織。24 ヶ国から 600 人を超える参加者を組織し、合計 1600 台のワークステーション等々で 8 ヶ月間かかって解読に成功(1994 年 4 月 26 日公表)。

出題された鍵

N=114,381,625,757,888,867,669,235,779,976,146,612,010,218,296,721,242,362,562,561,842,935,706,935,245,733,897,830,597,123,563,958,705,058,989,075,147,599,290,026,879,543,541

解読された結果

q=3,490,529,510,847,650,949,147,849,619,903,898,133,417,764,638,493,387,843,990,820,577

p=32,769,132,993,266,709,549,961,988,190,834,461,413,177,642,967,992,942,539,798,288,533

RSA Secret-Key Challenge

1997 年、RSA 社が開始。鍵長 56bit のものは同年 10 月に解読成功(全体の約 47%を試行した時点)。

RC5 cracking

RSA Data Security 社が主催。鍵長 64bit、総当たりの場合の数 2^64 = 18,446,744,073,709,551,616 通り(地球の表面積とコンタクトレンズの面積の比におよそ等しい)。

DES-II-1

RSA 社主催、鍵長 56bit。1998 年 1 月 13 日スタート、同 2 月 23 日解読。39 日間で解読成功(鍵空間の約 90%を探索)。稼働したマシンパワーの総計、Intel Pentium II 333MHz 換算で 22393 台。

DES-II-2

1998 年 7 月 13 日スタート。RSA 社主催で鍵長 56bit。EFF(Electoronic Frontier Foundation)のチームが、カスタムメイドのワークステーション 1 台・56 時間で解読に成功(しかし 25 万ドルはしないらしい)。クリントン政権は暗号の輸出規制で鍵長 40bit までに制限。56bit key のものはその 2^{16} 倍安全なはず。

●アルゴリズムは公開されない方が安全か?

Merkle-Hellman 暗号の例

Ralph Merkle (Stanford)、Martin Hellman(Paro Alto)という一流の暗号学者 2 名によって開発され、1977 年に特許申請された Merkle-Hellman 暗号は、ナップサック問題(重さの異なるいくつかの品物が与えられたとき、特定の重さになるようにナップサックに品物を詰めることができるかという問題)に基づく暗号であった。しかしその後の研究で、総当たり方式をとる必要はなく、アルゴリズム解析によって解読できる(つまり公開鍵暗号のための方式としては役に立たない)ことが証明された。

1978 年、Herlestam による危険性の指摘。

1979 年、Adi Shamir が特定条件下で解読可能性があることを指摘。

1982 年、Len Adleman (MIT)が解読プログラムを Apple II 上で動作させることに成功。

次のことを忘れないでいただきたい。暗号の安全性がアルゴリズムを秘密にすることによってのみ実現されている場合、攻撃者は単にそのプログラムのコピーを購入して内容を解析しさえすれば、アルゴリズムの動作方法を知ることができるのだ。アルゴリズムを破れば、秘密の通信を復号化できる。(Garfinkel 52)

イニシアティブの問題

Linux の法則(Eric S. Raymond): Given enough eyeballs, all bugs are shallow.

……欠陥を見つける方法は、できるだけ多くの人間がチェックするしかない。