第2篇　分组密码分析方法

第3章　朴素密码分析方法

本章将介绍几个朴素、通用的密码分析方法，也被统称为强力攻击。它们对任何分组密码都适用，且攻击的复杂度只依赖于分组长度和密钥长度。更严格一些地讲，攻击所需的时间复杂度依赖于分组密码的工作效率（包括加解密速度、密钥扩展速度以及存储空间等）。

3.1　引言

密码是用来对明文提供保护的，防止不期望的明文泄露，而密码分析人员的任务是在某种意义下破译密码。如果密码分析者能确定该密码所正在使用的密钥，以至于他能像合法用户一样阅读所有的消息，则称该密码是完全可破译的。如果密码分析者仅能频繁地从所窃获的密文恢复明文，但他却不能发现密钥，则称该密码是部分可破译的。

3.1.1　无条件安全性和计算安全性

Shannon在文献[210]中提出了无条件安全性（又被称为完善保密性）的概念：如果对于所有明文 P 和密文 C，都有 $\Pr(P)=\Pr(P|C)$ 成立（$\Pr(P)$ 表示明文 P 在消息空间上的分布概率，$\Pr(P|C)$ 表示条件概率），就称该分组密码关于当前密钥具有无条件安全性。在无条件安全性的模型下，假定攻击者具有无限计算资源（如时间、空间、设备和资金等）。要达到无条件安全性，一个最基本的要求是：密钥长度至少要和待加密的消息的总长度相等，一个密钥比特只使用一次，这在大多数情况下是不切实际的。实际中，一个固定的密钥可以用来加密很多个明文块。现代密码学中，通常是在计算安全性的模型下研究密码的安全性。一个密码系统是计算上安全的，指的是利用已有的最好的方法破译该系统所需要的努力超越了攻击者的破译能力（或者破译该系统的难度等价于求解数学上的某个已知难题，多适用于公钥密码学）。当然，这只是提供了系统是计算上安全的一些证据，并没有真正证明系统是计算上安全的。显然，在计算安全性的模型下，假定攻击者拥有的计算资源是有限的。

3.1.2 攻击的分类

在密码学中,“分析(analysis)”和“攻击(attack)”这两个术语含义相同,本书将交替使用这两个术语。

在密码分析学中,人们总是假定攻击者可以截获在不安全信道上所传输的所有密文。另一个容易被人们接受的假设是 Kerckhoff 假设:除了密钥之外,攻击者知道所有有关加密和解密的详细过程。Kerckhoff 假设蕴涵着密码的安全性完全依赖于密钥。在 Kerckhoff 假设下,根据攻击者所掌握的信息,可将分组密码的攻击分为以下几类。

唯密文攻击:攻击者除了所截获的密文,没有其他可利用的信息。

已知明文攻击:攻击者仅知道当前密钥下的一些明/密文对。

选择明文攻击:攻击者能获得当前密钥下的一些特定的明文所对应的密文。进一步如果明文基于以前获取的密文,则称之为适应性选择明文攻击;否则,称之为非适应性选择明文攻击。

选择密文攻击:攻击者能获得当前密钥下的一些特定的密文所对应的明文。进一步,如果密文基于以前获取的明文,则称之为适应性选择密文攻击;否则,称之为非适应性选择密文攻击。

显然,在上述几类攻击中,选择明(密)文攻击是密码分析者可能发动的最强有力的攻击。在许多场合这种攻击是不现实的,如果明文空间含有冗余,攻击者就很难欺骗合法用户对某些无意义明文进行加密。但如果密码算法在这种攻击下是安全的,则在其他攻击下也一定是安全的。因此尽管攻击者很少有发动选择明(密)文攻击的机会,设计者仍希望其密码算法能抵抗选择明(密)文攻击。另外,在某些场合,攻击者虽然不知密钥,但他知道或可以选取加密或解密算法所使用的密钥之间的关系,来发动相应的选择明(密)文等攻击,这就是近几年研究比较多的相关密钥攻击。

3.1.3 攻击的复杂度

一个攻击的有效性通常由实施该攻击所需的时间复杂度、空间复杂度和数据复杂度来衡量。数据复杂度是实施该攻击所需输入的数据量,已知明文攻击(或选择明文攻击)的数据复杂度可以用攻击中所需要的已知(或选择)明文/密文对的数量来确定;空间复杂度是攻击算法所需的存储量;时间复杂度是实施攻击所需的计算步骤,通常由加密、解密次数表示。一般地,空间比时间更昂贵。比如,一个需要 2^{64} 存储量的攻击相比于一个需要 2^{64} 步骤的算法,其代价要昂贵许多。数据量也比较昂贵,应尽量减少。在一些情况下,可以利用增加时间复杂度的方法减少空间复杂度和数据复杂度。

一般地,用数据复杂度和时间复杂度的主要部分来刻画攻击的复杂度。例如:在穷尽密钥搜索攻击中,所需要的数据量与计算量相比是微不足道的,因此,穷尽密钥搜索攻击的复杂度实际是时间复杂度。另一个例子是 Biham 和 Shamir 的差分密码分析,差分密码分析是一种选择明文攻击,其复杂度主要由该攻击所需的明/密文对的数量来确定,而实施该攻击所需的计算量相对来说是比较小的,因此,差分密码分析的复杂度是其数据复杂度。一般地,对分组长度为 n 比特和密钥长度为 k 比特的分组密码来说,一种已知明

文攻击(或选择明文攻击)的数据复杂度可由实施该攻击所需的已知(或选择)明文对的个数来度量,对一个固定的密钥而言,这样的明/密文对的个数最多是 2^n。考虑到生日攻击,文献[211]中建议,一个密钥最多用来加密 $2^{n/2}$ 个明文块,这个限制与密钥长度无关。

另外攻击的成功率、获取信息的类型以及数量也是衡量攻击有效性的参数,成功率是指攻击过程执行结束后实现目标的概率。攻击成功获取的信息可能有密钥比特、等价密钥、明文碎片等。

3.2 穷尽密钥搜索攻击

设 k 是密钥长度,在唯密文攻击下,攻击者依次试用密钥空间中所有 2^k 个密钥解密一个或多个截获的密文,直至得到一个或多个有意义的明文块。在已知(选择)明文攻击下,攻击者试用密钥空间中的所有 2^k 个密钥对一个已知明文加密,将加密结果同该明文相对应的已知密文比较,直至两者相等,然后再用其他几个已知明密文对来验证该密钥的正确性。

穷尽密钥搜索的复杂度平均为 2^{k-1} 次加密。

3.3 字典攻击

攻击者搜集明密文对,并把它们编排成一个“字典”。攻击者看见密文时,检查这个密文是否在字典里,如果在,他就获得了该密文相对应的明文。如果 n 是分组长度,那么字典攻击需要 2^n 个明密文对才能使攻击者在不知道密钥的情况下加解密任何消息。

3.4 查表攻击

设 k 是密钥长度,查表法采用选择明文攻击,其基本观点是:对一个给定的明文 x,用所有 2^k 个密钥 K(记其全体为 K),预计算密文 $y_K = E_K(x)$。构造一张有序对表 $\{(y_K, K)\}_{K\in K}$,以 y_K 给出 K 的标号。因此,对于给定的密文,攻击者只需从存储空间中找出相对应的密钥 K 即可。

3.5 时间存储折中攻击

时间存储折中(Time-memory trade-off)攻击是一种选择明文攻击方法,它由穷尽密钥搜索攻击和查表攻击两种方法混合而成,它在选择明文攻击中以时间换取空间。它比穷尽密钥搜索攻击的时间复杂度小,比查表攻击的空间复杂度小。我们以 DES 为例介绍时间存储折中攻击方法。

设 $R:\{0,1\}^{64}\to\{0,1\}^{56}$ 是一约化函数,将 64 比特串变为 56 比特串,比如简单地删掉 64 比特串的最后 8 比特。令 P_0 是一固定的 64 比特明文,定义函数:

$$g:\{0,1\}^{56}\to\{0,1\}^{56}$$

$$K \to g(K) = R(E_K(P_0))$$

在时间存储折中攻击的预处理阶段，攻击者首先随机选取 m 个长为 56 比特的串，记为 $X(i,0)(1\leqslant i\leqslant m)$；然后攻击者根据递推关系：

$$X(i,j) = g(X(i,j-1)) \tag{3-1}$$

计算 $X(i,j),(1\leqslant j\leqslant t)$，记 $X=(X(i,j))_{1\leqslant i\leqslant m,1\leqslant j\leqslant t}$。

最后攻击者构造一张有序对表 $T(P_0)=\{(X(i,0),X(i,t))\}_{1\leqslant i\leqslant m}$。表 $T(P_0)$所需的存储量为 $O(m)$，预计算时间为 $O(mt)$。

$$\begin{array}{l} X(1,0)\xrightarrow{g}X(1,1)\xrightarrow{g}\cdots\xrightarrow{g}X(1,t) \\ X(2,0)\xrightarrow{g}X(2,1)\xrightarrow{g}\cdots\xrightarrow{g}X(2,t) \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ X(m,0)\xrightarrow{g}X(m,1)\xrightarrow{g}\cdots\xrightarrow{g}X(m,t) \end{array}$$

图 3-1　$X=(X(i,j))_{1\leqslant i\leqslant m,1\leqslant j\leqslant t}$的计算

在时间存储折中攻击的攻击阶段，假定攻击者获得了明文 P_0 在某个密钥 K_0 作用下的密文 $C_0=\mathrm{DES}_{K_0}(P_0)$，它利用表 $T(P_0)$以如下方式恢复密钥 K_0。

首先攻击者计算 $Y_1=R(C_0)$，然后检查 $Y_1\in\{X(i,t),1\leqslant i\leqslant t\}$是否成立，如果成立，也就是存在 i_0，使得 $Y_1=X(i_0,t)$，则认为密钥 $K_0=X(i_0,t-1)$，攻击者利用等式(3-1)以 $X(i_0,0)$为起点递推计算 $K_0=X(i_0,t-1)$。

如果 $Y_1\notin\{X(i,t),1\leqslant i\leqslant t\}$，攻击者继续计算 $Y_2=g(Y_1)$，检查 $Y_2\in\{X(i,t),1\leqslant i\leqslant t\}$是否成立，如果成立，也就是存在 i_1，使得 $Y_2=X(i_1,t)$，则认为密钥 $K_0=X(i_1,t-2)$，攻击者利用等式(3-1)以 $X(i_1,0)$为起点递推计算 $K_0=X(i_1,t-2)$。

类似地，如果 $Y_2\notin\{X(i,t),1\leqslant i\leqslant t\}$，攻击者继续计算 $Y_3=g(Y_2),\cdots,Y_t=g(Y_{t-1})$，并检测 K_0 是否在表 3－1 的 $t-3$ 列，…，0 列。

注意，若 $K_0=X(i_0,t-1)$，则 $Y_1=X(i_0,t)$。但若 $Y_1=X(i_0,t)$，则未必有 $K_0=X(i_0,t-1)$；因为约化函数 R 不是单射，平均每个像有 $2^8=256$ 个原像。所以我们需要检查 $C_0=\mathrm{DES}_{X(i_0,t-1)}(P_0)$是否成立，来确定 $X(i_0,t-1)$是否真正的密钥。上述过程可描述为以下步骤。

第 1 步：计算 $Y_1=R(C_0)$。

第 2 步：如果对某一 $i(1\leqslant i\leqslant m)$，$Y_j=X(i,t)$，那么从 $X(i,0)$出发，将函数 g 迭代 $t-j$ 次，计算 $X(i,t-j)$。如果 $C_0=\mathrm{DES}_{X(i,t-j)}(P_0)$，那么置 $K=X(i,t-j)$并停机。否则转入第 3 步。

第 3 步：计算 $Y_{j+1}=g(Y_j)$。如果 $j\leqslant t$，转入第 3 步；否则停机。

文献[212]中已证明，若 $mt^2=N=2^{56}$，则上述算法成功的概率大约为 $0.8mt/N$。建议取 $m\approx t\approx N^{\frac{1}{3}}$并构造大约 $N^{\frac{1}{3}}$个表，不同的表使用不同的约化函数 R。如果这样做了，攻击的存储量为 $O(N^{\frac{2}{3}})$，搜索时间为 $O(N^{\frac{2}{3}})$，预计算时间为 $O(N)$。关于时间存储折中攻击的详细分析及进一步讨论参见文献[212]和[213]。