# 信息安全前沿课堂讨论

陆皓喆 2211044 网络空间安全学院

2023 年 12 月 19 日

## 1 密码攻击方法如何分类？如何衡量密码算法的安全性？

一般来说,密码攻击方法有以下这么几种——蛮力攻击(Brute-Force Attack)、唯密文攻击(Ciphertext-only attack)、选择明文攻击（Chosen plaintext attack）、选择密文攻击（Chosen plaintext attack）、已知明文攻击（Known plaintext attack）、密钥和算法攻击（Key and algorithm attack）、中间相遇(meet-in-the-middle)、中间人进攻（Man-in-the-Middle Attack)、重放攻击（Replay Attack)、功率分析攻击（Power Analysis Attack)、定时攻击（Timing Attack）等。

下面我简单的介绍一下这些攻击方法。

**1. 蛮力攻击**

蛮力攻击是密码攻击中最简单的。为了执行这些操作，攻击者只需通过猜测密钥并检查解密是否有效来尝试解密消息。如果有足够的时间和计算资源，暴力攻击将奏效，因为攻击者一定会找到正确的密钥。

现代密码通过使用一个足够长的密钥来防止暴力攻击，使得猜测变得不可能。高级加密标准（AES）具有最长的可用密钥长度-256 位。AES 密钥有个可能值。今天没有一台计算机能够在合理的时间内搜索到这样的密钥。

**2. 唯密文攻击**

唯密文攻击，在此攻击向量中，攻击者获得对密文集合的访问权限。尽管攻击者无法访问明文，但他们可以成功地从集合中确定密文。通过这种攻击技术，攻击者也可以偶尔确定密钥。

**3. 选择明文攻击**

选择明文攻击，在该攻击模型中，攻击者根据明文数据来获取密文。它简化了攻击者解析加密密钥的任务。这类攻击的一个众所周知的例子是对块密码进行的差分密码分析。

**4. 选择密文攻击**

选择密文攻击，在这个攻击模型中，攻击者分析与明文对应的选定密文。攻击者试图获取密钥或系统的详细信息。通过分析选定的密文并将其与明文关联，攻击者试图猜测密钥。旧版本的 RSA 加密容易受到此攻击。

**5. 已知明文攻击**

已知明文攻击，在这种攻击技术中，攻击者使用信息收集技术发现或知道密文的某些部分的明文。分组密码中的线性密码分析就是一个这样的例子。

6. **密钥和算法攻击**

密钥和算法攻击，这里，攻击者试图通过分析加密算法来恢复用于加密或解密数据的密钥。

7. **中间相遇**

中间相遇，中间相遇攻击使用两种已知的资产——明文块和相关的密文块——来破译最初用于促进加密的密钥。该攻击涉及从加密链的任一端向中间工作，而不是尝试从加密过程的一端向另一端进行暴力置换。本质上，中间相遇攻击涉及将加密过程分解为更简单、独立的步骤，而不是一条长而复杂的链。

中间相遇攻击通常用于解码多重数据加密标准（DES）技术。例如，双 DES 使用两个加密密钥将其明文输入转换为密文输出。这种加密方法使用它的两个唯一密钥来执行两个加密阶段。在这种情况下，中间相遇攻击的目标是使用中间值——加密阶段之间的值——来求解所有使用的加密密钥；如果没有一段明文和相应的密文，就无法进行攻击。这意味着攻击者必须有能力存储所有可能的中间密文值，这些值来自明文的暴力加密和密文的解密。虽然繁琐，但这并非不可能或不现实。事实上，中间相遇攻击的有效性导致 DES 技术变得不那么流行。虽然双 DES 不常用，但在某些情况下仍使用三 DES。然而，三重 DES 和双 DES 一样，可以由攻击者使用中间相遇攻击进行暴力强制。

中间相遇是一种被动攻击，这意味着尽管入侵者可以访问消息，但在大多数情况下，他们无法更改或发送自己的消息。根据黑客试图破译的加密方法，这种被动攻击可以在很长一段时间内进行。同样，这种攻击对于普通黑客来说并不实用，更可能用于公司或国际间谍活动或类似的环境，这些环境可以容纳执行攻击所需的存储空间。

**8. 中间人进攻**

中间人（MitM）攻击假定攻击者可以劫持双方之间正在进行的通信。攻击者可以拦截通过被劫持信道发送的所有消息（甚至是加密消息）。在成功的 MitM 攻击中，攻击者可以在消息传递给预期收件人之前解密、读取甚至修改消息。

为了阻止 MitM 攻击，攻击者需要让双方相信他们是对话的一部分。甲方应认为他们是乙方，反之亦然。只要这个诡计奏效，MitM 攻击就不会被发现。

**9. 重放攻击**

重放攻击是攻击者重放合法用户和服务器之间的有效会话。与将自己注入正在进行的通信不同，威胁行为体以服务器为目标劫持机密（甚至加密）数据。一旦完成，攻击者可以欺骗甲方他们是乙方，反之亦然。

使用密码，如随机数来识别唯一会话，可以防止重播攻击。每条消息都应该有一个唯一的号码，因此将甲方的消息重新发送给乙方将不起作用。

**10. 功率分析攻击**

计算机使用的功率和持续时间因其操作而异。密码算法使用多少功率可能会揭示它们正在处理什么数据。这使得攻击者能够猜测使用了何种加密，从而使猜测变得更快、更容易。

**11. 定时攻击**

定时攻击利用了这样一个事实，即不同的算法需要不同的时间来运行，这取决于加密的明文或使用的密钥。例如，确定用户登录安全系统时检查密码所需的时间。

**可以看出密码攻击方法具有多样性，且随着科技的发展，密码攻击方法越来越高级了。**

那么如何衡量密码算法的安全性呢?

根据被破译的难易程度，不同的密码算法具有不同的安全等级。如果破译算法的代价大于加密数据的价值，那么一般不会有人想去破译它，即，你可能是“安全的”。如果破译算法所需的时间比加密数据保密的时间更长，那么你可能也是“安全的”。如果用单密钥加密的数据量比破译算法需要的数据量少得多，那么你也可能是“安全的”。

在这里说“可能”，是因为在密码分析中总有新的突破。另一方面，随着时间的推移，大多数数据的价值会越来越小。

我们可以采用不同的方式来衡量攻击方法的复杂性：

1）数据复杂性（data complexity）。用于攻击输入所需要的数据量。

2）处理复杂性（processing complexity）。完成攻击所需要的时间，也经常称作工作因素（work factor）。

3）存储需求（storage requirement）。进行攻击所需要的存储量。

作为一个法则，攻击的复杂性取这三个因数的最小值。有些攻击包括这三种复杂性的折中：存储需求越大，攻击可能越快。

复杂性用数量级来表示。如果算法的处理复杂性是 2 的 128 次方，那么破译这个算法也需要 2 的 128 次方次运算（这些运算可能非常复杂和耗时）。假设我们拥有足够的计算速度去完成每秒 100 万次的运算，并且用 100 万个并行处理器完成这个任务，那么仍然需要花费 10 的 19 次方年以上才能找到密钥。（而这是宇宙年龄的 10 亿倍）。

当攻击的复杂性是常数时（除非一些密码分析者发现更好的密码分析攻击），就只取决于计算能力了。在过去的半个世纪中，计算能力已经得到了显著的提高，并且现在这种趋势还在发展。许多的密码分析攻击用并行处理的机制进行计算非常理想，一个任务可以分成亿万个子任务，并且处理之间不需要相互作用。一种算法在现有技术条件下不可破译就草率的宣称是安全的，是很冒险的。从中我们可以得出，一个好的密码系统应设计成能抵御未来多年后的计算能力的发展。

**由以上的资料，我们可以看出，一个密码的安全性衡量首先是要看该密码被破译的所需要的计算复杂度，然后还要看该密码的破译难度需要防御住未来的计算能力发展而导致的破译能力的提升。**