基于 Python 语言的 3DES 算法优化策略

□ 陈倩 陈建敏

摘要：随着计算机信息技术和互联网的快速发展，人类社会已经进入信息化时代。人们在使用网络的时候越来越重视信息的安全性，这也促进了加密技术的发展，同时应用加密技术能够有效提升网络信息的安全性。3DES加密算法在保障信息安全领域具有较好的效果。本文对3DES加密算法进行了相关概述，分析了3DES加密算法的相关原理，并基于Python语言探讨了3DES算法的优化策略。

关键词：Python语言；加密算法；3DES算法；优化策略

当今社会已经进入信息化时代，互联网已经普遍应用到人们的生活当中，并对人们发生了极大的改变，同时人们对信息在网络上传播的安全性也越来越重视。加密技术的发展有效提升了网络信息的安全性。随着加密技术的发展，3DES 加密算法具有较高的安全性，在保证信息安全领域发挥着重要的作用，并且已经广泛应用于各种网络设备的信息数据传输业务中。在 Python 语言的基础下，对 3DES 加密算法进行研究具有重要的意义，通过对 3DES 算法进行相应的分析，提出了相应的优化策略。

一、3DES 加密算法相关概述

（一）3DES 加密算法介绍

3DES 形成的基础来源于 DES 算法，DES 加密算法的相关标准形成较早，能够准确地对密码进行分析，有效防止各类黑客的攻击。DES 算法能够对 64 位的明文数据进行分组操作，通过对明文进行初始化的置换，可以将明文分为左半部分和右半部分两个不同的部分，而且还可以进行 16 轮完全相同的计算，最后经过进行一个末置换就可以得到 64 位密文。

在这个过程中的每一轮的运算都包含有扩展置换、S 盒代换、P 盒置换以及两次异或运算，除此之外，在每一轮中还存在一个轮密钥。而 3DES 是在 DES 基础上形成的，其与 DES 算法相比较具有更高的安全性，3DES 加密算法还是 DES 想 AES 过渡的加密算法，可以说是 DES 的一个更加安全的变形，其以 DES 为基础模块，通过进行组合分组方法来设计出分组加密算法。

（二）3DES 算法的原理分析

3DES 算法以 DES 为基础模块，能够进行加密和解密的过程，其具体的实现过程如下：对于 DES 算法，设 Ek(.) 和 Dk(.) 表示该算法的加密和解密过程，K 表示的是 DES 算法使用的密钥，明文用 P 表示，而密文用 C 表示，那么 3DES 算法的加密解密过程就可以表示为：

加密过程：$C = E_{k3}(D_{k2}(E_{k1}(P)))$

解密过程：$P = D_{k1}((E_{k2}(D_{k3}(C))))$

其中的 K1、K2、K3 决定了 3DES 算法的安全性。如果这三个密钥都是互相不同的，其本质上就是在使用一个长度为 168 位的密钥进行加密，相关的实践证明，该算法多年来在应对强有力的攻击时是比较安全的。如果进行传输的数据信息对安全性的要求不太高，那么密钥 K1 可以和 K3 相同，这种情况下的密钥的有效长度就变为了 112 位。

二、基于 Python 语言的 3DES 算法优化

（一）对于算法密钥的处理

根据密码学的相关理论，DES 算法能够运用 64 位的密码技术，但是在这 64 位中有 8 位是用来进行校验的，剩下的 56 位有效的长度是可以正常使用的，所以在进行数据信息加密的过程中应用 DES 算法时，由于该算法位数的有效长度较低，密码的安全性也相对不强。在 DES 算法实际的应用过程中，可以生成 DES 算法密钥的原始密钥的位数一般是不确定的，这就需要对原始密钥进行预处理，采用的是 SHA-1 算法来生成原始密钥的摘要，同时也利用到 Python 语言中的函数，并且应用该函数能够生成 40 个字节的字符串。

经过这一环节的处理后，不管多长的密钥都会变成具有固定长度的字符串，用低 7 位的 ASCII 值的二进制表示字符串中的每一个字符，会得到长度为 280 的二进制串。对 280 的二进制串进行相应的处理，会得到 56 长度的二进制串，然后再对二进制串进行相应的置换操作，就可以得到最终的预处理密钥。在对密钥进行置换的过程中，通常需要按照相应的时间顺序，对密钥置换的安全性也要进行分析。密钥在经过预处理后，破译者没有得到加解密模块或者不知道密钥的处理方法，就很难得到真正有效的密钥，也无法对密文进行相应的破解。

（二）对于 S 盒的处理

Python 语言具有简洁和清晰的语法结构，基于 Python 语言的 3DES 算法能够提高解密的效率，对于加密的 S 盒中的相关数据，可以进行统一的转换，转换为二进制的字符，这样在算法运算的过程中，就可以直接使用二进制的数据来完成任务，在进行解密的过程中也不需要去进行字符的转化，从而能够节省大量的时间，并在一定程度上提高了加密的运行速度。

（三）3DES 算法代码效率的优化

根据有关内存大小，可以将被加密的数据信息划分成大小不同的信息，利用 Python 语言中的 profile 工具，可以简化 pyDES 模块中的 3DES 算法，这不仅能够提高运算的效率，还能够分析完善后的系统的相关功能。通过使用 Python 语言中的 profile 工具对 8192 长度的字符串进行分析，与改进之前的相同字符串的测试输出进行比较，可以发现在两个函数的调用次数没有改变的情况下，运行的时间发生了较大的改变，测试后的字符串具有非常明显的改变，Append 操作的次数也明显减少，运算速度得到了提升。

三、总结

综上所述，随着人们对网络中信息传输的安全性的重视，相应的加密算法在多个领域得到了广泛的应用。利用 Python 语言中的函数以及相关工具，通过对 3DES 算法进行相应的简化，能够有效提升 3DES 算法的安全性和处理效率。通过对 3DES 算法进行相应的优化，能够使 3DES 算法得到更好的应用。

参考文献

[1] 高润博．基于 Python 语言的 3DES 算法完善 [J]．电子技术与软件工程，2015(19)：228.

[2] 李爱宁，唐勇，孙晓辉，刘昕彤．基于 Python 语言的 3DES 算法优化 [J]．计算机系统应用，2011,20(08)：184-187+173.

（作者单位：黄山职业技术学院）

作者简介：陈倩（1982～），本科，讲师，研究方向为计算机应用，程序设计。

基金项目：2018 年安徽高校自然科学研究重点项目

项目名称：基于 Python 的智慧旅游大数据智能分析平台的研发（项目编号：KJ2018A0953）