基于 FBR 特征的密码算法识别

向广利[1] 蒋 欣[1] 张于洁[1] 杨立新[2]

[1](武汉理工大学计算机科学与技术学院 湖北 武汉 430070)

[2](湖北省科技信息研究院 湖北 武汉 430071)

摘 要 针对现有的密码算法识别存在密文特征提取不足和识别准确率低等问题,提出一种 FBR 密文特征提取方法。该方法结合随机性测试中的频率(Frequency)、块内频率(Block Frequency)和游程(Run)三种方法,定义出密文的码元次数统计值、游程次数统计值和块内次数统计值,基于三种统计值构造出 FBR 特征。实验使用支持向量机对三种混合数据集分别进行密文二分类和多分类实验。实验结果表明,该方法所提取的 FBR 密文特征对比已有表现良好的密文特征,其平均识别准确率得到较高的提升,充分证明了该方法的有效性。

关键词 密码算法识别 特征提取 FBR 特征 支持向量机

中图分类号 TP39 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.11.049

RECOGNITION OF CRYPTOGRAPHIC ALGORITHMS BASED ON FBR FEATURE

Xiang Guangli[1] Jiang Xin[1] Zhang Yujie[1] Yang Lixin[2]

[1](*School of Computer Science and Technology*, *Wuhan University of Technology*, *Wuhan 430070*, *Hubei*, *China*)

[2](*Hubei Academy of Scientific and Technical Information*, *Wuhan 430071*, *Hubei*, *China*)

Abstract Aimed at the problems of insufficient ciphertext feature extraction and low recognition accuracy in existing cryptographic algorithm recognition, a FBR ciphertext feature extraction method is proposed. This method combined the three methods of frequency, block frequency and run in the randomness test to define the statistical value of the number of symbols in the ciphertext, the statistical value of the number of runs, and the statistical value of the number of times within the block. FBR features were constructed based on three statistical values. The experiment used support vector machines to perform ciphertext two-classification and multi-classification experiments on the three mixed data sets. The experimental results show that the FBR ciphertext features extracted by this method are compared with the ciphertext features that have performed well, and the average recognition accuracy is improved, which fully proves the effectiveness of the proposed method.

Keywords Cryptographic algorithm recognition Feature extraction FBR feature SVM

0 引 言

新型密码算法标准征集活动的兴起显著地促进密码算法的发展[1-2]。评判密码算法安全性的强弱,就是它在面对密码攻击时是否具有抵抗能力,可以将这种能力作为评价密码算法安全性的一种检验标准,为建设安全强度更高的密码算法提供坚实的基础和可靠的保障[3]。

大多数的密码分析通常都是假设在某一种密码下进行攻击,然后对密文数据进行分析并获取其中潜在有价值的信息,例如加密明文所用的密钥或者是将密文通过某种技术手段,将其还原成明文形式等。但在真实场景下,研究者们并不知道某些密文是用什么密码算法加密的。因此,针对密码算法识别研究是一项非常有意义的工作。理论上如果明文数据被不同类别的密码算法所加密后生成的密文,其内部信息能够做到完全随机标准。但在实际情况下,因密码算法设计

收稿日期:2021-04-23。湖北省技术创新专项重大项目(2019ABA102)。**向广利**,教授,主研领域:信息安全。**蒋欣**,硕士生。**张于洁**,硕士生。**杨立新**,博士。

的不足,所生成的密文数据之间依然存在某种细微的区别,这些细微的区别可以作为特征来识别密文所使用的密码算法。

目前国内外对这方面的研究主要是通过统计学和机器学习这两种方法去识别区分密文所用的密码算法[4]。基于统计学方法主要是统计并计算每种密码算法的密文中不同的字符出现的次数或者是码元 0、1 比特出现的次数等,将统计出来的值与现有的密码算法度量值进行比对,如果符合某一些度量值,则可以确定该密文是由哪种密码算法所加密的。但是这种方法存在较大的误差,同时也很难达到完全随机性[5]。基于机器学习方法主要是通过利用现有的密文特征提取算法提取密文中的某些特殊信息,而这些信息可以作为区分密码算法之间的一种特征,然后将带标签的数据输入到传统的机器学习模型的常规训练流程进行模型训练和测试分类。由于机器学习方法具有更强的可实践性,逐渐成为广大研究者所首选的方法。

在机器学习中,特征的提取与选择是一项非常重要的任务,直接影响着密码算法的识别准确率。由于密文数据自身的特殊性,导致它无法提取出较多的有效特征,因此密文特征提取也就成为密码算法识别研究中的关键环节。

针对现有的多数研究是直接使用随机性测试的返回值作为密文特征或者是使用熵特征进行密码算法的识别。但是由于这些特征之间具有高度的相关性和检测的片面性[6],导致出现识别准确率低等问题。提出一种 FBR 密文特征提取方法,该方法显著提高了密码算法的平均识别准确率。

1　相关研究

对于密码算法识别的研究,国内外研究学者们做了如下工作。

1.1　密文特征提取研究

2006 年,Dileep 等[7]通过借鉴使用固定长度和非固定长度的文档向量来提取密文特征,并用机器学习中的 SVM 和 KNN 两种分类模型来识别密文所使用的密码算法,最后成功识别出 AES、DES、3DES 等 5 种密码算法,但识别率较低,只有 50% 左右。

2008 年,Nagireddy 等[8]通过获取文本、图片和音频文件的直方图特征,使用 SVM 模型成功对 DES、AES、Blowfish、TDES 和 RC5 等五种密码算法进行区分,但实验结果显示该特征的分类效果较差。

2015 年,吴杨等[5]结合 NIST 公布的 16 种随机性测试标准,将测试所返回的随机性值作为密文随机性度量的特征,并计算不同特征之间的相似性,以此为根据进行密码算法识别实验。实验采用 K-means 模型对 AES、Camellia 等五种密码算法进行识别实验。最后识别出上述 5 种密码算法,但识别准确率不高,在 85% 左右。

2018 年,李洪超[9]结合熵的特性和密文的分组方式提出分组字节熵、分组比特熵和整体字节熵三种特征,这三种特征分别计算密文的每个分组下的字节熵、比特熵和整体的字节熵等。通过实验测试发现这些特征能反映出密文的分布特点,进而使用 SVM、随机森林和决策树等分类模型进行识别测试。实验结果发现所提特征具有较好的分类的效果,其中整体字节熵的分类效果最佳。

2019 年,赵志诚等[10]借鉴 NIST 随机性测试标准,改变测试参数以及密文分块数等方式,将密文分别按照固定分组长度进行分块,通过计算固定比特位的熵和不同组合形式的比特串所出现的概率,构造出多种不同维数的密文特征。随后使用随机森林模型对 AES、DES、3DES 和 IDEA 等六种密码算法进行两两组合实验,并以较高识别率成功区分。

1.2　分类模型选择研究

2007 年,Chandra 等[11]选择级联相关神经网络对 RC6 和 SEAL 两种密码算法进行了二分类的尝试,得到了较令人满意的 85% 准确率。并指出随着数据集的复杂度增加的情况下,测试结果仍能保持良好的一致性。但是该研究只是对密文的二分类进行研究,并未对密文的多分类进行研究。

2010 年,Sharif 等[12]使用 Bayes、SVM 等 8 种分类识别模型对 IDEA、AES 等 4 种密码算法进行区分实验。实验结果显示随机森林模型识别表现最好,而基于实例的学习方法表现较差。

2013 年,Souza 等[13]参考本文检索中的词组向量模型的思想,选择神经网络作为识别模型,对 MARS 等 3 种密码算法开展识别实验。但是由于这种方案的计算复杂度高、模型训练耗时长,所以不适合用于大量的密文数据识别。

2018 年,黄良韬等[3]通过将现有的密码算法大致分成四种密码体制,采用分层的识别方式,使用随机森林模型,先识别出密码算法的类别,然后再识别这个类别中具体所使用的密码算法。实验结果显示,该方案对比于现有的方案,识别准确率均得到较高的提升。

从上述的密文特征提取研究和分类模型的选择研究中可知,影响密码算法识别效果的根本原因在于密

文特征上。虽然不同的分类模型会给识别效果带来一定差异,但是这些差异不足影响整体的识别效果,研究的核心依然在于密文特征提取方面。

2 常见密文特征提取方法

常见的密文特征提取方法有随机性测试方法和熵特征测试方法,下面简要介绍这两种密文特征提取方法。

2.1 随机性测试方法

随机性方法主要用于检测密码算法输出密文序列的统计特性[14]。当使用 NIST sts 集成软件[15]对密文的随机性进行检验时,它对要识别密文的数量要求较高且并非所有的检测结果都可以作为高效的分类识别特征。NIST 公布的 16 种随机性方法如表 1 所示。

表 1 16 种随机性测试方法

方法名称	原理
频率	判断序列中 0 和 1 个数是否接近
块内频数	判断子块中 1 的个数是否接近 $m/2$
游程	判断不同长度的游程总数是否符合期望值
块内最长游程	最长 1 游程的长度是否符合期望值
二元矩阵秩	判断矩阵行和列之间的线性独立性
离散傅里叶变换	判断变换后的尖峰高度是否超过门限值
非重叠模块匹配	检测设置好的目标数据串发生的次数
重叠模块匹配	数据串的次数是否与非重叠模块匹配一致
Maurer 的通用统计	判断序列是否可以被无损压缩
Lempel-Ziv 压缩	判断待测序列能够被压缩到什么程度
线性复杂度	判断子序列的线性复杂度是否符合期望值
序列	判断重叠子序列每个模式的个数是否接近
近似熵	重叠子块的频数和随机序列是否一致
累加和	最大累计和与随机序列的最大偏移比较
随机游动	判断特征状态节点数是否与随机序列一致
随机游动状态频数	随机游动状态值与期望值之间的偏离程度

下面介绍 16 种随机性方法中对于密文分类有效的 5 种方法的原理。

(1) 块内最长 1 游程。该方法是先将密文序列按照固定长度大小进行分块,计算每个分块中长度达到最长的 1 游程的分布,然后再统计出随机序列中最长 1 游程的分布情况,最后通过判断二者分布情况是否近似相同,如果相同则通过测试。

(2) 重叠模板匹配。重叠模板匹配方法主要是将密文序列与提前设定的模板进行匹配,先计算二者匹配的次数,最后再按照相同的做法计算随机序列中的匹配次数。最后判断二者次数是否近似一致。如果一致则通过测试。反之未通过测试。它主要是通过不断向前移动大小为 1 比特的窗口进行匹配。无论是否匹配成功,都不会影响窗口继续移动。

(3) 线性复杂度。该方法主要是使用 LFSR 去检测密文序列的随机性,当 LFSR 的长度不合格,则认定这个密文序列随机性差,不能通过测试。LFSR 的长度代表着随机程度的高低。

(4) 通用统计。该方法主要是测试当密文序列在被压缩的条件下,密文自身携带的数据信息如果没有出现丢失,则可以认定这个序列已经达到随机的条件。

(5) 二进制矩阵秩。该方法主要是将密文序列 0、1 分成若干个子序列并转换成密文矩阵,然后测试每个子序列之间是否存在某种相关性,当这种相关性较大时,则可以认定这个密文序列随机性较差,无法通过测试,反之,可以通过测试。

利用 NIST sts 集成软件对密码算法所生成的密文进行随机性检测,当不同密码算法重叠的区域越稀疏则说明算法在该特征上表现出的统计特点不相似,也就越容易区分。

虽然随机性方法的返回值作为密文特征能对密码算法进行识别,但是由于随机性特征之间存在高度相关性和检测的片面性等问题,它无法有效区分出密文的特点,在对多种密码算法进行识别时,出现识别准确率差等问题。

2.2 熵特征测试方法

熵特征方法主要用于计算密文每个比特位上的熵值,该值可以反映出密文的随机性。在信息论中,熵是用来判断信息无规律性的计算方式。因此熵越大,表示无规律性因素越多。侧面可以反映密文的安全性能,是密文混沌程度的特征。

文献[9]提出几种熵特征计算方法,这里主要介绍分组间比特熵和分组间字节熵两种密文熵特征。其中分组间比特熵记为 F_56b、F_128b、F_256b 和 F_1024b,即分别将密文按照 56 bits、128 bits、256 bits 和 1 024 bits 分块,计算每一比特位上出现 0、1 的熵,依次得到 56 维、128 维、256 维和 1 024 维的熵特征。i 维特征可以表示成 $(f_1^i, f_2^i, \cdots, f_i^i)$,其中 $i = 56, 128, 256, 1\,024$。每一项计算公式如下:

$$f_j^i = -p_j^i \log p_j^i - (1 - p_j^i \log(1 - p_j^i)) \tag{1}$$

式中:$p_j^i = \dfrac{\sum_{j=1}^{\lfloor l_b / i \rfloor} b_j + 56 \cdot (j-1)}{\lfloor l_b / i \rfloor}$, $j = 1, 2, \cdots, i$。

分组字节熵可以记为 F_56cut7、F_128cut16 和 F_256cut32,计算所有块中某一固定字节的熵,依次可得到 7 维、16 维和 32 维的熵特征,i 维特征可以表示成$(f_1^i, f_2^i, \cdots, f_{256}^i)$,其中 $i=7,16,32$。具体计算公式如下:

$$f_j^i = -\sum_{k=1}^{256} p_{j,k}^{iX256} \log p_{j,k}^{iX256} \quad (2)$$

当每个块中第 j 个字节取值是 $k-1$ 的字节时所出现的概率可以用 $p_{j,k}^{iX256}$ 表示,$j=1,2,\cdots,i$。

熵特征虽然能够在一定程度上作为密文特征,但是它随着密文件大小的增加,特征提取所消耗的时间也会呈现几何级增长,因此只适用于小批量数据的密码算法识别。

3　FBR 密文特征提取方法

针对常见密文特征提取方法所表现出的问题,结合随机性方法中的频率、块内频率和游程三种测试方法,提出一种新的 FBR 密文特征提取方法。

3.1　方法基础

下面详细介绍游程、频率和块内频率三种方法的特点和具体的形式化算法过程。

(1) 频率方法。频率方法是计算码元 0 和码元 1 在整个密文序列中的百分比。先计算出密文序列中的码元 0 和码元 1 出现的数量,然后再计算出随机序列中,码元 0 和码元 1 出现的数量。通过对比二者间的数量,如果大致相等,则通过该测试,并认为密文序列是随机的。反之,未通过该测试。具体的过程如算法 1 所示。

算法 1　频率算法

输入:密文序列(0、1 比特串),密文序列长度 l bits 密文分块数 K。

输出:密文序列的特征值。

1. 将密文序列分成 K 块,K 取值为 64、128、256 等,密文 $B=(B_1,B_2,\cdots,B_k)$,K 个块大小均为 $l_k=l/K$

2. 将密文比特串 $B_i=(b_{i,1},b_{i,2},\cdots,b_{i,l_k})$ 中的 0 和 1 分别用 1 和 -1 来表示,随后计算 $S_n=X_1+X_2+\cdots+X_n$,$X_j=2b_{i,j}-1$,$n=l_k$

3. 计算测试统计量 $S_i=|S_n|/\sqrt{n}$

4. 计算:$f_i^F=p_i=erfc(s_i/\sqrt{2})$,其中 $erfc$ 为余补误差函数,$erfc(\sigma)=2/\sqrt{\pi}\int_{\sigma}^{\infty} e^{-u^2}du$

5. 将 K 个分块的 K 个 f_i^F 值作为密文特征,即作为 K 维特征$(f_1^F,f_2^F,\cdots,f_K^F)$返回

(2) 游程方法。游程是指连续相同的二进制数的序列,如游程可以是“111111”或者是“00000”。它的原理是计算出密文序列中不同长度的 0 游程数量和不同长度的 1 游程数量,然后再计算随机序列中不同长度的 0 游程数量和 1 游程数量,最后通过对比二者之间数量是否近似相同。如果相同则通过测试。反之,未通过测试。具体的过程如算法 2 所示。

算法 2　游程算法

输入:密文序列(0、1 比特串),密文序列长度 l bits 密文分块数 K。

输出:密文序列的特征值。

1. 将密文序列分成 K 块,K 取值为 64、128、256 等,密文 $B=(B_1,B_2,\cdots,B_k)$,K 个块大小均为 $l_k=l/K$;然后对 K 个小块密文 B_i 进行测试并返回其值

2. 对密文比特串 $B_i=(b_{i,1},b_{i,2},\cdots,b_{i,l_K})$,其中 $i=0,1,2,\cdots,K$,计算 $V_{i,n}=\sum_{j=1}^{n-1} r(j)+1$,其中 $n=l_k$ 表示密文块 B_i 的比特数量。如果 $b_{i,j}=b_{i,j+1}$,则 $r(j)=0$,否则 $r(j)=1$

3. 随机测试值 p_i 如下:

$$p_i = erfc\left(\frac{|V_n - 2n\pi(1-\pi)|}{2\sqrt{2}\pi(1-\pi)}\right)$$

4. 将 K 个分块的 K 个 p_i 值作为密文特征,即作为 K 维特征$(p_1^{BF},p_2^{BF},\cdots,p_K^{BF})$返回

(3) 块内频率方法。块内频率方法主要原理是计算以 M 作为不同长度的密文分块中码元 1 的比例。通过判断不同分块内 1 出现的频率是否和随机序列中子块中的码元 1 的频率一致。如果一致,则通过测试。反之,未通过测试。下面给出具体的过程如算法 3 所示。

算法 3　块内频率算法

输入:密文序列(0、1 比特串),密文序列长度 l bits 密文分块数 K。

输出:密文序列的特征值。

1. 将密文序列分成 K 块,密文分块序列 $B=(B_1,B_2,\cdots,B_k)$,其中每小块密文大小均为 $l_k=l/K$,K 取值为 64、128、256 等

2. 在测试中设置 Block Size M 设为 512 bits。将第 i 个密文块 $B_i=(b_{i,1},b_{i,2},\cdots,b_{i,l_K})$,其中 $i=0,1,2,\cdots,K$,划分成 $N=\lfloor l_k/M \rfloor$ 个不叠加的比特串,对于划分剩余的比特串进行丢弃

3. 对每个长度为 M bits 的密文比特串计算:

$$x_j = \frac{\sum_{j=1}^{M} b_{(i-1)M+j}}{M} (i=0,1,\cdots,N)$$

4. 计算 χ^2 统计量,具体如下:

$$\chi^2 = 4M\sum_{i=1}^{N}\left(x_i - \frac{1}{2}\right)^2$$

5. 计算随机测试值 p_i:

$$p_i = \text{igamc}(N/2, \chi^2/2)$$

6. 将 K 个分块所计算的值 p_i,即作为 K 维特征$(p_1^{BF},p_2^{BF},\cdots,p_K^{BF})$返回

3.2 方法设计

基于统计学的角度,明文数据被分组密码算法加密后所得到的密文中比特 0 和比特 1 出现的次数、连续不同长度的比特 0 和比特 1 出现的次数、在不同分块长度内比特 0 和比特 1 出现的次数都存在不同程度的差异性。通过统计上述三种情况下的比特 0 和比特 1 出现的次数,重新定义并构造码元次数统计值、游程次数统计值和块内次数统计值三种方法。基于上述方法的返回值,构造出一种 FBR 密文特征提取算法。

3.2.1 码元次数统计值

码元在通信领域中常用来表示时间间隔内的信号,在这里码元表示所传输的比特 0 或 1。码元次数统计是先得到由不同分组加密算法所加密的密文,然后分别计算出每个密文中比特 0 和比特 1 出现的次数,通过这些出现的次数可以得到它的分布情况。从理论上讲,分组密码所加密得到的密文中,比特 0 和比特 1 有相同的出现次数。为此定义长度为 n_i 的密文 $C_i = c_1, c_2, \cdots, c_{n_i}$,在获取 C_i 的码元次数统计值时,本文先对 $x_j = 2c_j - 1$ 进行变换,重新构造一种新的密文序列 $X_i = x_1, x_2, \cdots, x_{n_i}$,然后统计出 X_i 的和 $S_i = x_1 + x_2 + \cdots + x_{n_i}$。经过上述变换过程,重新定义 C_i 的码元次数统计值 f_i:

$$f_i = \frac{S_i}{n_i} \tag{3}$$

就 C_i 的码元次数统计值,当所加密得到的密文中,全部都是比特 0 或比特 1 的时候,对应的 f_i 的可能取值只有 -1 或者 1 两种。所以,$f_i \in [-1, 1]$。针对密文的码元次数统计值,它的取值通常是以 0 为中心点,取值走势符合正态分布的特性。形式化表示为算法 4。

算法 4 码元次数统计值算法

输入:密文序列 C_i(0,1 比特串),密文序列长度 l_i。

输出:密文序列 C_i 的码元次数统计值 f_i。

1. 对于密文序列 C_i,先通过 $x_j = 2c_j - 1$ 进行变换,得到新密文序列 $X_i = x_1, x_2, \cdots, x_{l_i}$
2. 对 X_i 的每一项进行求和 $S_i = x_1 + x_2 + \cdots + x_{l_i}$
3. 计算 $f_i = \frac{S_i}{l_i}$

注意:这里进行变换的目的在于借鉴 NIST 的频率测试中将密文比特串中的 0、1 用 1 和 -1 表示,主要是便于后面构造新序列,方便统计计算。

3.2.2 游程次数统计值

游程其实就是比特串(序列)中的子串,它们通常是由连续(两个及其以上)的 0 或 1 所组成,被称为 0 游程和 1 游程。定义密文中的 0 游程或者 1 游程是为了更好地统计密文中的连续比特 0 和连续比特 1 出现的分布情况。本文重新定义密文的游程次数统计值 r_i:

$$r_i = \frac{1}{\alpha}\sum_{j=1}^{\alpha}(O_{ij} - Z_{ij})^2 \tag{4}$$

式中:Z_{ij} 表示当在密文 C_i 中,长度为 j 的序列中 0 游程出现的次数;α 表示最长的游程的长度;O_{ij} 表示当在密文 C_i 中,长度为 j 的序列中 1 游程出现的次数。

从理论上讲,0 游程出现的次数应当和 1 游程出现的次数相同,理论游程 O_{ij} 和 Z_{ij} 与序列长度 n_i 保持一种对应关系:

$$\frac{n_i}{8} = O_{i1} = 2O_{i2} = 2^2 O_{i3} = \cdots = 2^{j-1}O_{ij} \tag{5}$$

从以上关系可推出:

$$\begin{cases} O_{i1} = \dfrac{n_i}{8} \\ O_{i2} = \dfrac{n_i}{16} \\ O_{ij_{\max}} = \dfrac{n_i}{2^{j_{\max}+2}} \end{cases} \tag{6}$$

当序列中的 0 游程或者是 1 游程长度达到最大值时,可以认定这个游程出现的次数接近为 1,所以,0 游程或者是 1 游程最长可以表示为 $\alpha = \log_2 n_i - 2$。因分组密码的分组长度固定且有限,并不能达到真正上的随机序列,所以这个 α 通常是小于真正意义上的游程长度。形式化表示为算法 5。

算法 5 游程次数统计值算法

输入:密文序列 C_i(0,1 比特串),密文序列长度 l_i。

输出:密文序列 C_i 的码元次数统计值 r_i。

1. 在密文序列 C_i 中,先计算 0 游程出现的次数 Z_{ij} 和 1 游程出现的次数 O_{ij},计算的序列区间长度的为 j
2. 计算 $r_i = \frac{1}{\alpha}\sum_{j=1}^{\alpha}(O_{ij} - Z_{ij})^2$,其中 α 表示最长的游程的长度

3.2.3 块内次数统计值

块内次数统计的意义在于当密文按照不同的长度进行分块时,假设这个长度是固定值 K,计算出所有分块中比特 0 和比特 1 出现的次数。这里本文使用非叠加的方式将密文 C_i 分成 $N_i = \lfloor n_i/K \rfloor$ 个块,并重新定义块内次数统计值 b_i:

$$b_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\left(\pi_j - \frac{1}{2}\right)^2 \tag{7}$$

式中:π_j 表示比特 1 出现的概率(第 j 个分块内)。所以,由式(7)所知:当每个密文分块中的 π_j 等于 1/2 时,b_i 将达到最小值 0。反之,当 π_j 等于 1 时,b_i 将达到最大值 1/4。从理论上来讲,固定分块长度时,每个分块中比特 0 和比特 1 出现的概率尽可能地接近 1/2。

因此在理论上,块内次数统计量也符合正态分布。形式化表示为算法 6。

算法 6　块内次数统计值算法

输入:密文序列 C_i(0,1 比特串),密文序列长度 l_i,密文分块长度 K。

输出:块内次数统计值 b_i。

1. 将密文序列 C_i,按照 K 作为固定分块,K 通常取值为 64、128、256 等,划分成 $N_i=\lfloor l_i/K \rfloor$个非重叠的块

2. 计算 $b_i=\frac{1}{N_i}\sum_{j=1}^{N_i}\left(\pi_j-\frac{1}{2}\right)^2$,其中 π_j表示比特 1 出现的概率

3.3　方法实现

在一定量密文样本数的前提下,分别计算并统计每个由密码算法所加密的密文中,每条密文的三种统计值的不同值,将不同的取值存在一个集合中,并计算得到的同一种统计值中的极大值和极小值。用该值表示每个密码算法的差异特性,以此来区分不同的密码算法。针对同一组的任意一个密码算法 A,经过加密所产生的密文用 G 来表示,密文 G 的分组数假定是 g,分组后的每组密文数量为 m。则算法 A 的任意一组密文可以用 G_i^{A}来表示:$G_i^{\mathrm{A}}=(G_{i1}^{\mathrm{A}},G_{i2}^{\mathrm{A}},\cdots,G_{im}^{\mathrm{A}})$,其中$1\leqslant i\leqslant g$,$G_{im}^{\mathrm{A}}$表示算法 A 中第 i 组下的第 m 条密文。同组中的其他密码算法均可由上述式子进行表示。FBR 密文特征提取方法的具体过程可形式化如算法 7 所示。

算法 7　FBR 密文特征提取算法

输入:密文序列 C_i(0,1 比特串)长度 l_i,密文分块数 K,各自的密文分组数 g,每组密文样本量 m。

输出:密文序列 FBR 特征值。

1. 使用上述所构造的三种统计值算法计算获取每个分组密码算法下的密文分组 G_i的统计值 f、r 和 b

2. 将该算法的每个统计值封装成集合 $F_i=(f_{i1},f_{i2},\cdots,f_{im})$,$R_i=(r_{i1},r_{i2},\cdots,r_{im})$,$B_i=(b_{i1},b_{i2},\cdots,b_{im})$。其中$f_{im}$、$r_{im}$及 b_{im}分别为密文的第 i 组密文中第 m 条密文对应的统计值

3. 重复步骤 1、步骤 2,分别计算并统计所有 g 组的密文的统计值集合,可表示为 $F_U=F_1\cup F_2\cup\cdots\cup F_g$、$R_U=R_1\cup R_2\cup\cdots\cup R_g$和 $B_U=B_1\cup B_2\cup\cdots\cup B_g$

4. 在获得每个分组加密算法的上述三个统计量集合 F_U、R_U和 B_U之后,将每个分组加密算法所对应的同一种统计量集合进行统计分析并获得其中具有较大差异的值作为极值,如下:$E_F=(e_{F\to1},e_{F\to2},\cdots,e_{F\to\omega_1})$,$E_R=(e_{R\to1},e_{R\to2},\cdots,e_{R\to\omega_2})$,$E_B=(e_{B\to1},e_{B\to2},\cdots,e_{B\to\omega_3})$。其中:$E_F$、$E_R$及 E_B是上述定义的三种方法所返回的值 f、r 及 b 的极值均值集合;ω_1、ω_2及 ω_3是上述定义的三种方法 f、r 及 b 的分组个数;$e_{F\to i}$、$e_{R\to i}$及 $e_{B\to i}$表示三个所返回的集合 E_F、E_R及 E_B中的第 i 个分组的码元次数统计值、游程次数统计值和块内次数统计值三种极值的均值

5. 返回 g 组极值均值(极大和极小值),形成 g 维特征 $F_{BR}=(E_F,E_B,E_R)$,将其作为密文特征值返回

3.4　特征评估

从 Caltech256 图片库[16]中任意抽取不同大小的图片进行拼接,然后使用 Java 程序按照固定大小将图片划分成 1 000 份。在同一种工作模式 ECB 下,选择 OpenSSL 密码库中的 AES、3DES 和 Blowfish 三种密码算法分别对这 1 000 份图片进行加密,得到 3 000 份密文数据用于算法评估。相关的统计值分析均使用 VC 6.0 中完成。

在评估过程中,通过不断反复尝试,设定密文的分组数量 $g=30$,这样才能确保每组密文都具有足够的样本量进行实验。在确定了密文的样本数量后,通过所构造的三种统计值算法分别计算出每个分组密文中的每条密文的统计值f_i、r_i和 b_i值的极值均值。结果如表 2 所示。

表 2　三种统计值极值的均值

极值的均值	AES	3DES	Blowfish
$\tilde{f}_{\min}$	−0.210 6	−0.265 4	−0.185 7
$\tilde{f}_{\max}$	0.183 4	0.261 2	0.185 4
$\tilde{r}_{\min}$	0.238 7	0.238 7	0.162 3
$\tilde{r}_{\max}$	43.168 0	40.935 6	40.345 7
$\tilde{b}_{\min}$	0.002 0	0.002 8	0.000 8
$\tilde{b}_{\max}$	0.034 2	0.048 5	0.003 6

可以看出,AES、3DES 和 Blowfish 的 f 值均符合 3.2.1 节中的理论分析取值范围[−1,1],这也进一步验证了本文所设计的统计检测方法的正确性。从 $\tilde{f}_{\min}$和 $\tilde{f}_{\max}$的极值的均值中看,3DES 与 AES、Blowfish 具有较大的差别,通过 f 的极值的均值可以从密文中区分出 3DES。从 $\tilde{r}_{\min}$和 $\tilde{r}_{\max}$的极值的均值中看,AES 与 3DES、Blowfish 具有较大的差别,可以作为区分的依据。而 Blowfish 的 $\tilde{b}_{\min}$值和其他密文之间具有较大的差异。

通过上述分析发现,本文所定义的三种统计值 f、r 和 b 极值的均值,即 FBR 特征可以作为识别密码算法的密文特征。

4　实验与分析

4.1　环境配置

本文实验主机配置为 Windows 10 操作系统,Intel Core i5-9300H 处理器,2.80 GHz 频率,16 GB 的内存。实验所用的密码算法均来源于 OpenSSL 加密库,分

类模型选择支持向量机模型,通过调用软件函数包 Scikit-learn 实现。

4.2 数据集

本文根据明文数据的不同类型采用三种数据集来评估本文方法的识别效果:

(1) Caltech256 数据集[16]:包含约 30 607 幅图片,每幅图片大小不固定,总大小为 1.1 GB,经过 5 种密码算法加密后生成的密文数据集大小为 6.25 GB。选取其中 5 GB 左右的数据作为训练集。

(2) LibriSpeech 数据集[17]:包含大约 1 000 小时的英语语音的大型语料库,通过分割对齐方式,将其整理成每条 10 s 左右的音频文件。总大小为 6.3 GB,经过 5 种密码算法加密后生成的密文数据集为 28.6 GB,选取其中的 10 GB 左右的数据作为训练集。

(3) THUCNews 数据集[18]:包含 14 类共计 74 万篇的新闻文档,均为纯文字且大小不固定。选取其中 568 个文件(15.3 MB),经过 5 种密码算法加密后生成的密文数据集为 125 MB,使用其中的 90 MB 的数据作为训练集。

将三种数据集中未选取的密文数据按照图片、音频和文本混合 10∶10∶1的比例进行混合作为测试集。

4.3 评价指标

对于密码算法识别的评价指标主要依据识别准确率来进行评价,准确率(所有预测正确的占总的比重)的计算方法如下:

$$a_{\text{ccuracy}} = \frac{T_{\text{P}} + T_{\text{N}}}{T_{\text{P}} + T_{\text{N}} + F_{\text{N}} + F_{\text{P}}} \tag{8}$$

对于分类模型而言,计算属于某类密文正确识别为该类型的数量可以用真阳性(True Positive,TP)表示;计算属于某类的密文错误识别为该类型的数量用假阳性(False Positive,FP)表示;计算属于某类的密文错误识别为其他类型的数量用假阴性(False Negative,FN)表示;计算属于某类的密文正确识别为其他类型的数量用真阴性(True Negative,TN)表示。其中 $T_{\text{P}} + F_{\text{P}} + F_{\text{N}} + T_{\text{N}}$ 的结果就是样本总数。

使用 K-Fold($K = 10$)交叉验证,将验证后的平均识别准确率作为最后的识别效果。

4.4 实验结果分析

4.4.1 密文二分类识别测试

目前大多数文献中识别准确率表现较好的都是对两种密文进行分类。本文首先进行横向对比,选取 AES、3DES、Blowfish、RSA 和 SHA-1 五种密码算法加密得到的密文进行两两分类识别,组成 10 种分类识别任务进行实验。

为了验证所提 FBR 密文特征的分类识别效果,本文选取文献[10,14]中的随机性测试方法和熵特征提取方法与其对比。在表 3 中展示出 FBR 特征与其他两种特征在不同的对称密码算法二分类任务下的识别准确率比较。从实验结果可以得出,FBR 密文特征提取方法在密文二分类效果上优于随机性测试和熵特征提取方法。

表 3 FBR 特征与其他特征的分类效果对比(%)

分类任务	随机性特征	熵特征	FBR 特征
AES 和 3DES	64.6	81.3	87.1
AES 和 Blowfish	66.3	86.6	89.5
3DES 和 Blowfish	67.2	71.4	91.3

此外,为证明 FBR 特征可以分类识别出不同的密码体制,将 10 种分类识别任务细分为对称与非对称二分类、对称密码算法二分类、对称与哈希算法二分类,共计 3 类密码体制进行实验,实验结果如图 1、图 2 和图 3 所示。

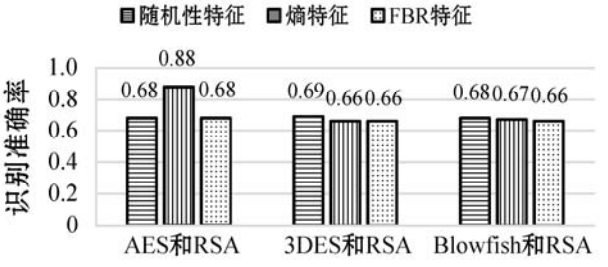

图 1 对称与非对称加密算法二分类结果对比

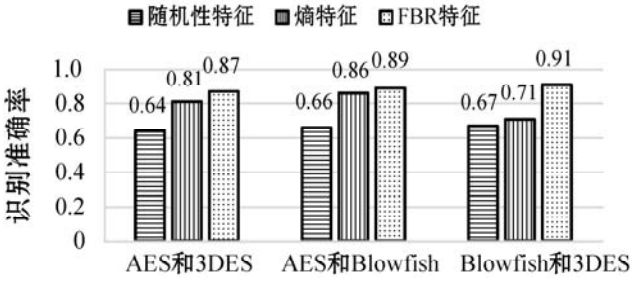

图 2 对称加密算法二分类结果对比

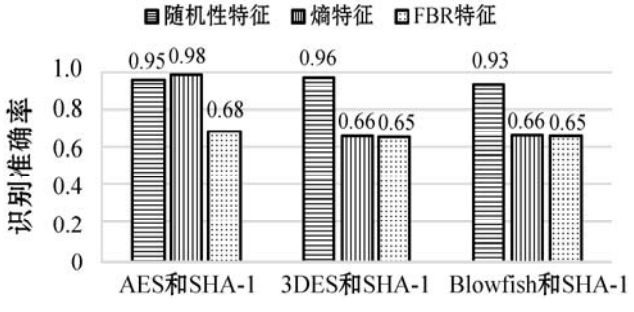

图 3 对称加密算法与哈希算法二分类结果对比

可以看出,FBR 特征在对称加密算法的分类中有更好的表现,特别是在 Blowfish 和 3DES 的二分类上效果非常好,准确率达到 91%。其原因是 AES、3DES 和

Blowfish 这三种算法属于对称密码算法下的分组密码,FBR 特征的设计与提取均是在分组密码下进行的,它更能表征由分组密码生成的密文信息。在对称加密算法与非对称加密算法的二分类中,熵特征在 AES 与 RSA 的分类准确率高于随机性特征和 FBR 特征,准确率达到 88%。此外,随机性特征在对称与哈希加密算法的分类中也同样有不错的表现,准确率均在 90% 以上。

4.4.2　密文多分类识别测试

现有研究在密文多分类问题上分类效果较差,本文采用 FBR 特征进行密文多分类实验,在 AES、Blowfish、Camellia、DES 和 IDEA 五种分组密码算法下进行密文的多分类实验。对比文献[4]表现较好的 H_128 特征,FBR 特征在密文多分类的平均识别准确率提高 7.7 百分点。实验结果如表 4 所示。

表 4　多分类任务识别准确率比较(%)

分类任务	H_128 特征	FBR 特征
AES Blowfish Camellia DES IDEA	79.1	86.8

5　结　语

本文提出一种 FBR 密文特征提取方法。该方法结合随机性测试中的频率、块内频率和游程三种方法,定义出码元次数统计值、块内码元次数统计值和游程次数统计值。基于上述三种统计值构造出一种新的 FBR 密文特征。实验使用支持向量机对三种混合数据集分别进行密文二分类和多分类实验。实验结果表明,本文方法所提取的 FBR 密文特征对比已有表现良好的密文特征,其平均识别准确率得到较高的提升,充分证明了本文方法的有效性。

参　考　文　献

[1] Schneier B. Applied cryptography: Protocols, algorithms, and source code in C[M]. John Wiley & Sons,1996.

[2] 吴文玲,冯登国,张文涛. 分组密码的设计与分析[M]. 2 版. 北京:清华大学出版社,2009.

[3] 黄良韬,赵志诚,赵亚群. 基于随机森林的密码体制分层识别方案[J]. 计算机学报,2018,41(2):382 - 399.

[4] 赵志诚. 基于机器学习的密码体制识别研究[D]. 郑州:战略支援部队信息工程大学,2018.

[5] 吴杨,王韬,邢萌,等. 基于密文随机性度量值分布特征的分组密码算法识别方案[J]. 通信学报,2015,36(4):150 - 159.

[6] 范丽敏,冯登国,陈华. 基于熵的随机性检测相关性研究[J]. 软件学报,2009,20(7):269 - 278.

[7] Dileep A D, Sekhar C. Identification of block ciphers using support vector machines[C]//IEEE International Joint Conference on Neural Network Proceedings,2006:2696 - 2701.

[8] Nagireddy S. A pattern recognition approach to block cipher identification[D]. Institute of Technology Madras,2008.

[9] 李洪超. 基于密文特征的密码算法识别研究[D]. 西安:西安电子科技大学,2018.

[10] 赵志诚,赵亚群,刘凤梅. 基于随机性测试的分组密码体制识别方案[J]. 密码学报,2019,6(2):177 - 190.

[11] Chandra B, Varghese P. Applications of cascade correlation neural networks for cipher system identification[J]. International Journal of Computer, Electrical, Automation, Control and Information Engineering,2007,1:369 - 372.

[12] Sharif S O, Kuncheva L I, Mansoor S P. Classifying encryption algorithms using pattern recognition techniques[C]//IEEE International Conference on Information Theory and Information Security,2010:1168 - 1172.

[13] Souza W A R, Tomlinson A. A distinguishing attack with a neural network[C]//13th International Conference on Data Mining Workshops,2013:154 - 161.

[14] 师国栋,康绯,顾海文. 随机性测试的研究与实现[J]. 计算机工程,2009,35(20):145 - 147,150.

[15] Random bit generation[EB/OL]. [2021 - 04 - 20]. http://csrc. nist. gov/groups/ST/toolkit/rng/documentation_software. html.

[16] Caltech256 数据集[EB/OL]. [2021 - 04 - 20]. http://www. vision. caltech. edu/Image_Datasets/Caltech256/256_ObjectCategories. tar.

[17] Open SLR[EB/OL]. [2021 - 04 - 20]. https://www. openslr. org/12/.

[18] 清华文本分类数据集[EB/OL]. [2021 - 04 - 20]. https://thunlp. oss-cn-qingdao. aliyuncs. com/THUCNews. zip.

(上接第 326 页)

[13] Wang C, Bochkovskiy A, Liao H. Scaled-YOLOv4: Scaling cross stage partial network[EB]. arXiv:2011.08036,2020.

[14] Wang C, Liao H, Wu Y, et al. CSPNet: A new backbone that can enhance learning capability of CNN [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops,2020:1571 - 1580.

[15] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018: 8759 - 8768.

[16] Han K, Wang Y, Tian Q, et al. GhostNet: More features from cheap operations[EB]. arXiv:1911.11907,2020.