关于哈希算法的报告

作者：聂劭质　学号：1500012645

【摘要】哈希算法（Hash algorithm）是一类算法，以可以将任意长度的二进制数据映射为有限长度的二进制散列（称为哈希值，hash values）为特征。不同的哈希函数有不同的特性。利用哈希算法的高度压缩能力与散列的抗修改能力，哈希算法在数据加密、数据完整性验证与数据索引等方面有重要应用。本文将介绍几种常见的哈希函数（Hash function），分析哈希算法的应用范围，最后探讨目前哈希算法的发展。

【关键词】哈希算法；哈希函数；MD5；数据加密；应用

一、哈希算法的实质与几种常见哈希函数

哈希算法实质上是任意长度可以转化为二进制的数据到较短的固定长度二进制数据的一种映射。相同的原始数据通过哈希算法转化得到的哈希值必须相同，所以，哈希算法是一种函数[1]。哈希函数的定义域是无限的，值域却受限于哈希值的长度，是有限的，因此，哈希函数不可能是单射函数，且不存在反函数。

尽管哈希算法理论上可以由多个不同的原始数据得到相同的哈希值，但是哈希值在其容量内可储存的文件数远远超过了目前应用哈希值的文件数。一个常见的 128 位哈希值可以标识 2^{128} 个不同的文件，可见不同原始数据哈希值的碰撞可以忽略。

符合以上说明的函数有很多，比较简单的有加法 hash、位运算 hash、乘法 hash、除法 hash。

加法 hash 即将输入的数据逐个相加得出结果，再对一个常数取模而得到固定长度结果。为了增强哈希值的随机性与元素分布的均匀性，这个常数常常是质数，尽管这种方法的原理尚未被确定。位运算 hash 则是对数据按位运算再去摸。乘法、除法 hash 原理类似，不再介绍。

以上简单的哈希算法由于运算简单，易于从哈希值中获得部分原始数据信息，并且目前已经可以通过特定途径构造出不同的，具有相同哈希值的原始数据，所以加密效果不好。经过几十年的发展，应用中的哈希算法安全程度已经足够应对一般的破解方法。下面介绍几种常见的哈希算法。

MD4 算法：MD 指信息摘要算法（Message Digest Algorithm）。MD4 算法是 MD 算法家族中的一个比较成熟的算法，由麻省理工学院的 Ronald Rivest 教授于 1990 年提出[2]。MD4 算法的哈希值有 32 位十六进制数组成，其计算方式如下：先将数据长度补充到对 512 取模余 448 的长度，补充数据长度为 1 至 512 不等，但必须补足。再通过每 512 位分组迭代计算出最终值，经过计算后，任意长的数据均可得出 128 位哈希函数值[3]。

但是，通过寻找 MD4 算法输出值碰撞对 MD4 算法进行攻击的方式已经被找到，例如以下不同数据的哈希值相同：

k1=839c7a4d7a92cb5678a5d5b9eea5a7573c8a74deb366c3dc20a083b69f5d2a3b
b3719dc69891e9f95e809fd7e8b23ba6318edd45e51fe39708bf9427e9c3e8b9

---

[1] https://en.wikipedia.org/wiki/Hash_function

[2] "The MD4 Message Digest Algorithm". Network Working Group. October 1990. Retrieved 2011-04-29.

[3] “MD4 算法分析”, 黎琳, 山东大学学报（理学版）2007 年 4 月

k2=839c7a4d7a92cbd678a5d529eea5a7573c8a74deb366c3dc20a083b69f5d2a3b
b3719dc69891e9f95e809fd7e8b23ba6318edc45e51fe39708bf9427e9c3e8b9

其 MD4 哈希值都是 4d7e6a1defa93d2dde05b45d864c429b[4]。

现在的 MD4 因为其漏洞已经不再被大规模使用，但是它的基本思想却被另外的算法继承，比如 MD5 与 SHA1 算法。

MD5 算法的输入方式和输出长度与 MD4 算法完全相同，并保留了 MD4 对数据的 512 位分组形式。不同的是，MD5 算法引入了 safety-belt 功能，并增加了第四轮运算，减小了函数的对称性，使得数据碰撞的可能性大大降低。

SHA1 算法由美国国家标准与技术研究院、美国国家安全局一同设计，其名称意为安全哈希算法。与 MD4 和 MD5 算法不同，它对于 2^{64} 以下的数据产生一个 160 位的散列，所以抗穷举性比前两者都要强，同时其运算速度也比前两者更高[5]

二、哈希算法的应用

凭借哈希函数加密的单向性，哈希值的离散型与高度压缩能力，哈希算法在数据结构、密码处理、数字签名、校对信息、识别文件等方面有重要应用。

在 Python 语言中的字典中，通过 key 调用 value 的过程与字典大小无关。这种复杂度为 O(1)的算法既是通过哈希函数实现的。字典中的每一个 key 可以得出一个二进制数，而 value 就储存在这个二进制数所标记的内存地址中，所以，任意输入的 key 经过运算就可以直接得出 value 的地址。这就是哈希算法在数据结构中的一种应用。

密码处理上，对用户设置的密码可以先通过哈希算法加密，再储存在设备上。如果不通过加密直接储存明码，一旦数据库数据泄露，用户的密码就全部被暴露，后果不堪设想。但是通过哈希函数加密后的数据即使被暴露，也无法得知原始密码。在验证密码时，通过对比在客户端用输入密码得出的哈希值与数据库中的哈希值是否相同即可。例如，在 UNIX 系统中，用户密码就是以 MD5 加密后的哈希值储存。

数字签名是为了验证文件发布者身份而设计出的电子签名。在传输文件时，传输方先以特定的哈希算法计算出文件的报文摘要。文件传输完成后，接收方以相同的哈希算法计算接收到的文件的摘要。如果在传输中有人恶意篡改文件，通过哈希值的对比即可轻松发现。如果哈希值相同，则可确定文件未被篡改。这方面的典例是微软发布软件时会同时发布软件的 SHA1 值来确保文件安全。

校对信息的原理与数字签名类似，但其主要目的不是防止外部攻击，而是发现大文件由于传输中的小错误。因为原始数据中细微的不同反映在哈希值上也存在巨大差异，所以在接受大文件后经过一次哈希值验证可以有效发现并防止文件错误。

最后要介绍的是识别文件的功能。这方面最为人熟知的就是 P2P 的数据传输方式，而其中又以 Cohen 以 Python 最初写出的比特流（BitTorrent）最为著名[6]。以下就以 BitTorrent 为例来介绍哈希算法的此应用。对于任何一个文件，都可以通过哈希算法生成一个独一无二的哈希值，torrent 文件通过对比自己将要下载的文件的哈希值与网络上其他用户已有文件的哈希值确定目标文件在网络上的存

---

[4] https://en.wikipedia.org/wiki/MD4

[5] “关于对哈希算法的研究与应用”, 黄云轲等, 计算机光盘软件与应用 2012 年第 3 期

[6] https://en.wikipedia.org/wiki/Bram_Cohen

在位置，然后通过文件原站点与用户间的传输行为进行文件传输，大大的加快了文件传输速度，并减轻了原服务器的负担。这种方式的文件传输无源头、无中心、分布式传输，所以文件一旦存在于互联网上，就几乎不会消失。这也给它带来了不小的争论。

三、哈希算法的发展状况

目前普遍使用的哈希算法为 MD5 算法，不幸的是，此算法的弱点已经渐渐展现出来。2004 年在美国加州举办的美国密码学会议（Crypto）上，中国山东大学的王小云教授公布了数个哈希算法的破解结果，其中就包括 MD5 算法[7]。尽管不是完全破解，但造成杂凑冲撞也使 MD5 算法面临危机。更令人惊奇的是，一年之后，王小云又宣布破解了 SHA1 算法。

不过，在非绝密的加密过程中，MD5 算法的地位仍未被撼动。目前最有可能取代 MD5 算法的仍然是 SHA 系列算法，因为 MD5 算法的安全性与速度都没有 SHA 系列的算法好。

包括 MD2 在内的一系列 MD 哈希算法都是由 MIT 的 Ronald L. Rivest 教授提出的。为应对 MD5 的被破解，在 2008 年的 Crypto 上，Rivest 教授又提出了 MD6 算法[8]，并在之后向美国国家标准与技术研究院的 SHA-3 研究提出建议。这几乎是目前最新的哈希算法。

四、结语

哈希算法是一种发展时间较长但仍有活力的算法。目前的哈希算法在计算机科学、密码学甚至地理信息科学等方面都重要应用。正因为这样，不断地破解现有哈希算法与设计新的难以破解的哈希算法是目前密码学研究的焦点之一。这方面，希望相关技术人员能够越走越远，让我们的信息安全得到提升。

---

[7] “How to Find Weak Input Differences for MD5 Collision Attacks”, T Xie, D Feng, Cryptology ePrint Archive, Report 2009/223

[8] The MD6 Hash Function. Ronald L. Rivest. Invited keynote talk given at CRYPTO 2008 (2008-08)