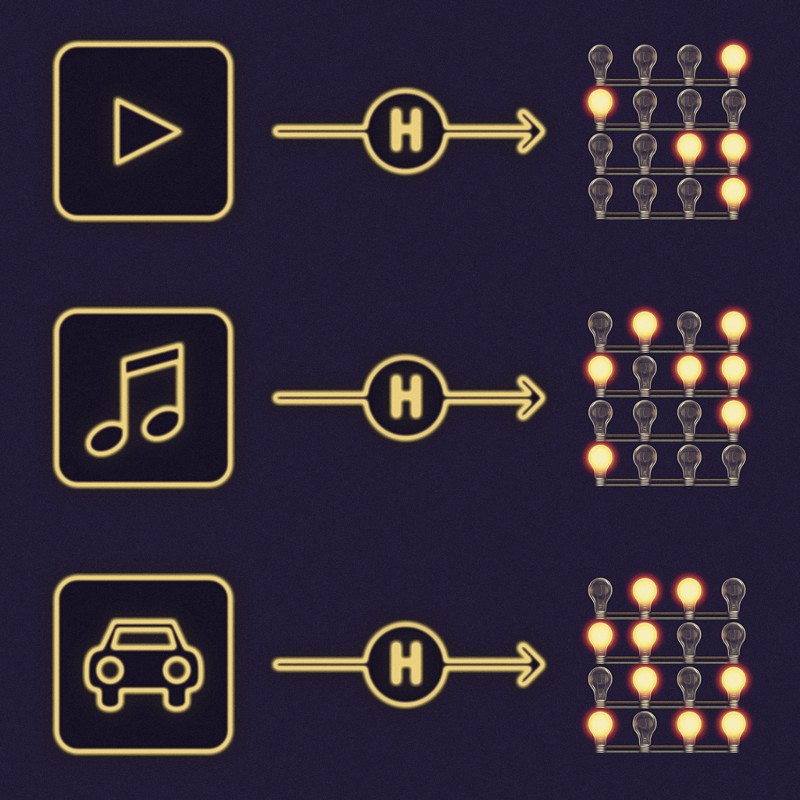
区块链基础：散列法（Hashing）

[**洒脱喜**](http://www.8btc.com/author/3708)**2016-01-14 22:16 发布在**[**区块链**](http://www.8btc.com/blockchain)**,**[**技术指南**](http://www.8btc.com/how)[**12**](http://www.8btc.com/blockchain-underpinnings-hashing#comment)**8403**



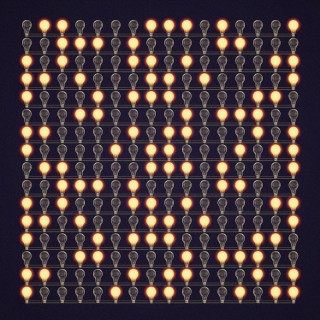
灯泡，比特（bits）与字节（bytes）

你可能知道计算机中所有的数据都是由0或1组成的，最小的数据单位就是一个比特（bit，或位），它也是0或者1。想象一下，一台计算机拥有着很多的灯泡，而这个灯泡的状态有两种，亮（1）或者灭（0）。而不同的数据，由灯泡显示的图案也是不同的。大数据如视频，就使用了相当多的灯泡，而一个简短的电子邮件，其所需要的灯泡就较少。一个单一的灯泡代表着一个比特。另外，你可能听说过一个词叫字节，一个字节就相当于8个灯泡的组合。而1MB的数据约为100万个字节，也就相当于800万个灯泡。

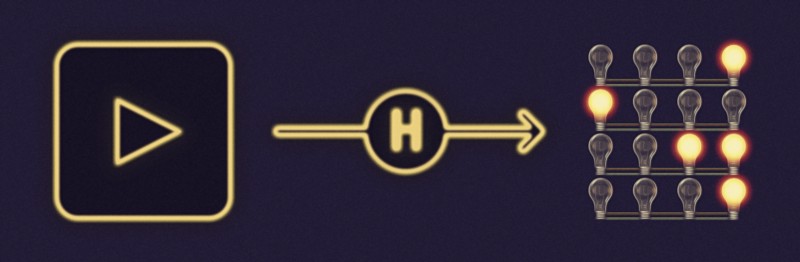
如今，家用的电脑就拥有了数十亿甚至万亿级数量的灯泡。但我们发现，即使只是由256个灯泡组成的集合，也足以代表宇宙中能够观察到的任何颗粒。想象一下256个灯泡组能够产生的所有图案，那将是一个天文数字：也就是2^256种可能性。

加密散列函数（或加密哈希函数）

一个散列函数(hash function)，即取任何的输入，就可以产出一个特定大小的输出。这个运用散列函数，然后产出某些数据的过程，我们称之为散列法（hashing）或音译为哈希法。而散列函数的输出，我们称之为一个散列（hash）。一个特定散列函数的基本特征，就是它产出输出的大小。比方说本文中的示例，我们使用一个产出输出为256 bits（32字节）的散列函数。当然也有散列函数能够产出较小的输出，或者也可以产出较大的输出，也存在另外一些能够产出256 bits的散列函数，但这个例子中，我们并不关心具体所使用的散列函数。



使用这个例子的散列函数，当一部N兆（MB）的视频被散列运算时，那它的输出结果为：256个灯泡中有一些灯泡是点亮的。当一个简短的电子邮件被散列运算时，这256个灯泡的输出显示，则是另外的一种图案。在某些方面，散列法看起来就像是压缩。**简单地解释下这两者之间的区别，散列法总是会产生相同数量的灯泡，而压缩一部N兆（MB）视频的结果，仍然会产生数以百万计灯泡的一个输出。一个压缩过的视频，可被解压缩然后获得原始的视频。而当一个视频被散列到仅仅只有256个灯泡时，从这个散列来重新构建原始视频的可能性就很小了。**这可能听起来并不是理想的，但实际上这正是散列函数的一个强大功能。



一个安全的加密散列函数，它的一个关键特征就是，它是单向的。这意味着，从数学和计算机学角度上来看讲，从输出来反推输入，这几乎是不可能的。也就是说，给定一个散列，想要了解或查到提供给这个散列函数的输入数据，它应该是不可行的。技术术语上来讲，我们称它为逆原像阻力（pre-image resistance）。

结果是，无论是散列法运算一个较大或者一个较小的输入，散列函数应消耗大约相同的时间量。另一个理想中的结果是，这个散列，也就是由散列函数而产生的灯泡图案，似乎应是随机的，对数据“password1”进行散列法运算，其产生的灯泡图案，与对数据“password2”进行散列法运算而产生的灯泡图案，两者是有很大不同的。否则，如果图案是相似的，那对方就可以推断出输入也是类似的，而如果相关的词（如“pass”，“word”）被发现时，那密码也很容易被找到。安全的散列函数，即使输入仅相差一个bit，也会产生显著不同的输出。

安全的理想行为，是给定一个散列，而唯一找到输入数据的方法，就是通过对所有输入的组合进行散列法运算，直到正确的输入是被散列运算了。如果输入是随机的，那找到它的时间既是不确定的。

虽然找到一个散列的输入应该是非常困难的，它需要花费很长的时间，但计算一个散列却是很快就能完成的。一个带有大量输入的散列函数，可能在不到一秒的时间内，就能得到输出。考虑到今天智能手机，每秒能够进行数十亿次的计算，1秒对于计算而言，就相当于很长的时间了。

加密散列函数也应该是抗碰撞的（collision resistant）。**一个碰撞过程，意指当一个散列函数为超过1个输入进行运算，而产出相同输出的结果。如果用散列法运算数据1（可能是一份电子表格），而用散列法运算数据2（可能是一张图片），这两者产生了相同的输出，那么这个碰撞冲突就发生了**。

加密散列函数，其安全性的重要性，在我们描述区块链和散列法部分时，会显得更为清楚。

区块链和散列法

散列法（Hashing）广泛地应用于区块链，这里也有一些例子。

区块链上的地址，是由散列法运算公钥而得到的。一个以太坊的账户地址，是以[Keccak-256](http://keccak.noekeon.org/yes_this_is_keccak.html)（开发者应该阅读下它与SHA3-256的关键区别）散列法运算一个公钥而得出的。而一个比特币地址，则是通过[SHA2–256](https://en.wikipedia.org/wiki/SHA-2)和[RIPEMD160](https://en.wikipedia.org/wiki/RIPEMD)来散列法运算一个公钥而得出的。

散列函数的抗碰撞性是重要的，因为如果2个人产生了相同的地址（发生了冲突），那任何一方都可以花费这个地址上的钱。

签名也是区块链的基本组成部分。类似于签署一张支票，加密签名决定哪些交易是有效的。签名是由私钥和需要被签名的数据散列而生成的。

交易散列在区块链中是非常明显的。比方说描述一笔交易：“Alice在D日T时，向Bob发送了X单位的货币”，那么交易就会被提交为他们的散列，例如[5c504ed432cb51138bcf09aa5e8a410dd4a1e204ef84bfed1be16dfba1b22060](https://live.ether.camp/transaction/5c504ed432cb51138bcf09aa5e8a410dd4a1e204ef84bfed1be16dfba1b22060)是以太坊区块链中的一笔交易。交易散列也是更为直接可用的，例如“在1337个区块中的第1024笔交易”这样的描述，你只需要复制这个散列，并粘贴到一个区块链浏览器中，然后就可以查看这笔交易的细节。

形而上学地讲，区块链中的区块是由它们的散列来确定的，其充当了鉴别和完整验证的双重角色。一个识别字符串还会提供它自有的完整性，被称为自认证标识符。

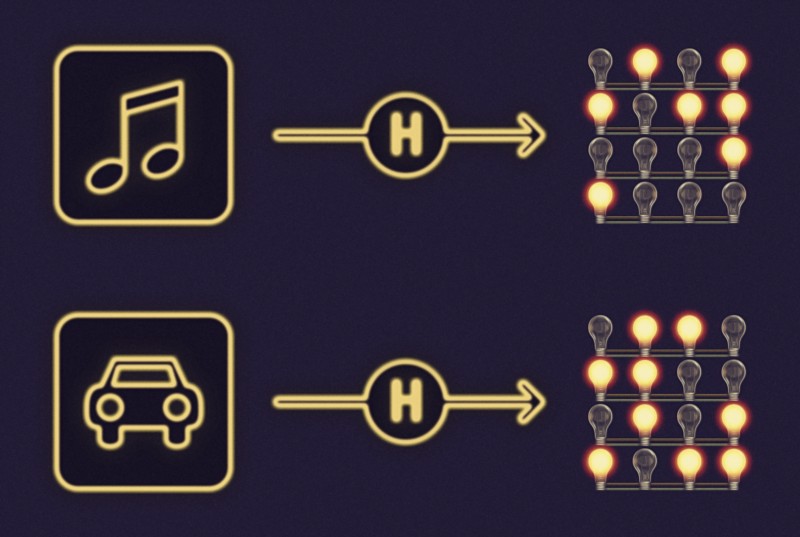
对于使用挖矿机制的区块链来说，工作量证明（Proof-of-Work）就是一个数字，我们称它为随机数（nonce），当它和其他散列过的数据进行合并时，会产生一个比规定目标值更小的值。挖矿使得散列法成为一种快速运算、单向不可逆的算法。找到一个有效的随机数需要时间，因为（矿工）没有可用的线索来帮助它们找到一个足够小的散列，而唯一找到一个小于目标值的方法，就是计算很多的散列：在比特币中，目前存在了超过10^25(10 septillion)数量级的散列。当一个（nonce）随机数被找到时，验证它的时间就需要1秒，然后这个新区块会在网络中广播，形成最新的共识和区块链。

在区块链上的存储数据是永久性的，但把大量的数据存储在区块链上则是不明智的，而实用的区块链存储方法，是将固定大小（通常是小的）的数据代表存储在区块链上，我们称之为“数据的散列”。区块链的一个应用是作为一个时间戳服务。假设你想要证明一张当前存在的图片，保证在未来时它不是编造出来的。你可以将图片的散列存储在区块链上，一年以后，当法官问起这张图片是否在一年前真实存在时，你可以提供这个图片，然后法官就可以散列运算这张图片，并与你存储在区块链上的散列进行对比。

散列法还涉及到更多高级的例子，例如区块链、可扩展性、轻钱包创新的根本 —— 梅克尔树（Merkle tree）。

用于安全识别的散列

安全加密散列函数是单向、快速计算，并且抗碰撞的。结合这些特点后，它们会处理任何类型的输入，然后产生一个固定大小的输出，称之为散列，散列作为任何数据的标识而言，是非常有用的。长度256 bits的散列，代表了一个天文数字的组合，将它们用于全球物联网的唯一标识符，那也是绰绰有余的，即使是在纳米技术规模下，这些散列也可以被写为64个字符（十六进制），这使得它们足以作为标识符来使用。**在区块链中，散列是作为区块、交易和地址的标识符**。



散列还享有安全与隐私的优势。如果一首歌是以数字格式被记录的，并且这首歌的散列是被记录在区块链之上的，那任何他人都无法声称是他们是第一个创造了这首歌，并生成了这个散列，他们也不会知道歌曲本身：某人不能写歌，也没法篡改这个散列。同样地，除非歌曲或其他数字化财产或数据被表明了，展示在区块链上的仅仅是散列本身而已。 所有权记录也可以存储在区块链上，举个简单的例子，车辆登记处可以将汽车数据散列（照片，VIN, 车牌）存储在区块链上，只有车辆所有者，保险公司以及政府会知道这个车辆的实际细节。

深入理论，广泛应用

设计加密散列函数，需要艺术与科学的结合。为了证明它们的安全性，就需要用到先进的数学与计算机科学。区块链是为广大人群提供的，第一个充满散列的用户界面。好的用户体验，其背后隐藏着很多的散列，但正如我们今天看到的各种 id和序列号，有时候散列会是替代长篇大论的最佳标识符。随着加密技术与物联网技术变得更加普及化，希望在未来能够看到更多64字符的散列！